
**Universidad Autónoma de Madrid**

Departamento de Ingeniería Informática

# On-Line Video Abstraction

PhD Dissertation written by

**Víctor Valdés López**

under the supervision of

**Dr. José María Martínez Sánchez**

Madrid, April 2010

**Universidad Autónoma de Madrid**

Departamento de Ingeniería Informática

En Madrid a ..... de ..... de 2010.

Constituido el tribunal compuesto por:

- Presidente
  - D. .....
- Vocales:
  - D. .....
  - D. .....
  - D. .....
- Secretario:
  - D. .....

Por cuanto D. Víctor Valdés López ha defendido su tesis titulada "*On-Line Video Abstraction*", este tribunal le otorga la calificación de:

.....

Fdo:..... Fdo:.....

(Presidente) (Secretario)

Fdo:..... Fdo:..... Fdo:.....

(Vocal) (Vocal) (Vocal)

*To María and my family.*

# Acknowledgments

First of all I would like to thank Chema and Jesús because they trusted me from the beginning, especially Chema: thank you for guiding me during these years, this would have not been possible without your help, advice and patience.

Thanks to my parents, my brother, and Laura, who encouraged me to begin this journey and have supported me all these years.

I can not imagine how this moment would be without María, my companion during this adventure, who has suffered and enjoyed all the aspects of writing a PhD (and living with me): I love you.

This experience would have been incomplete without sharing all those moments, projects, meetings and papers with my 'classic' *VPULab* mates (Javi, Luis I, Fabrizio, Víctor F., Fernando, Juan Carlos, Álvaro II, Marcos, Miguel Ángel), the 'new' ones (Álvaro III, Luis II, Virginia), those who left (Jorge H. and Álvaro I) and the people from the *Information Retrieval Group* (Pablo, David, Iván, Miriam, Alex).

I would also like to thank the rest of inhabitants of the B-408, Ana, Lobato brothers, Mariano and Ignacio, and, of course, Jorge 'paquete', my favourite squash opponent.

I want to thank Noel O'Connor for his support during my stay in the *Dublin City University*, my first foreign experience. Also thanks to Charlie, Anna and Ana, whom with I shared many moments there. My stay in Pittsburgh was also very important for me, I'd like to thank Alex Hauptmann from the *Carnegie Mellon University* for his advice and his warm hospitality. Of course, I have to thank 'livepittsburgh' people (you made my stay a great experience guys!), very specially to Cris and Keko.

Finally, I would like to thank all my friends, for being there always, José and Nuria, for your support, and to everyone who helped me but I forgot to mention here: thank you.

Work supported by the European Commission (IST-FP6-027685 - Mesh), Spanish Government (TIN2007-65400 - SemanticVideo), Comunidad de Madrid (S-0505/TIC-0223 - ProMultiDis - CM), and by the Consejería de Educación of the Comunidad de Madrid and The European Social Fund through a Contrato de Personal Investigador de Apoyo.

# Abstract

Nowadays, the huge amount of video material stored in multimedia repositories makes its search and retrieval a very slow and usually difficult task. Existing video abstraction systems aim to relieve this problem by providing condensed versions of the original content which ease the search and navigation processes and reduce the browsing time. In the last years, many different video abstraction approaches, based on the optimal selection and presentation of a subset of fragments (keyframes, shots, etc.) from the original video attending to different criteria, have been developed. The applied mechanisms usually depend on the application scenario and almost any kind of video content has been subject of interest for the development of video abstraction techniques: music videos, sports, news, surveillance recordings, movies, home videos, etc. The developed techniques have proven to provide useful tools for easing the search and retrieval of video content. Nevertheless, given the huge size and growth rate of existing video repositories, the efficiency of the developed approaches is a limiting factor for their integration in practical scenarios or commercial applications: there is an increasing need for providing efficient techniques. This work is focused in the study and development of such efficient techniques for video content abstraction, aiming for *on-line* performance.

After presenting an overview of existing video abstraction approaches, aimed to provide a general idea of the wide variety of existing techniques, a novel taxonomy and a general architecture for video abstraction systems are proposed in order to establish a common framework for the study and classification of existing abstraction algorithms based on their operational characteristics.

The definition and general requirements for the development of *on-line* and *real-time* abstraction systems are then established, together with a study of the possible implications of such operation modalities in terms of development constraints or limitations in the applicable techniques. Taking into consideration those implications and constraints, two generic *on-line* video abstraction systems are proposed.

The evaluation of video abstraction approaches has always been a very difficult task, due to the high amount of required time and human resources. In this work, as part of the evaluation of the developed algorithms, a novel framework for automatic video summaries evaluation is proposed. Such framework is applied for an exhaustive evaluation of the proposed algorithms, comparing them with other *off-line* abstraction approaches and demonstrating the high competitiveness of the proposed techniques.

Finally, the last part of this work describes two applications based in the proposed *on-line* video abstraction algorithms which demonstrate the potential possibilities of *on-line* abstraction techniques. The first one is devoted to the generation of news broadcasts bulletins in broadcast time. The second application consists in an interactive *real-time* video summaries generator and player which allows the user to watch and modify the generation parameters of video summaries on the fly.

# Resumen

La gran cantidad de material almacenado hoy en día en repositorios multimedia provoca que su búsqueda y acceso se conviertan en actividades lentas y, en muchos casos, difíciles. Los sistemas de generación de resúmenes de vídeo abordan este problema proporcionando versiones condensadas del contenido original que facilitan y reducen el tiempo invertido en su búsqueda y recuperación. En los últimos años se han desarrollado gran cantidad de algoritmos basados en la selección y presentación, en función de un criterio determinado, de un subconjunto óptimo de elementos del vídeo original (imágenes, segmentos de vídeo). El tipo de técnicas de empleadas varía en función del escenario de aplicación, siendo objeto de interés prácticamente todos los tipos posibles de contenido: vídeos musicales, noticias, deportes, vigilancia, vídeos caseros, etc. Las técnicas aplicadas han demostrado su utilidad pero, dada la gran cantidad de contenido existente y su ritmo de crecimiento, la eficiencia de las técnicas de generación de resúmenes es un factor que limita su integración en aplicaciones comerciales: hay una creciente necesidad de desarrollar técnicas computacionalmente eficientes. Este trabajo se centra en el desarrollo y estudio de dichas técnicas, aplicadas a la generación de resúmenes de vídeo, con el objetivo final de generar resúmenes 'en vivo'.

Tras la presentación de un resumen de técnicas actuales de generación de resúmenes, con el objeto de proporcionar una idea general de la gran variedad existente, se propone una taxonomía y arquitectura genéricas que permiten establecer un marco común para el estudio de las distintas aproximaciones al problema desde un punto de vista 'operacional'.

A continuación se detallan los requisitos de los sistemas que hemos caracterizado como *on-line* (en vivo) y *real-time* (en tiempo real) junto con un estudio de las implicaciones y limitaciones que dichos sistemas pueden presentar a la hora de su desarrollo. Teniendo en cuenta dichos requisitos y limitaciones, se proponen dos sistemas de generación de resúmenes *on-line*/*real-time*.

La evaluación de los sistemas de generación de resúmenes de vídeo ha sido una tarea generalmente muy costosa debido a la subjetividad respecto a la calidad de un resumen de vídeo y a la gran cantidad de recursos (humanos y de tiempo) necesarios para llevarlas a cabo. En este trabajo se ha desarrollado un sistema para la evaluación automática de resúmenes. Dicho sistema es utilizado para la evaluación en profundidad de los algoritmos desarrollados, comparándolos con técnicas existentes y demostrando la competitividad de las técnicas propuestas.

Finalmente, la última parte de este trabajo se centra en el desarrollo de dos aplicaciones que demuestran las posibilidades de las técnicas de generación de resúmenes en vivo. La primera aplicación consiste en un sistema completo para la generación de resúmenes de telediarios en tiempo de emisión mientras que la segunda consiste en un visualizador interactivo de resúmenes de vídeo que permite al usuario modificar los parámetros de generación y visualizar los resultados en tiempo real.

# Contents

# List of Figures

# List of Tables

# Acronyms

**BBC** *British Broadcasting Corporation:* British largest broadcasting network

**BoF** *Block of Frames*

**BT** *Binary Tree:* Video skimming approach based on binary trees

**BU** *Basic Unit:* Minimal processing unit applied within a video abstraction approach composed by information extracted from the original video (e.g. frames, audio samples)

**CCTV** *China Central Television:* China major state television broadcaster

**DCT** *Discrete Cosine Transform:* Mathematical transform related to the Fourier transform

**DU** *Duration of the summary:* One of the evaluation metrics applied in the 2007 and 2008 TRECVid rushes summaries evaluation campaigns

**DW** *Deutsche Welle*: German international broadcasting organization

**EA** *Easiness for Understanding:* One of the evaluation metrics applied in the 2007 TRECVid rushes summaries evaluation campaigns

**GoB** *Group of BUs*

**GoF** *Group of Frames*

**GoP** *Group of Pictures*

**HMM** *Hidden Markov Model:* A type of statistical model

**IN** *Inclusion:* One of the evaluation metrics applied in the 2007 and 2008 TRECVid rushes summaries evaluation campaigns. Corresponds to the fraction of events from the original video included in a summary

**JU** *Junk:* One of the evaluation metrics applied in the 2008 TRECVid rushes summaries evaluation campaigns. Corresponds to the perceived amount of junk content (e.g. blank frames, clapboards) included in a summary

**MPEG** *Motion Pictures Experts Group:* An ISO/ITU standard for compressing digital video

**RE** *Redundancy:* One of the evaluation metrics applied in the 2007 and 2008 TRECVid rushes summaries evaluation campaigns. Corresponds to the amount of redundancy perceived in a summary

**RGB** *Red Green Blue Color Model:* An additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors

**RISPlayer** *Real-Time Interactive Video Summaries Player:* Video browsing and summarization application able to interactively generate and display video summaries in *real-time*

**SCC** *Sufficient Content Change:* Video skimming approach based on the selection of original video fragments different enough to already selected ones

**SIFT** *Scale-Invariant Feature Transform:* An algorithm in computer vision to detect and describe local features in images

**SNR** *Signal To Noise Ratio:* Measure applied to quantify how much a signal has been corrupted by noise

**SURF** *Speeded-Up Robust Features:* An algorithm in computer vision to detect and describe local features in images partly inspired by the SIFT features but several times faster

**SVM** *Support Vector Machine:* A set of related supervised learning methods used for classification and regression

**TRECVid** *TREC Video Retrieval Evaluation:* Series of workshops focusing on a list of different information retrieval research areas in content based retrieval of video

**TT** *Total Time Judging:* One of the evaluation metrics applied in the 2007 and 2008 TRECVid rushes summaries evaluation campaigns. Corresponds to the time required by an assessor to judge a summary

**VT** *Total Time Video Play:* One of the evaluation metrics applied in the 2007 and 2008 TRECVid rushes summaries evaluation campaigns. Corresponds to the video playing time spent by an assessor judging a summary

**XD** *Length Difference with Target:* One of the evaluation metrics applied in the 2007 and 2008 TRECVid rushes summaries evaluation campaigns. Corresponds to the difference between a summary length and the targeted length

**YCbCr** *YCbCr Color Space:* Family of color spaces used in video and digital photography systems. $Y$ is the luma component and $Cb$ and $Cb$ are the blue-difference and red-difference chroma components

# Part I

# Introduction

# Chapter 1

# Introduction

## 1.1 Motivation

Nowadays, video abstraction (also called video summarization) is becoming a need in order to deal with the increasing amount of available video content in networked or home repositories. The amount and variety of available video makes its search and retrieval a more and more difficult task and many times content is lost and never used due to the difficulties to navigate in such large repositories. The search and visualization effort implies a waste of time and, in some cases, of bandwidth because many videos must be downloaded and visualized before the user finds the content he is looking for. These problems can be reduced or eliminated with the application of proper browsing methods. Existing video abstraction techniques offer solutions providing short and representative versions of original videos that can be easily downloaded and watched in a shorter amount of time, reducing as well the employed bandwidth.

There exist a lot of video abstraction approaches mainly focused on the selection of the most representative fragments from the original video attending to a specific criteria. Nevertheless, despite the great number of available approaches, their application and integration in real systems is not spread at all. One of the main reasons for this situation may be the high computational resources required by most of the existing techniques: their application become useless either because the great amount of time needed to compute a video abstract can reduce its value (for example in live events) or because it is not computationally possible to generate video abstracts for all the content in large repositories or at a high enough rate to process the incoming video clips. Most of the commercial video streaming portals with higher number of visitors such as Youtube[1], Metacafe[2], Break[3], Daily Motion[4] or Google Video[5] provide only single keyframe previews of the available videos. The most sophisticated approaches are limited to uniform subsampling (Imeem[6]) or the possibility (available at the experimental video library Open Video Project[7]) of choosing between the first 7 seconds of the

[1] www.youtube.com
[2] www.metacafe.com
[3] www.break.com
[4] www.dailymotion.com
[5] video.google.com
[6] www.imeem.com
[7] www.open-video.org

video, a uniform subsampling storyboard or a fast-forward version of the video. In 2007[8] and 2008[9], the TRECVid BBC Rushes summarization tasks provided an evaluation framework which, beyond the specific characteristics of rushes sequences, enables the possibility of comparing different abstraction approaches. Nevertheless, it was not possible to find any relation between the computational effort required by the different approaches and the quality of the generated summaries (according to the metrics applied in the submissions evaluation).

Taking into account these facts, it should be worth to study the performance of abstraction methods in order to develop useful tools applicable in real/commercial environments.

## 1.2 Objectives

The main objective of this work is the study and proposal of systems not just computationally efficient but fulfilling more restrictive requirements in order to raise up the possibility of applying *on-line* abstract generation. The *on-line* modality, as we define it, implies that the video abstract is generated in a progressive way, that is, while the video is being received, recorded or decoded, with a controlled delay. Although the development of such *on-line* approaches implies to deal with several technical and practical limitations, it will permit novel functionalities for the abstract generation processes aimed to solve part of the explained practical issues which prevent for a higher spread of abstraction systems being implemented in commercial applications.

In order to completely understand and define the operational requirements of *on-line* video abstraction approaches, a prior study of existing abstraction systems must be carried out for the identification of the key issues which may influence their computational performance. Based on such information, alternative solutions for the implementation of *on-line* systems are analyzed, identifying as well possible associated drawbacks and proposing complete *on-line* abstraction systems.

The proposed techniques impose limitations to the way in which abstraction systems may be implemented and, for this reason, an important part of the present work deals with the evaluation of the proposed *on-line* video abstraction systems. Such evaluation is devoted to the comparison between existing abstraction systems and the newly proposed *on-line* ones, aiming to quantify the possible loss of quality due to the constraints imposed to fulfill the *on-line* operational requirements.

Finally, the last objective of this work is to explore the specific functionalities that *on-line* abstraction approaches may provide, apart from their computational efficiency, with the implementation of real applications making use of the proposed algorithms. One of the functionalities is the continuous generation of video abstracts while content is being recorded, broadcasted or uploaded to a repository. In those cases, the user may have an almost instantly available video abstract of already received content without the need of waiting for *off-line* processing. Besides those 'instantly' available video abstracts, the development of fast enough *on-line* generation processes will make possible the generation of personalized video abstracts (from stored content) for *real-time* delivery without the unrealistic option of storing milliards of pre-generated versions and therefore enhancing the system's utility for the user.

[8] http://www-nlpir.nist.gov/projects/tv2007/tv2007.html
[9] http://www-nlpir.nist.gov/projects/tv2008/tv2008.html

## 1.3 Main Achievements

In the present work, we have carried out a study of existing abstraction approaches, from an operational point of view, with the proposal of a novel abstraction systems taxonomy. For the analysis of existing and future abstraction approaches, we have defined a generic abstraction architecture able to model a wide variety of abstraction systems. Taking into consideration the established concepts and models, we have defined the computational requirements of both *on-line* and *real-time* abstraction approaches as well as the practical issues that must be solved for the implementation of such kind of systems.

Two novel *on-line* abstraction approaches, fulfilling the established computational requirements, are proposed. These approaches provide different levels of complexity and functionalities which have been analyzed by means of an original developed automatic video abstract evaluation system. The obtained results demonstrate how the proposed *on-line* abstraction approaches can generate video summaries with quality comparable to *off-line* algorithms.

Finally, two innovative applications are presented. The first application consists in a news abstraction system on broadcast time. The second one conforms a *real-time* interactive video summary player which allows watching the video summaries as they are being generated and permits the interactive modification of the abstract generation parameters displaying the results on the fly. Both applications demonstrate the potential of the developed *on-line* abstraction techniques.

A more detailed description of the novel aspects and contributions of this work can be found in the conclusions chapter 10 while the list of related publications is detailed in appendix D.

## 1.4 Document Overview

The rest of this document is divided in six main parts, including conclusions and appendixes, which are organized as follows:

- **Part II: Existing Video Abstraction Approaches**
  - *Chapter 2 - Overview of State of the Art on Video Abstraction:* This chapter presents an overview of existing video abstraction approaches, summarizing several existing video abstraction classifications and outlining the high variety of techniques, mechanisms and applied features found in the literature.
  - *Chapter 3 - A Framework for Video Abstraction Systems:* This chapter describes a unified taxonomy and a generic architectural model aimed for the study of existing abstraction systems characteristics and computational performance.
- **Part III: On-Line Video Abstraction**
  - *Chapter 4 - On-Line Video Abstraction Systems Requirements and Implications:* This chapter defines the *on-line* and *real-time* video abstract generation modalities and includes a study on the requirements and development implications of both approaches.
  - *Chapter 5 - On-Line Video Skimming Algorithms:* This chapter presents two generic *on-line* video skimming approaches, analyzes the foundations of the underlying abstraction mechanism of both approaches (redundancy removal) and, finally, presents the results

of two submissions, based on both algorithms, sent to the TRECVid 2007 and 2008 BBC Rushes Summarization Task.

- **Part IV: Evaluation**
  - *Chapter 6 - Automatic Evaluation of Video Summaries:* In this chapter a novel automatic video summaries evaluation system is described. The approach makes use of the TRECVid 2008 submitted summaries and respective obtained evaluations for training different measures predictors.
  - *Chapter 7 - On-Line Video Skimming Systems Evaluation:* In this chapter the two *on-line* video skimming approaches previously presented are thoroughly evaluated and compared with *off-line* approaches making use of the proposed automatic evaluation system. The functionalities of the proposed approaches are as well presented and analyzed.
- **Part V: Applications**
  - *Chapter 8 - On-line Video Abstract Generation of Multimedia News:* This chapter presents a complete end-to-end system for the generation of *on-line* news summaries. The proposed approach, based in a combination of content classification, video summarization and composition techniques, allows the generation of TV news video abstracts on broadcast time.
  - *Chapter 9 - Real-Time Video Summaries Player:* In this chapter an application for *real-time* generation and visualization of different types of video summaries, which allows the user to interact with the summary generation process watching the effects of the modifications on the fly, is described.
- **Part VI: Conclusions**
  - *Chapter 10 - Conclusions:* This chapter concludes the present work, summarizing the contents of the different chapters and obtained conclusions, as well as the original contributions together with future work proposals.
- **Part VII: Appendixes**
  - *Appendix A - Example of Application of the Abstraction Systems Framework:* This appendix includes a practical example of application of the taxonomy and video abstraction systems architecture proposed in chapter 3.
  - *Appendix B - News Content On-Line Classification:* This appendix details the video segment category classification approach, based on SVMs, applied for the news content categorization system as part of the news content abstraction application presented in chapter 8.
  - *Appendix C - On-Line News Summarization Evaluation:* This appendix includes the objective and subjective evaluations carried out for the validation of the news content abstraction application presented in chapter 8.
  - *Appendix D - Publications:* This appendix includes the list of published papers grouped by related topic and associated thesis chapter.

# Part II

# Existing Video Abstraction Approaches

# Chapter 2

# Overview of State of the Art in Video Abstraction

## 2.1 Introduction

Research on video abstraction or summarization techniques (both terms are applied indistinctly in different works) has been very productive in the last years. The wide spread of digital multimedia content, together with the high growing rate of existing video repositories and media producers made the video abstraction technologies an interesting field of research. Such active research has generated an enormous variety of video analysis and abstraction approaches focusing on their application for different types of content and scenarios. Almost any kind of video content has been subject of interest for research on specific abstraction techniques. In [1] and [2] a number of possible applications are enumerated: generic approaches [3, 4], music videos [5, 6], sport videos [7, 8, 9, 10], news [11, 12, 13, 14], surveillance [15], movies [16, 17], home videos [18], video lectures [19, 20] or even cooking videos [21].

A video abstraction approach can be defined as "*[...] a technique that abstracts video content and represents it in a compact manner*" [16] and, focusing on the definition of what a video abstract is, an appropriate definition can be found in [17], where a video abstract is defined as "*[...] a sequence of still or moving images representing the content of a video in such a way that the target party is rapidly provided with concise information about the content while the essential message of the original is well preserved*".

Hundreds of references describing different video abstraction approaches can be found in the literature. However most survey works differentiate between two basic types of abstraction algorithms: those which extract static images, denoted as 'keyframes' in [1, 16], 'still image abstracts' in [2], 'static storyboards' or 'video summaries' [22], and those which generate a reduced length video from the original content, commonly denoted as 'video skims' [1, 16] but also 'short clips' [2] or 'moving image abstracts' [22]. Although both approaches are, in many cases, based on the same video processing principles, many authors maintain the conceptual division between both categories because there exist several differences: in the cases of keyframe extraction, the amount of data to process can be reduced and the audio is usually ignored. Additionally, video keyframes provide more possibilities for the presentation of the results. On the other hand, video skims have the possibility of providing a more complete information about the original content by including audio and motion information, but its processing can be more resource consuming that in the case of keyframe extraction. However,

as stated in [1], "*[...] these two forms of video abstract can be transformed from one to another. Video skims can be created from keyframes by joining fixed-size segments, subshots, or the whole shots that enclose them[...]. On the other hand, the keyframe set can be created from the video skim by uniform sampling or selecting one frame from each skim excerpt".*

In this chapter we provide an overview of video abstraction approaches aimed to outline the high variety of existing techniques without trying to exhaustively enumerate them. For a more detailed overview of the state of the art, the interested reader is referred to the different surveys on video abstraction approaches (e.g., [1, 16, 22, 23, 24, 25]) which contain more details about classification categories and examples.

The rest of the chapter is organized as follows: in section 2.2, an overview of existing approaches for video keyframe extraction is provided. Section 2.3 summarizes different categories for video skimming techniques, while section 2.4 describes different types of features in which existing video abstraction techniques rely. Finally, section 2.5 presents the chapter conclusions.

## 2.2 Keyframe Extraction

In the literature, there exist different proposals for the classification of keyframe extraction systems. In this section we will enumerate some of them, depicted in figure 2.1, providing explanation and examples of the categories identified in previous works.

The first classification we will consider (figure 2.1 -A-) is proposed in one of the earliest surveys on video abstraction approaches [22], as well as in [16], and it identifies three different categories depending on the type of video unit applied in the keyframe extraction approach: *sampling based, shot based* and *segment based*. A different approach is provided in [25] (figure 2.1 -B-) , which describes a classification which considers four different categories. Such categories are based, in this case, on the kind of processing mechanism that guides the abstraction process: *shot boundary based, perceptual feature based, feature vector space based* and *cluster based*. It should be pointed that, even providing different perspectives for the keyframe extraction system classifications, those first two proposals contain overlapping categories which will be later explained (e.g. *shot based* and *shot boundary based* categories). The last approach we will take into consideration [1] provides an exhaustive classification of keyframe extraction systems (which contains as well the previously enumerated categories) differentiating between different aspects of the process: *keyframe number*, *unit*, *representation scope* and *underlying mechanism* (see figure 2.1 -C-) . All the enumerated categories will be hereafter explained.

The previously mentioned works, [22] and [16], provide a classification scheme (figure 2.1 -A-) based on the video unit used for the keyframe selection. The 'video unit' concept refers to how the original content, typically frames and audio samples, are grouped for its processing:

- *Sampling based*: those approaches consist on simple random or uniformly sampling approaches for key frame selection [26, 27, 28, 29, 30, 31].

- *Shot based*: this category includes approaches taking a single frame per shot, usually the first frame [32, 33], but also approaches based on color, like [34], where more keyframes are extracted within a shot if their histogram difference with the first keyframe exceeds a limit, or [35], where the shot segmentation and selection of keyframes within a shot are both carried out by histogram-based clustering techniques. Other 'shot based' selection approaches rely on motion measurement, for example based on pixel differences [36] or optical flow [37], among others [38, 39, 40, 41].

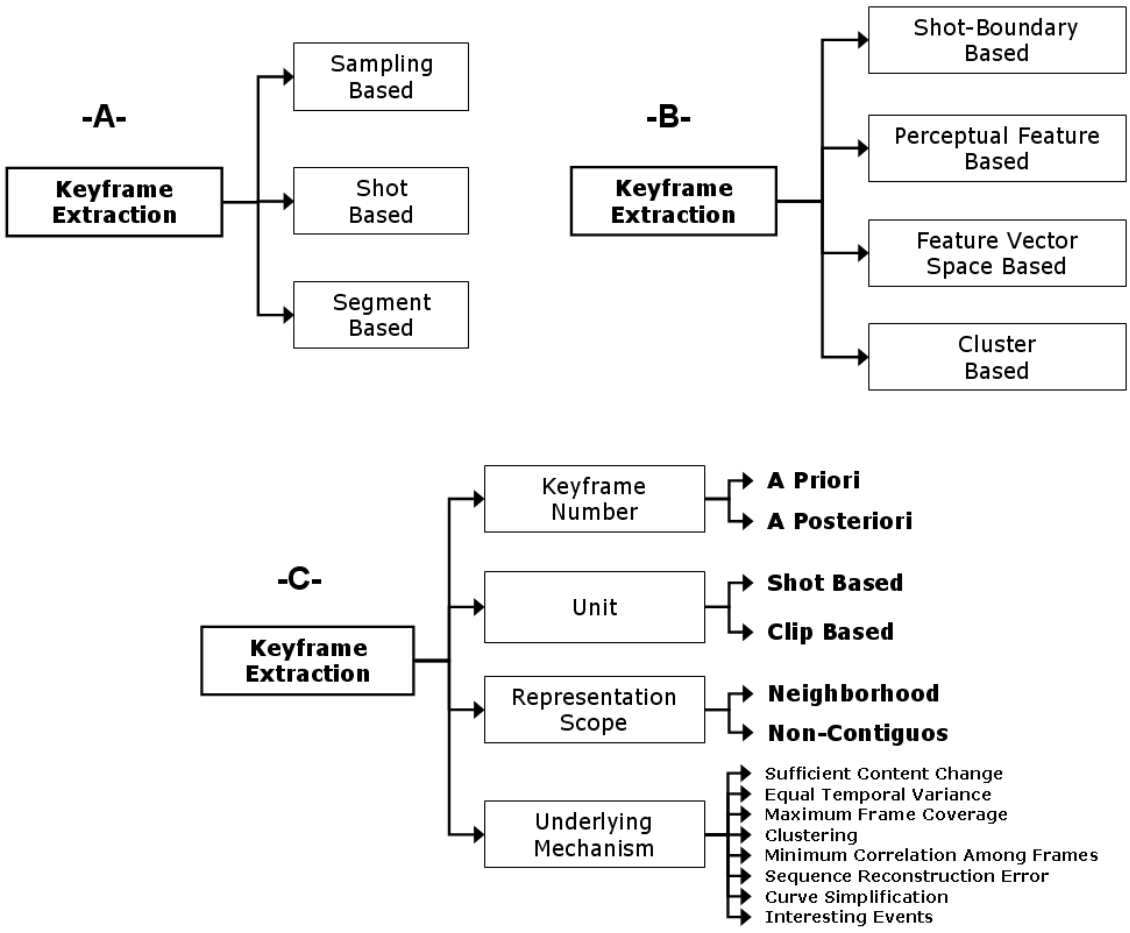


Figure 2.1: Keyframe Extraction Categorizations

- *Segment based*: groups approaches that make use of video units of higher level than a shot, which could contain for a example a scene or event composed by different shots. Authors include in this category global clustering approaches [42, 43] or scene based approaches like [44], where shots are clustered into scenes which are then rated and selected.

The classification provided in [25] considered four categories differentiated by the processing mechanisms applied for the selection of the keyframes (see figure 2.1 -B-):

- *Shot boundary-based:* Including approaches where the original videos are segmented in shots and one or more keyframes are extracted from each shot (analogous to the *Shot based* category described in [22] and [16]). One keyframe is selected from the beginning, middle or end of a shot [45] or from multiple positions according to changes within the shot [46].

- *Perceptual feature-based:* Including approaches where, after selecting a first key frame, subsequent keyframes are picked based on the comparison with the previously selected ones using visual perception features. Different types of features, like color [38], motion [47] or object-based approaches [35] (based on region descriptors in the images), are considered.

- *Feature vector space-based:* Including approaches were the original video frames are characterized as a set of feature vectors which constitute a curve in the space [48, 49]. Key frames are selected based on particularities of the feature curve (sharp corners, changes, etc.).
- *Cluster based:* Including approaches which aim to group data elements (in this case frames) according to their distance in a feature space for a subsequent selection of representative elements from the groups created (clusters) [50, 43].

According to [1], which provides a more exhaustive classification and includes several of the previously enumerated categories (see figure 2.1 -C-) , the keyframe extraction approaches can be considered attending to four possible aspects of the generating algorithms:

- *Keyframe number*: refers to the mechanism for determining the number of extracted keyframes, differentiating between approaches where the number of keyframes can be fixed a priori [39, 51], or decided by the abstraction process [52, 53].
- *Unit*: classifies the methods according to the temporal unit that each keyframe represents and differentiates between *shot-based* and *clip-based* approaches (which are analogous to the previously defined [16] and [22] categories *shot based* and *segment based*).
- *Representation scope*: differentiates between methods where a keyframe represents a neighborhood segment [54] or non contiguous segments of the videoclip [55, 40].
- *Underlying computational mechanism:* the keyframe extraction methods are differentiated in *'[...] eight somewhat overlapping classes´* [1] according to the applied processing mechanisms:
  - *Sufficient content change*: Selects keyframes as long as their visual content significantly differs from previous selected content [44, 56, 57, 58].
  - *Equal temporal variance*: A variant of the sufficient content change approaches, where the number of keyframes is set a priori [39, 51, 59].
  - *Maximum frame coverage*: Aims to maximize the number of frames represented by the selected keyframes [60, 61].
  - *Clustering*: As previously described, clustering approaches group frames in clusters according to their distance in a feature space. A selection process is then carried out with the generated set of clusters. [54, 62, 63, 64, 65].
  - *Minimum correlation among keyframes*: Methods included in this category aim to produce sets of keyframes with a minimal correlation between their elements [66, 67].
  - *Sequence reconstruction error*: This category refers to approaches which make use of a measurement of the capacity of the selected keyframes for reconstructing the original shot or sequence [68, 69, 70, 71].
  - *Curve simplification*: In this kind of approaches, frames are treated as points in a feature space connected by a curve. The abstraction process consists on the selection of those points which produce a smaller distortion in the shape of the curve [48, 72, 73].
  - *Interesting events*: In this case, the selection of keyframes relies on their 'interestingness' measured according to different criteria. Typical approaches for the measurement of interestingness are motion patterns [74, 75] or camera motion [76, 77].

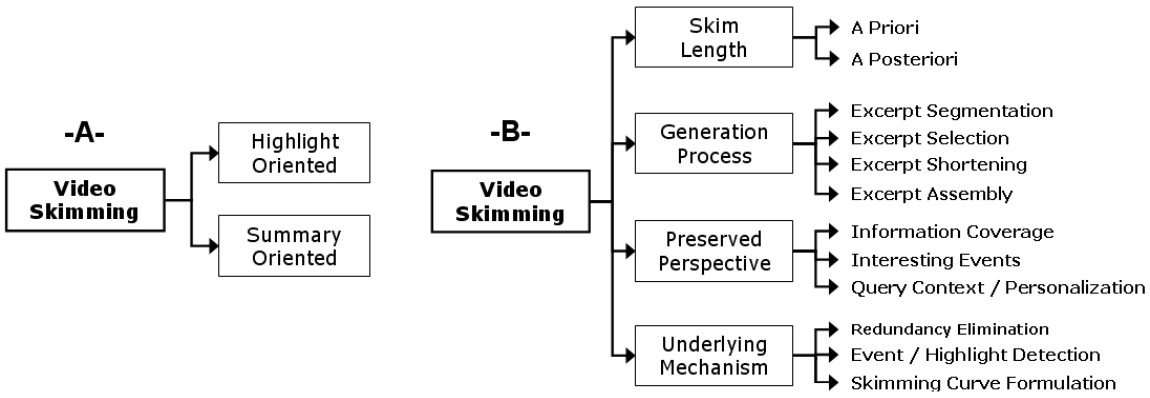


Figure 2.2: Video Skimming Categorizations

A completely different perspective for static images generation are the mosaic-based approaches (as defined in [22]) which, instead of extracting a set of keyframes, aim to represent the original content with a synthesized panoramic image composed by the fusion of different frames (or part of them) from the original content [78, 79]. However, most approaches keep the extracted keyframes unmodified, although several of them apply advanced presentation techniques, for example different kind of storyboards [80, 43], comic-like layouts [81] or slideshows [82].

## 2.3 Video Skimming

Video skimming consists in the extraction of several continuous video segments from the original video which can be later composed (edited) in different ways. In this case, it is considered that the temporal sequence of frames is preserved between the beginning and the end of each selected segment. A clear advantage of this method is to provide motion information and the possibility of including synchronized audio information. We will differentiate between two possible categorizations, shown in figure 2.2. The first one (figure 2.2 -A-), proposed in [16] and [25] presents two principal video skimming categories according to the segment selection criterion: *highlight oriented* and *summary oriented* approaches. On the other hand, [1] provides a more complex classification of video skimming approaches considering some additional aspects: *skim length*, *skim generation process*, *preserved perspective, underlying mechanisms* and *features used* (see figure 2.2 -B-, where the *features used* category has been omitted for its later description in section 2.4).

The two categories defined according to the first classification scheme [16, 25] (figure 2.2 -A-) are defined as follows:

- *Highlight oriented*: the output is composed by a set of relevant parts of the original content, as in the case of movie trailers or sport highlights summaries [74].

- *Summary oriented*: the output is composed by different segments which provide an overview of the whole original video [83]. This category is usually related to approaches where the abstraction process is treated as a global optimization problem. Clustering [50] and rate-distortion optimization methods [84] fall into this category.

Both categories are included within the more complex classification provided in [1], which differentiates between the following video skimming aspects (see figure 2.2 -B-):

- *Skim length*: As in the case of keyframes extraction, the video skim length can be defined a priori [19, 85] or determined by the content and skimming approach [7, 86].
- *Skim generation process*: This category is based on the different steps in which the original video excerpts (video segments) are processed.
  - *Excerpt segmentation*: Refers to the applied mechanism for the segmentation of the original video in separate units (without including shot segmentation techniques, usually considered as a previous process). Approaches include, for example, speech segmentation [87], application of interesting events [88, 89] or changes in the dominant motion [90].
  - *Excerpt selection*: Consists in the selection of the excerpts to be included in the video skim and can be based, among others, in clustering [3], event-based selection [89] or filtering based on extracted features [21].
  - *Excerpt shortening*: Corresponds to the reduction of the length of the original segmented excerpts. May be based, for example, on the selection of a predetermined portion of the excerpt [62, 3], selecting keyframes based on an attention curve and picking the surrounding segments [85], or selecting a portion of the excerpt which adequately represents the whole excerpt [91].
  - *Excerpt assembly*: Consists in the composition of the final video skim by taking the excerpts resulting from the previous steps. The most straightforward and common technique is to join the excerpts sequentially but there exist exceptions like joining segments with fades and wipes [92], alter the order of the excerpts [93] or compose advanced presentation layouts [94, 95].
- *Preserved perspective*: Refers to which aspects of the original video must be preserved in the video skim, differentiating between three categories:
  - *Information coverage*: Aims to generate a video skim able to represent the whole original video [50, 93, 96].
  - *Interesting events*: Also denoted as *video highlights*, represents interesting or important events in the video (according to the specific characteristics of the application). For example, goal scoring [97], high motion [85], applause and cheering [74], etc.
  - *Query context/personalization*: In this case, video skims are generated according to a set of user preferences or query. For example the selection of fragments which audio transcript corresponds to a user query [98], or the application of user-defined weights over a set of extracted features [99] for the selection of the excerpts.
- *Underlying mechanisms*: As in the case of keyframe extraction, the type of generated video skim depends on the applied underlying mechanism.
  - *Redundancy elimination*: Relies on the elimination of redundant content (according to the applied perspective on each specific approach). Such redundancy elimination can be achieved, for example, with clustering approaches [50, 62, 100] or sufficient content change approaches applied to video skimming [57].

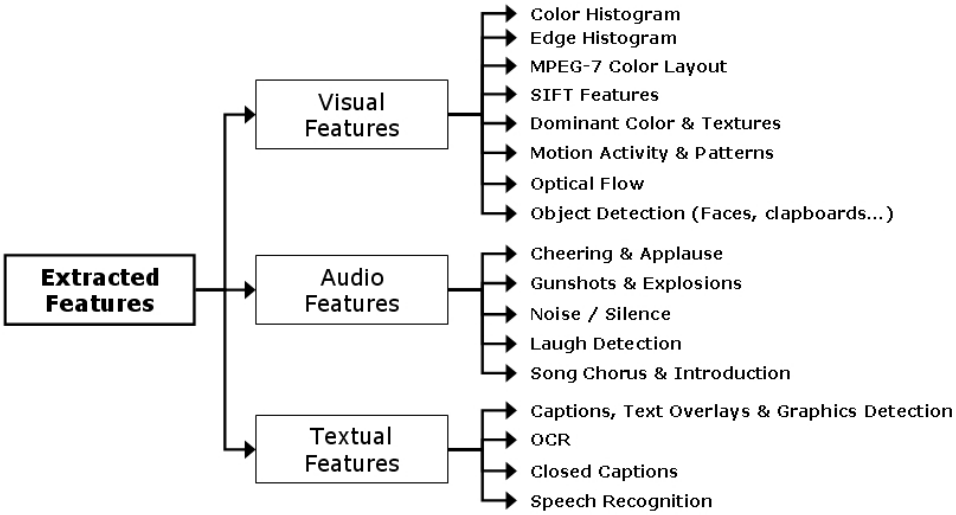


Figure 2.3: Commonly Extracted Features for Video Abstraction

  - *Event/highlight detection*: Video skimming approaches belonging to this category aim to preserve specific types of events that must be identified in the original video. Examples include different types of sports summarization systems like baseball [101], soccer [102] or different kind of events in surveillance recordings [103].
  - *Skimming curve formulation*: Those approaches compute a score curve with different values associated to the original video units according to an applied perspective. The video skim is generated by taking original fragments according to their score. Examples include, among others, skimming curve calculation based on motion activity and audio energy [8], motion and face detection combination [104] or original video annotations [105].

- *Features used:* In this category, video skimming approaches are classified according to the type of features extracted during the abstraction process. See section 2.4 for additional details.

## 2.4 Extracted Features

In the previous sections existing approaches for video keyframe extraction (section 2.2) and video skimming (section 2.3) have been overviewed. In this section, we will describe the different types of features that existing video abstraction approaches extract, combining generalized techniques with more specific ones. Although in some works (e.g. [1]) the techniques are differentiated between those applied for skimming or keyframe extraction, we consider that there is no difference between both approaches in terms of the extracted features. Every abstraction approach, depending on its application scenario or type of content, may rely on any of the enumerated features.

Similarly to other video abstraction surveys [1, 24] we will differentiate between three basic types of extracted features, depicted in figure 2.3:

- *Visual features:* visual features are, of course, of the major relevance for video summarization. Most approaches which rely on visual redundancy removal require at least a technique

for image or video segment comparison (usually applied for shot segmentation as well). Many of those works rely in the comparison by applying color histograms [57, 106, 107], MPEG-7 Color Layout [56, 96, 108], edge histograms [109, 110], SIFT features [111], dominant colors and textures[101, 89]. Image saliency calculation [4, 85] is applied as well for determining the relevance of the original video segments. Other techniques for selecting video segments rely on camera motion patterns [97, 102], optical flow [112], or motion activity [8, 104, 113]. Finally, some approaches include detection or analysis techniques for specific features in the video, for example faces [104, 82], clapboards [96, 114] or gestures [115].

- *Audio features:* Audio features are commonly used as well for video abstraction purposes, specially for the detection of specific conditions or situations, that is, events in the videos. For example, different sports highlight detection approaches aim to detect people cheering or applause for locating relevant footage [116, 89]. In [93] gun shots and explosions are detected for highlight selection in movies. Speech segments and their emphasis are detected in [117] for the identification of relevant content. In [118] the summarization conditions change according to silence or noise in the video. Other approaches summarize, for example, sitcoms based on laugh detection [119] or music videos [6] according to specific audio conditions (choruses, song introduction).

- *Textual features:* Text, extracted from different sources, can be a very useful tool for complementing other information sources in video abstraction approaches. Methods for extracting text include superimposed captions, text overlays and graphics [17, 93] extracted, for example, with OCR techniques [120, 121]. In other cases, textual information is extracted from closed captions included in the video stream [122, 123] or speech recognition [89, 124] techniques.

## 2.5 Conclusions

In this section, we have reviewed a number of existing video abstraction approaches following several classifications that exist in the literature. The classification categories proposed in such works are manifold, in tune with the wide variety of existing video abstraction approaches. The huge collection of existing techniques may be applied to many different types of content and application scenarios and the studied works usually categorize the abstraction approaches based on such applications or the techniques applied for the selection of the abstracts content.

After the overview of existing classification techniques and abstraction approaches, a video abstraction systems taxonomy and framework are proposed in the next chapter 3 for the characterization of abstraction approaches from an operational point of view, according to the objectives of this work.

# Chapter 3

# A Framework for Video Abstraction Systems

## 3.1 Introduction

This chapter presents a unified taxonomy and a generic architectural model aimed for the study of the characteristics and computational performance of existing abstraction systems. The taxonomy has been developed taking into account and identifying the operative characteristics of current state of the art video abstraction techniques. The proposed video abstraction architecture model characterizes the stages needed to build a generic abstraction process and establishes the basic architectural aspects and requirements for the modeling of systems with specific operative requirements.

A video abstraction systems taxonomy based on the operational aspects of the algorithms is firstly presented. Domain specific considerations such as particular content selection criteria, extracted features or selection mechanisms have been omitted and can be found in the taxonomies described in previous chapter 2. Once the taxonomy is presented, the chapter focuses on the definition of a common framework aimed to model the possible abstraction approaches in the taxonomy. There exist a high heterogeneity in the different approaches but most of them share conceptual stages which can be represented in a generic video abstraction architecture. The proposed taxonomy and architecture do not pretend to be an universal survey about video abstraction systems but to provide a common framework in which heterogeneous abstraction approaches could be compared and analyzed from an operational point of view. The work has been carried out studying the existing video abstraction methods and systems found in the literature and synthesizing their approaches together to generalize them into a unified model.

There exist works in the literature which depict and classify many of the existing abstraction approaches [1, 16] attending to different criteria and algorithm characteristics but there are few dealing with generic architectures. [24] presents a conceptual framework in which different categories of video abstraction are considered without dealing with specific stage definition. [93] presents an abstraction system in which three stages are roughly defined: video segmentation and analysis, clip selection and clip assembly. In addition there exist many specific approaches such as [10] which depicts a framework for sports video summarization (applied to a specific scenario but including several concepts that can be extrapolated to generic event or highlight oriented video abstraction) or many others [50, 85, 125] where different approaches are presented without trying to model the abstraction process in generic terms.

The rest of the chapter is organized as follows: section 3.2 depicts the proposed video abstraction taxonomy divided in external and internal characteristics. Section 3.3 explains the proposed abstraction system architecture. Examples of systems decomposition and the application of the proposed taxonomy and architecture to a real system can be found in Annex A. Finally, section 3.4 summarizes the obtained conclusions.

## 3.2 Abstraction Systems Taxonomy

The proposed taxonomy is organized at two different levels (namely, external and internal characteristics of the abstraction process) and can be used as a starting point for the understanding, modeling and development of efficient video abstraction architectures. It focuses on operational aspects and desired functionalities without dealing with specific algorithms for the different stages (alternative video abstraction systems taxonomies are described in chapter 2). Whilst the taxonomy has been developed considering the large number of existing references (see, for example, [16] and [1]), only a limited number of them are referenced in order to provide examples as this work is not aimed to be a survey of existing techniques.

### 3.2.1 Abstraction Process External Characteristics

This section characterizes the video abstraction techniques attending to their external properties: what kind of abstract is generated and what external interfaces and observable characteristics the abstraction system presents. Figure 3.1 shows the proposed classification of abstraction methods according to five external characteristics: *Output*, *Presentation*, *Size*, *Performance* and *Generation Delay*.

#### Output

The *Output* category refers to the kind of generated video abstract. From this point of view, abstraction approaches' output can be classified in:

- *Keyframes*: The output of the abstraction process is a set of still images that represent the original video according to a specific criterion.
- *Video Skims*: This approach consists on the extraction of several video fragments from the original video presented as a continuous sequence (even though presentation variations could be applied). The temporal order of frames between the beginning and end of each fragment is preserved but intermediate frames could be dropped.

Keyframe generation systems are very popular in the literature and allow very compact and fast-browsable representations of the original video. One of their main disadvantages is the absence of motion and audio information, desirable in certain applications. The usage of motion measures is a common approach for keyframe extraction: in [39] the incoming video is divided in segments of equal accumulated motion activity, selecting the middle frame of each obtained segment as keyframe. In [75] the points with motion acceleration or deceleration are selected as keyframes. Another typical approach is the selection of keyframes based on their visual dissimilarity, for example with clustering techniques [40]. The techniques for video skimming do not differ in many cases from those

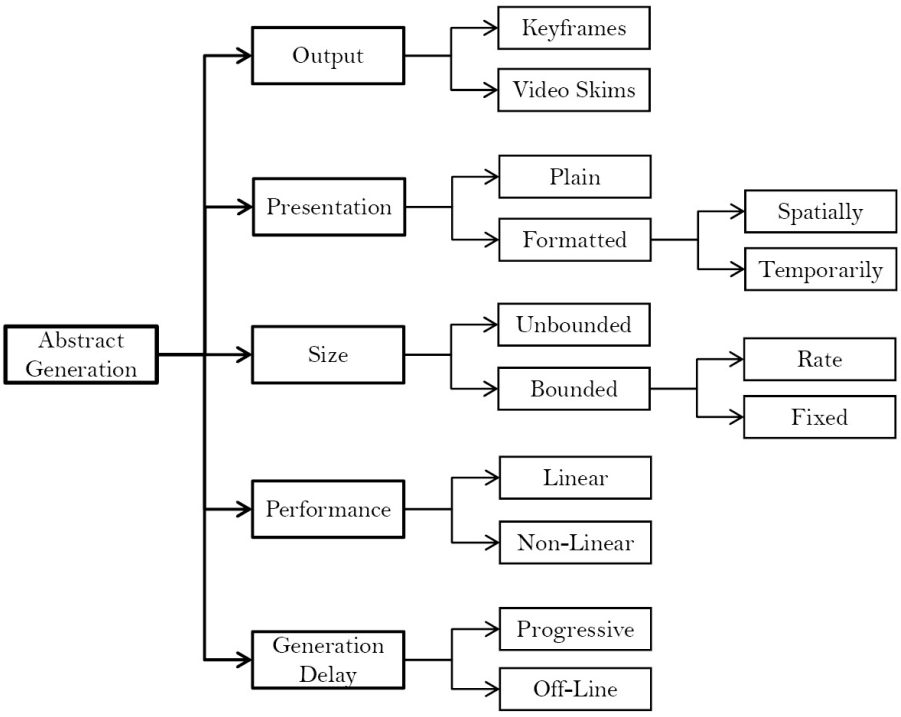


Figure 3.1: External Abstract Generation Characteristics

applied for keyframe extraction: [85] proposes a system which analyzes the incoming video and computes a saliency curve based on visual, audio and linguistic features. Given a target abstract length the system selects those fragments which maximize the relevance. The system is able to generate keyframes as well, just by selecting the intermediate frames from the selected video fragments (this skim/keyframe modality change can be easily extrapolated to almost any video abstraction system). [86] shows a completely different approach in which an original video is skimmed by analyzing each shot complexity to determine the minimal duration needed for its understanding and identifying scenes -groups of shots- for reducing their content.

**Presentation**

The *Presentation* category is related to the *Output* modality but is independent as any combination of *Output* and *Presentation* is, in principle, possible. Two modalities have been identified:

- *Plain*: The output, keyframes or video skims, are not formatted at all.
- *Formatted*: Includes abstraction mechanisms where the output format is modified in two possible ways: Spatially, implying spatial content variation or Temporarily, where the temporal dimension of the content is altered.

Most part of the existing abstraction systems present plain presentation, for example all the systems presented in the previous section for keyframe extraction [39, 75] or video skimming [85, 86]. *Formatted* presentations are usually applied for specific applications, higher condensation of the information or trying to provide pleasant interfaces. One typical spatial presentation is the storyboard,

where the extracted keyframes are presented in a equally distributed canvas, as in the case of [80] for news abstraction. Video Manga [43] displays a comic-like output with different size areas for each keyframe. The assigned area depends on a relevance measure according to the keyframe shot length and redundancy. Spatial formatting is present on video skimming systems as well: [94] and [95], presented to the TRECVid 2007 BBC rushes summarization task, display multiple simultaneous video playing areas aiming to maximize the amount of information included in the output. Temporally formatted outputs involve modifications in the original content temporal dimension, for example slide-show approaches like [82], where selected keyframes are presented with an assigned fixed time or fast forward approaches where the original video content is accelerated [27].

**Size**

The *Size* external characteristic refers to the output summary size, defined as the number of extracted keyframes or number of frames if considering a video skim. Attending to this criterion abstraction methods can be grouped in two modalities:

- *Bounded*: The abstract size is defined as a fixed value or as a ratio of the original size: if the size of the original video is a priori known, both modalities are equivalent.
- *Unbounded*: The size of the output summary depends only on the abstraction process, its configuration parameters and the original video content.

The abstraction systems output size type mainly relies on the kind of application or underlying abstraction algorithms. In certain situations, such as a limited space presentation layout, a fixed number of keyframes could be required. The disadvantage of this method is the possible loss of representative or relevant events (attending to some criterion) if the target output size is too limited, or the inclusion of redundant information if it is too large. An example can be found in [84] where a predefined number of keyframes minimizing the distortion with respect to the original video are extracted. Nevertheless the same approach is able to deal with *unbounded* output size, selecting keyframes until a specific distortion value is reached. [57] aims to generate a specific original size ratio length video skim and presents two steps: in the first one, original video segments are appended to the skim if no similar fragments are already selected obtaining an undefined length output. This output size is reduced in the second step, if needed, by dropping the most redundant fragments until the output length is under the defined target size ratio. In *unbounded* size systems, the output abstract size is unknown until the end of the abstraction process like in the first step of [57]. Many early abstraction methods based on content variation such as [58], which outputs a keyframe per each detected scene change, or [75], where the keyframes are extracted in points where motion acceleration or deceleration are detected, work without a priori defined abstract length. Clustering approaches may work in unbounded modality if the number of clusters is not predefined, like in [50] where the centroid of every obtained cluster is selected as keyframe.

**Performance**

The Performance category refers to the amount of processing needed to complete the abstraction process in the sense of the algorithm's computational complexity and is one of the main aspects of an abstraction process to be taken into account for the characterization of *on-line* systems. Two main modalities can be defined:

| CATEGORY | MODALITY | | REFERENCES |
|---|---|---|---|
| Output | Keyframes | | [85, 75, 39, 40, 80] |
| | Video Skims | | [85, 86, 94, 95] |
| Presentation | Plain | | [85, 75, 39, 86] |
| | Formatted | Spatially | [80, 43, 94, 95] |
| | | Temporarily | [94, 95, 82, 27] |
| Size | Unbounded | | [50, 75, 58] |
| | Bounded | Rate | [57, 127] |
| | | Fixed | [39, 84] |
| Performance | Linear | | [39, 27, 57, 58, 52, 64] |
| | Non Linear | | [50, 40, 60, 67, 126, 100] |
| Generation Delay | Progressive | | [75, 39, 27, 57, 58, 128, 64] |
| | Off-line | | [50, 40, 84, 100, 72] |

Table 3.1: External Characteristics Abstract System Examples Classification

- *Linear*: The amount of processing resources needed by the abstraction algorithm scales proportionally with respect to the original video length.
- *Non-Linear*: Include video abstraction techniques which require computationally costly algorithms which do not scale linearly and, in consequence, are commonly applied only in *off-line* scenarios.

In most of the cases the abstraction processes with linear complexity are those that perform local optimization or selection of the original video fragments maintaining a constant analysis and selection complexity (as it will be described in chapter 4, *linear* performance is one of the requirements of *on-line* abstraction systems). Many abstraction approaches rely on visual redundancy elimination and, in those cases, costly image and video fragment comparisons must be carried out. If those comparisons are avoided or reduced the abstraction systems are more likely to perform linearly. Straightforward solutions like selecting the first frame of each shot [58], video subsampling [64] or more complex systems where the number of comparisons are applied only to surrounding frames [52] or a limited amount of preceding video fragments [57] fall into the *linear* performance category (if no other *non-linear* component is integrated in the system). On the other hand, methods dealing with the abstraction problem as an optimization problem [60, 67], maximization of an objective function [126] or clustering based approaches [100] make use of the whole available original content for the abstract generation and require a number of comparisons which heavily increases with respect to the amount of original information, yielding to *non-linear* performance.

**Generation Delay**

The generation delay category is defined as the latency between the beginning of the video processing (the instant when video frames are read or received) and the instant when the abstract output starts to be generated. Two modalities have been defined:

- *Progressive*: Do not require the complete original video available in order to start the abstract output.

- *Off-Line*: The abstract generation does not begin until the complete original content is available and analyzed.

Most subsampling based methods such as fast-forward approaches [27, 64] or systems where one keyframe is selected from each incoming shot [58] or group of frames (e.g. in [39] keyframes are extracted from each video segment accumulating a predefined amount of variation) are able to generate the output progressively. Other more complex methods such as sufficient content change based approaches [57], where video segments are added to the output if no visually similar fragments are already included, are able to generate a progressive output. This category includes other approaches such as [75] with a progressive analysis for the identification of motion acceleration or deceleration points as keyframes or methods based on local analysis of a feature curve extracted from the original video [128]. *Progressive* abstract generation is one of the requirements of *on-line* abstraction approaches (see chapter 4 for the definition of *on-line* abstraction systems).

The *off-line* operation mode is typical of systems which apply an algorithm requiring the complete original data for the abstract generation: clustering approaches [50, 40, 100], the previously commented rate-distortion approach [84] or other methods such as [72] where the complete original video is mapped to a polyline which is later simplified for the generation of the video abstract.

#### External Characteristics Taxonomy

Table 3.1 summarizes the previously commented representative examples of each of the categories in the External Characteristics Taxonomy. Some algorithms can appear in several modalities within the same main category due to their flexibility.

It can be observed that *Performance* and *Generation Delay* are closely related categories: a progressive generation system requires local processing of the original video implying a reduction of the information to be processed by the algorithms. Therefore, such methods are more likely to perform linearly. On the other hand, *off-line* approaches making use of the complete input video, usually present lower performance due to the great amount of information to deal with. For characteristics such as the *Size* it is easier for an *off-line* and iterative method to reach a specific output abstract size than for progressive methods which do not have complete video information available and, in many cases, can not change the already selected fragments.

### 3.2.2 Abstraction Process Internal Characteristics

This section presents the proposed classification with respect to the internal mechanisms applied in the abstraction process. Figure 3.2 shows the different categories proposed for the internal characteristics: *Basic Unit*, *Analysis*, and *Scoring & Selection*.

#### Basic Unit

The Basic Unit *(*BU*)* category refers to the kind of processing unit used for reading, analyzing and selecting within the abstraction process. The frame is the minimal and indivisible visual unit and therefore the BUs will be composed by one or more frames. How the number of frames in a BU is selected can be based on fixed values, visual characteristics, or any other approach like, for example, segments of homogeneous audio. Two BU modalities are considered:

- *Fixed Size*: Includes approaches making use of fixed size BUs being the single frame the most typical kind of BU.

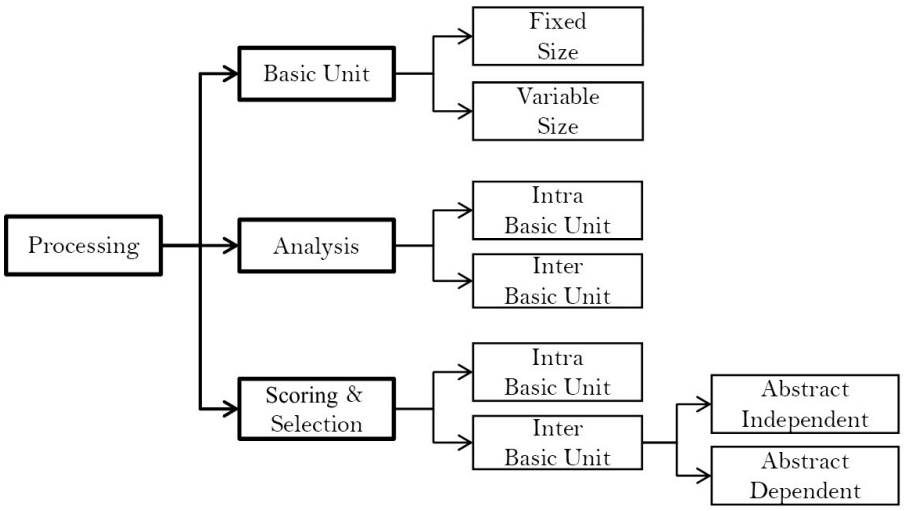


Figure 3.2: Internal Abstract Generation Characteristics

- *Variable Size*: The BUs have a variable length, for example approaches making use of the shot (variable length) as basic unit.

Single frame based methods, such as subsampling approaches [27, 64], make use of single-frame BUs as basic unit. In [39] motion activity is extracted and accumulated per frame. [66] represents an example where a clustering process is carried out with a set of features extracted in a per-frame basis. The usage of fixed size block of frames -BoF- is less common but can be found for example in [104], where the original video is split in fixed size blocks which are then selected based on a set of extracted features (e.g. motion activity, face presence) or [129], which analyzes a priority curve associated to the original video looking for peaks and is able to work in both fixed size (BoFs) or variable size (shots) BUs. Numerous approaches make use of shot segmentation methods, resulting on variable size BUs (shots): in [130] an initial division of the content in shots is carried out and then such shots are processed by analysis algorithms, compared and finally selected or discarded. Other methods make use of varying size BUs, for example in [57] where original video is split in fixed-size fragments unless a shot change is detected within a fragment which is, in this case, sub-partitioned. It is possible to deal with different types of BUs within a single abstraction system: different stages or included algorithms may require different modalities. For example the algorithm proposed in [131], an automatic video editing system, works at both shot, performing visual shot boundary detection and sub-shot levels, dividing each shot taking into account audio analysis.

The type of applied BUs can have impact in other abstraction process characteristics. The computational performance of a two shot comparison mechanism may vary if only two representative keyframes, one from each shot, are used or if a complete frame-by-frame shot comparison is carried out. In the first case, the computational cost is usually constant while, in the second case, the computational complexity will grow quadratically with the shot length. On the other hand, comparison results, and hence the abstraction result, will be necessarily affected if only one representative keyframe or the whole shots are compared.

**Analysis**

The *Analysis* category refers to the scope of application of the analysis algorithms within the abstraction process, considering an analysis algorithm as any kind of method used for the extraction of features (e.g., statistics, MPEG-7 descriptions, object detection) from the original video. In general terms, an analysis algorithm provides information which would be later used for the selection of BUs to be included in the output abstract. Two different modalities are considered:

- *Intra Basic Unit*: Methods in which the feature extraction is performed on single BUs.
- *Inter Basic Unit*: Involves the usage of two or more BUs in the feature extraction process.

*Intra-BU* analysis methods are those in which the extraction of features depends on individual BUs. This kind of systems are more likely to be able to perform progressively because there is no need to have all the original video BUs available for their analysis (nevertheless other steps in the abstraction system could limit the overall performance). The single frame analysis (*intra-BU*) is very common in abstraction approaches, for example the extraction of color histograms usually applied for frame comparison [57] or face detection, which is applied in [82] for the calculation of segments relevance (in many approaches video segments are considered as more relevant if faces are detected on them). An example of *intra-BU* analysis group of frames -GoF- based BUs can be found in [104], where each GoF motion activity is measured. In [127] the audio energy for each 5ms audio fragment is extracted for the detection of clapboards in BBC unedited content (in this case the small audio fragments can be considered as fixed size BUs). Regarding *inter-BU* analysis it is possible to find single frame BU approaches where the extracted feature is the distance between frames: in [58] the distance between the color histogram of a given frame and an average histogram of previous frames (BUs) is calculated for the identification of shot boundaries. For BUs composed by more than one frame, it is possible to find cases in which feature extraction is performed making use of several BUs, for example shot similarity metrics applied for redundancy reduction [57, 132]. *Inter-BU* approaches would, in many cases, require a higher amount of the original video information available (and hence higher memory consumption) and could constrain some of the abstraction system external characteristics.

**Scoring and Selection**

The *Scoring & Selection* category refers to the different ways in which those steps can be applied to the original video BUs. As in the *Analysis* category, two modalities are defined:

- *Intra Basic Unit*: The scoring or decision about the inclusion of a given BU in the output abstract is based only on intrinsic BU characteristics. This is compatible with both *Intra-BU* or *Inter-BU* analysis for BU annotation.
- *Inter Basic Unit*: The selection of a given BU takes into account its intrinsic features as well as those associated to other BUs. If the selection does not depend on other BUs already included in the abstract, the system will be categorized as *Abstract Independent* or *Abstract Dependent* otherwise.

Highlight abstraction methods, which include in the output BUs fulfilling a set of predefined conditions (e.g., presence of face, motion activity over a threshold), fall into this category. An example is [74] where relevant sport events are identified and selected for the abstract taking into account motion activity and audio features. An *Inter-BU Abstract Independent* approach can be found in [126]

| CATEGORY | MODALITY | | REFERENCES |
|---|---|---|---|
| Basic Unit | Fixed Size | | [39, 27, 66, 104, 64, 129] |
| | Variable Size | | [133, 57, 131, 130, 129] |
| Analysis | Intra Basic Unit | | [82, 57, 127] |
| | Inter Basic Unit | | [57, 58, 132] |
| Scoring & Selection | Intra Basic Unit | | [52, 74] |
| | Inter Basic Unit | Abstract Independent | [126, 75] |
| | | Abstract Dependent | [39, 57] |

Table 3.2: Internal Characteristics Abstract System Examples Classification

where priority, continuity and non-repetition criteria must be fulfilled by the generated abstract and must be calculated taking into account different combinations of BUs. In [75], classified in the same category, abstracts are composed as a subset of the original content maximizing a 'perceived motion energy' function and, for this reason, the selection of a given BU depends on the score of others. The *Abstract Dependent* categorization groups methods which take into account BUs previously selected for its inclusion in the generated abstract. For example methods based in the maximization of the included content coverage avoid the inclusion of BUs too similar to other already included in the abstract [39, 57].

**Internal Characteristics Taxonomy**

Without providing an exhaustive classification of each method according to each explained category, Table 3.2 summarizes a number of representative examples of each of the categories in the Internal Characteristics Taxonomy. As in the External Characteristics Taxonomy case, there are examples which appear on several of the existing categories while, in other cases, it is possible to classify an abstraction algorithm in several modalities within the same main category. In opposition to the External Characteristics Taxonomy, here there are not clear relationships between the different internal categories so arbitrary combinations can be found on existing methods.

## 3.3 Architectural Models for Video Abstraction

The aim of the proposed architecture is to provide a modular, as simpler as possible, solution which could allow to fit inside most part of the (current and future) existing video abstraction approaches and to identify the different steps required to complete the abstract generation. It has been developed taking into account the previously depicted video taxonomy and it is compatible with the defined concepts and categories. Furthermore, a relationship between an abstraction approach classification and its corresponding modeling can be stated in many cases.

Although the proposed approach may not be the most practical or natural implementation for several abstraction algorithms, it provides a common conceptual division that will allow the study and comparison of different abstraction processes in terms of functionalities and performance. The studied abstraction algorithms and their possibilities and limitations will be defined by the way in which they can be mapped into the proposed architecture.

The abstraction process is modeled as a chain of independent stages through which the video BUs (as defined in section 3.2.2) "travel trough" while being analyzed, compared and selected. The

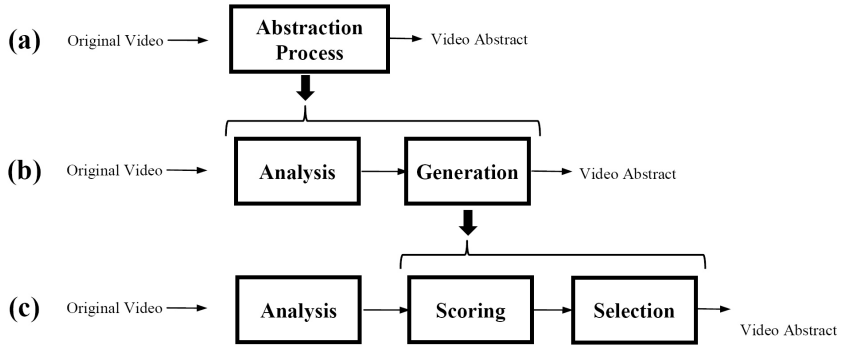


Figure 3.3: Abstraction Stages Decomposition

defined stages will be considered as independent modules but it will be allowed to share information and metadata about the abstract generation and processed BUs among them. Such information will be useful for guiding the different stages processing or for allowing the modeling of complex scoring mechanisms.

The proposed abstraction model will ease the generic study of abstraction mechanisms and the restrictions required for building systems with specific external characteristics (see section 3.2.1). It is aimed as well to ease the understanding of the dependencies between different stages of an abstraction approach, to improve the algorithms performance or even to identify potential parallelizations in the process.

### 3.3.1 Simplified Functional Architecture

Any abstraction process can be considered as a black box where an original video is processed to produce a video summary (see Figure 3.3 (a)). Every abstraction algorithm may be described in this way: a process which receives an original video input and outputs a video abstract. Nevertheless this model has not utility for abstraction mechanisms analysis and a more detailed model is required.

The first significative characterization of the abstraction architecture is shown in Figure 3.3 (b) where the abstraction process is divided in two stages: 'analysis' and 'generation'. The 'analysis' stage will be in charge of extracting relevant features from the original video that will be taken into consideration for the production of the final video abstract in the 'generation' stage, where any possible application of the extracted information will be carried out (including BU comparison , ranking and selection).

The 'generation' stage is, in many cases, the most complex one and, for this reason, a further conceptual division will be considered. Figure 3.3 (c) shows the division of the 'generation' stage in 'scoring' and 'selection'. The 'scoring' stage will be in charge of providing a score or rank for each original video BU. Such score is not necessarily defined as a single numeric value but as a combination of an arbitrary number of numeric values, tags, classifications, etc. On the other hand the 'selection' stage will decide which of the incoming BUs must be included in the video abstract based on their associated score. The complexity balance between the two stages depends on the specific abstraction algorithm considered. There are abstraction mechanisms suitable to be modeled with a very simple scoring mechanism followed by complex selection algorithms (for example [126] where different possible combinations of abstracts are evaluated considering a complex score based on priority, con-

tinuity and redundancy criteria) or abstraction mechanisms where the selection stage can be reduced to a basic score thresholding (for example [57] where the visual similarity of each incoming BU is calculated with respect to previously selected BUs and those with a dissimilarity value over a predefined threshold are included in the abstract).

It is straightforward to demonstrate that any abstraction technique fits inside the proposed model by encapsulating all the complexity in the 'scoring' stage which would tag each BU to be included in the final abstract as '1' or '0' otherwise. The selection module will just drop those BUs rated as '0' and will write in the output abstract those with score '1'. In this way it would be possible to fit any abstraction modality inside the analysis-scoring-selection model although a balance between the different stages would be usually possible and desirable.

The defined functional modules can be considered as a minimal set of stages needed for a generic enough abstraction architecture: there is no need for considering all of them in the design of a working abstraction system, being the 'selection' stage the only mandatory one (a minimal abstraction system can be built with a single selection stage in which subsampling [64] or random selection of BUs is performed). Nevertheless most of the existing abstraction approaches can be modeled with this 'analysis'-'scoring'-'selection' stages approach.

### 3.3.2 Abstraction Systems Modeling

A basic stage-based functional architecture for video abstraction systems has been depicted in the previous section. The abstraction process is considered as a flow of independent BUs through the defined stages which ends when the complete set of original video BUs has leaved the system. The different abstraction stages, type of BUs, how those BUs are processed, the time needed to complete the process and other considerations vary depending on each abstraction approach and will be determined by the external and internal system characteristics (previously defined in section 3.2). In this section different abstraction approaches are taken into account for the identification of the different components and data flows that should added for completing the proposed stage architecture.

The different approaches are grouped in non-iterative systems, that is, systems where the BUs are processed at the most one time per stage, and iterative systems, where the BUs can be resent to the 'scoring' stage after being processed by the 'selection' stage.

#### Non-iterative Video Abstraction Systems

Figure 3.4 (a) depicts the most simple abstraction architecture possible. As it is based on a simple subsampling mechanism all the needed algorithms are included in a single 'selection' stage in charge of picking 1 out of every $n$ BUs (e.g. just by direct subsampling or random selection). The 'User Preferences' which, in this case, will be limited to the selection of the output abstract rate $\frac{1}{n}$ guide the process. This kind of systems are able to generate progressive, bounded size abstracts with negligible delay and linear performance, as no analysis over the original content is needed and BUs can be immediately selected or discarded. An example of this architecture can be found in [64] which depicts one system in which the abstract generation consist on simple speed-up of the original video carried out by uniform frame subsampling.

With the same architecture and considering different kind of input and output BUs, keyframe or video skims could be generated and formatted in different ways if a 'presentation' stage (see section 3.3.3) is appended to the system; this consideration can be generalized for any of the abstraction systems that will be described next.

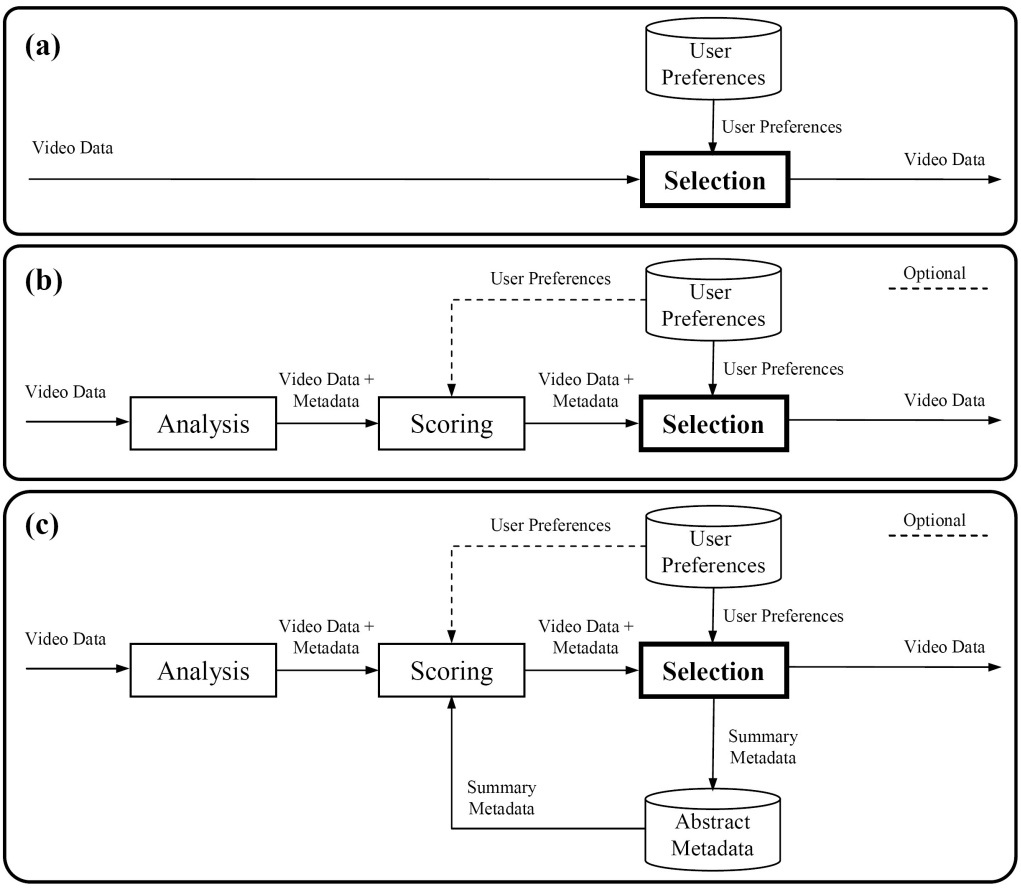


Figure 3.4: Non-iterative Abstraction Architecture Examples

Figure 3.4 (b) shows an architecture including the 'analysis' and 'scoring' stages. In this case the BUs scoring depends only on the original content and it is based on the results of the analysis stage and the 'User Preferences' which could specify, for example, the desired abstract length or be used in the 'scoring' stage for personalization purposes. One kind of abstraction systems belonging to this architecture are adaptive subsampling abstraction systems where a varying amount of original BUs are selected based on the analysis results. In [28] a single frame BU system is presented where the BU selection rate is proportional to the visual activity in the video. The system described in [39] generates keyframes by non-uniform sampling based on motion activity. It can be modeled with an initial 'analysis' stage in charge of measuring the incoming BUs motion activity. The 'scoring' stage should rate each incoming BU with the accumulated motion activity for each frame composing the BU. Finally, the 'selection' stage performs a selection of $n$ frames from each shot based on its accumulated motion activity value. The model can be generalized for any adaptive sampling mechanism by considering different extracted features and scoring approaches.

Relevance curve-based abstraction systems can be included in this category as well. In those systems a relevance curve is generated applying different criteria and such curve is later taken into account for the selection of the final output abstract. In [104] the relevance curve is generated depending on the video activity and face detection and the blocks of frames with a relevance value over

a predefined threshold are selected. Another example can be found in [129], where soccer videos are split into blocks and annotated with concepts such as goal, red card, etc., each one with different assigned priority. The generated curve is then analyzed identifying peaks, eliminating irrelevant blocks and merging relevant ones. The final set of selected blocks is then reduced in length to fit in an specified output abstract size. In both systems the 'scoring' stage is in charge of combining the features extracted in the 'analysis' stage generating a relevance/priority curve. The 'selection' stage selects a limited size subset of the incoming BUs which maximizes the accumulated relevance. Other considerations can be taken into account in the process, for example user preferences for relevance curve generation or continuity in the set of selected BUs for the generation of abstracts with pleasant rhythm. The relevance curve model is a generalization which covers many abstraction approaches due to the arbitrary criteria applicable in the curve generation.

Those kind of abstraction mechanisms, adaptive subsampling or relevance-curve based, where the scoring of each BU is independent from other BUs or have a limited dependency, are particularly suitable for *progressive* abstract generation (defined in section 3.2.1): BUs can leave the system even when other BUs have not passed trough all stages yet. Nevertheless the model presented in 3.4 (b) is also suitable for other abstraction systems in which this condition is not fulfilled, for example clustering based approaches [50, 65]. In a typical clustering process a first stage of data analysis is carried out extracting features applicable in a subsequent clustering process. Those two stages are clearly identified with the 'analysis' -feature extraction- and 'scoring' stage in which each BU is tagged with a cluster number and sometimes scored (for example, with its distance to the cluster centroid). The further 'selection' stage will be in charge of selecting the final set of BUs from the calculated clusters (for example selecting the closest BUs to each cluster centroid [55]). In those cases the whole set of original video BUs is needed to complete the clustering process in the 'scoring' stage and, therefore, the approach is not suitable for *progressive* abstract generation (no BU leaves the 'scoring' stage until all of them have been scored).

Figure 3.4 (c) shows a variation of the architecture in which an 'Abstract Metadata' database has been enabled. This database is a representation of any possible information feedback mechanism between the 'selection' and 'scoring' stages needed by certain approaches. For example, sufficient content change (as called in [1]) abstraction approaches include BUs in the output abstract only if their visual difference with previously selected keyframes is significant. In those cases the 'analysis' stage extracts visual features from the original content such as color histograms [57] or the MPEG-7 Color Layout descriptor [56] which are used in the 'scoring' stage to compare and rate the BUs attending to its similarity to already selected BUs. As the BUs selected to be part of the abstract leave the system their visual features must be kept in the 'Abstract Metadata' repository for its usage in the rest of the abstraction process.

#### Iterative Video Abstraction Systems

The models shown in figure 3.4 are valid for abstraction systems where the incoming BUs are processed only once by each defined stage and allow the modeling of most of the existing abstraction approaches. Nevertheless, there are abstraction systems based on iterative processing which require a more complex solution: in [71] an initial set of keyframes is sequentially selected from the original video. The position of such keyframes is then iteratively refined reducing the abstract distortion until a specific value is reached. [60] presents a maximum frame coverage abstraction approach which aims to generate a video abstract selecting a set of BUs as most representative as possible of the orig-

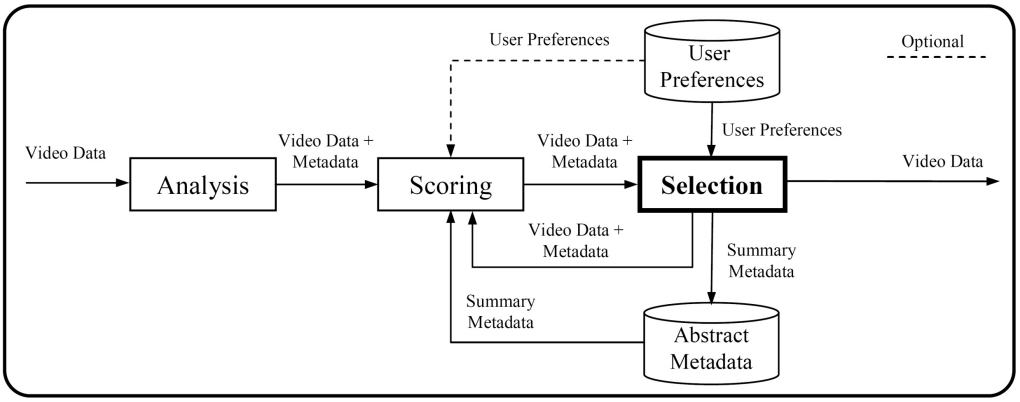


Figure 3.5: Iterative Abstraction Architecture

inal content. In this case, each time a new frame is selected for the abstract a recalculation of each frame coverage is carried out. Figure 3.5 architecture includes an additional video data (BUs) and metadata flow from the 'selection' to the 'scoring' stage, allowing the iterative scoring of the BUs. In [60] the coverage of each original frame is calculated as the set of all frames in the original video which are considered as similar to the given one. The most representative frame (the one with higher number of similar frames) is iteratively selected to be included in the output abstract. On each iteration the the set of similar frames to the selected one are removed from the remaining selectable frames and the most representative frame must be recalculated. This scheme can be modeled with the proposed architecture as a frame-BU system with an 'analysis' stage where visual features are extracted (e.g. color histograms, color layout), a 'scoring' stage where the coverage value (i.e. the set of similar BUs to each given one) is calculated and a 'selection' stage where the BUs for the video abstract are iteratively selected as those with the highest coverage values. On each iteration a BU is selected and some others -those similar to the selected one-, considered as already 'represented', are discarded. All the BUs which have not been selected to be part of the output abstract nor eliminated from the system are sent back to the scoring stage (via the defined feedback data flow) for a recalculation of their coverage value; this process is repeated until there are no more BUs available in the system. This example demonstrates the need of a feedback data flow between the 'selection' and 'scoring' for those systems in which the original scores associated to the incoming BUs must be periodically recalculated.

**Abstraction Systems Architecture Summary**

Table 3.3 summarizes the architectural categories divided in Iterative -I- or Not Iterative -NI- and depicting the set of abstraction modules which are included: Analysis -A-, Scoring -Sc-, Selection -Sel- and the inclusion of a Metadata Feedback -MF- data flow. It is not surprising to find a lack of references corresponding to the [NI,Sel] category due to its simplicity (e.g. simple subsampling or selection of the beginning of the original video) while more approaches can be easily found in other categories. The computational complexity of every abstraction system will depend on each abstraction stage internal complexity as well as the system architectural category.

Other abstraction systems, implementable with different combinations of abstraction modules and data flows, may exist in the literature: the different presented combinations have been selected because they are considered as representative models.

| ARCHITECTURAL CATEGORY | REFERENCES |
|---|---|
| [NI,Sel] | [64] |
| [NI,A,Sc,Sel] | [50, 39, 104, 28, 129, 65, 55] |
| [NI,A,Sc,Sel,MF] | [57, 56] |
| [I, A, Sc, Sel, MF] | [60, 71] |

Table 3.3: Abstraction System Architectural Classification

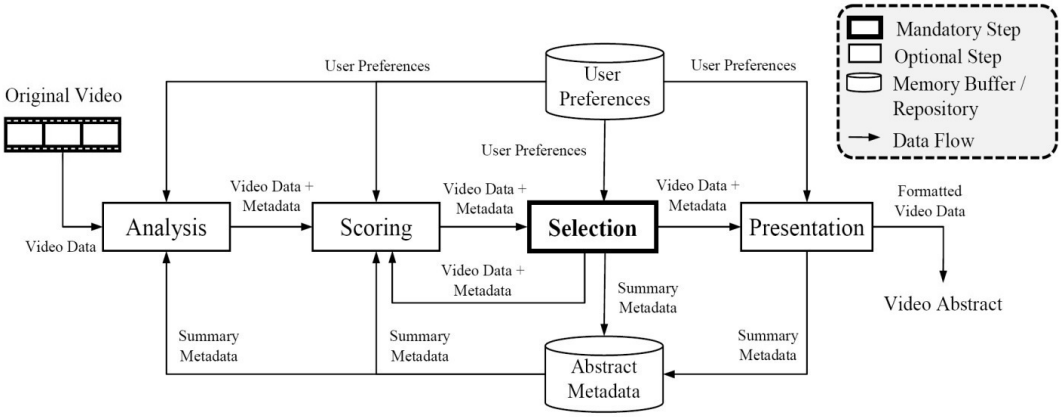


Figure 3.6: Generic Abstraction Architecture

### 3.3.3 Generic Video Abstraction Architecture

Figure 3.6 depicts the final generic abstraction architecture following the analysis-scoring-selection stages model. The data flow between the different stages and the repositories/databases for metadata storage are shown in the architecture. Such repositories represent information storage or interchange with independence of the mechanisms applied for its implementation (direct memory access, disk storage, databases, etc.).

The 'selection' stage is mandatory in any abstraction architecture because a mechanism to output part of the original video content must be included in any abstraction system. The rest of the stages -'analysis', 'scoring' and 'presentation'- are not mandatory: for example, in the case of a video abstraction system based on a original video subsampling, there is no need to analyze or rate the original content. A 'presentation' step has been included in the generic architecture providing coverage to those abstraction approaches in which video editing or formatting is needed. This stage is rarely present in the studied abstraction systems and has neither impact in the BUs selection methods nor in the overall system efficiency. It has been considered as a element which can be appended to the system and would have an impact in the users perception about the generated abstracts but without relevance in terms of systems characterization.

The abstraction process will be considered as the flow of the available BUs through the depicted stages until all of them have left the system (being included in the output abstract or discarded). All the processing stages receive BUs which can be accumulated, processed (analyzed, annotated, rated, combined, etc.), redirected to other stage, selected or discarded. Any stage can produce metadata (e.g. low-level video analysis results, mid-level features, semantic annotations, tags, etc.) that can be appended to the BUs and/or stored in the 'Abstract Metadata' repository making them available for

its usage by other stages.

The video an metadata feedback flow displayed in Figure 3.6 enables iterative abstraction systems (see section 3.3.2) where the selection or discard of a specific BU as part of the output abstract yields to the recalculation of other BUs scores.

The 'User Preferences' repository makes available user preferences or system configurations for cases in which this information can be applied for customized abstract generation. For example, the user could define abstract characteristics such as its length (to be considered in the 'selection' stage), modality, or media format for the output abstract (to be considered by the 'presentation' stage). The definition of what categories of content (e.g., economy, sports, weather forecast) the user is interested in could be used in the 'selection' stage as a filtering constraint or in the 'analysis' stage, applying different analysis algorithms depending on the user preferences (for example the application of a face detector algorithm only if the user specifies any preference about faces inclusion in the abstract).

#### Composed Abstraction Systems

The architecture depicted in previous sections enables the modeling of arbitrary abstraction approaches by the encapsulation of their algorithms in the proposed conceptual stages. Nevertheless, there are cases where an abstraction system may be difficult to map to such stages or the reached solution is not intuitive due to the system complexity. In such cases, a decomposition of the system as a combination of two or more 'simple' abstraction approaches can ease the system modeling. We will define an abstraction subsystem as a set of the techniques or algorithms applied within a given abstraction approach that could be isolated from the rest of the system and modeled with the proposed 'analysis'-'scoring'-'selection' stages. Most part of the studied approaches can be modeled with a single subsystem but several cases may be more clearly modeled if presented as a combination of several subsystems. Figure 3.7 shows two basic composed architectures. Figure 3.7 (A) shows a serial abstraction architecture where the overall abstraction process is modeled as a concatenation of subsystems in which the output of each subsystem serve as input for the following one.

An example can be found in [57], a video skimming approach later described in chapter 5, where the abstraction process can be separated in two independent subsystems. The first one consists in a sufficient content change approach where the incoming video shots are split in fixed size blocks. The analysis stage extracts each frame color histogram and the scoring stage compares every incoming video segment with previously selected ones (the first one is automatically included in the abstract) obtaining a dissimilarity measure. In the selection stage those fragments with a dissimilarity value over a predefined threshold are selected for the output abstract. The described steps can be considered as a complete *on-line* abstraction approach (that is, it is able to process the original video in a *progressive* way with *linear* performance, see chapter 4 for further clarification of the *on-line* concept) but have the inconvenience of an uncontrolled output size. For this reason, a second subsystem was appended in order to control the output abstract length. In this case no analysis is carried out as color histograms are already available. The scoring stage is in charge of measuring the dissimilarity between all possible combinations of the BUs (video segments) received from the first subsystem obtaining an average dissimilarity measure for each of them. Finally, those BUs are ranked and discarded until the desired output size is reached. The second subsystem constitutes a complete *off-line* abstraction system that could be directly applied to a original video although it would be quite inefficient because all the possible comparisons between BUs are carried out. In the complete approach the number of BUs is heavily reduced in the first *on-line* subsystem obtaining a efficient overall pro-

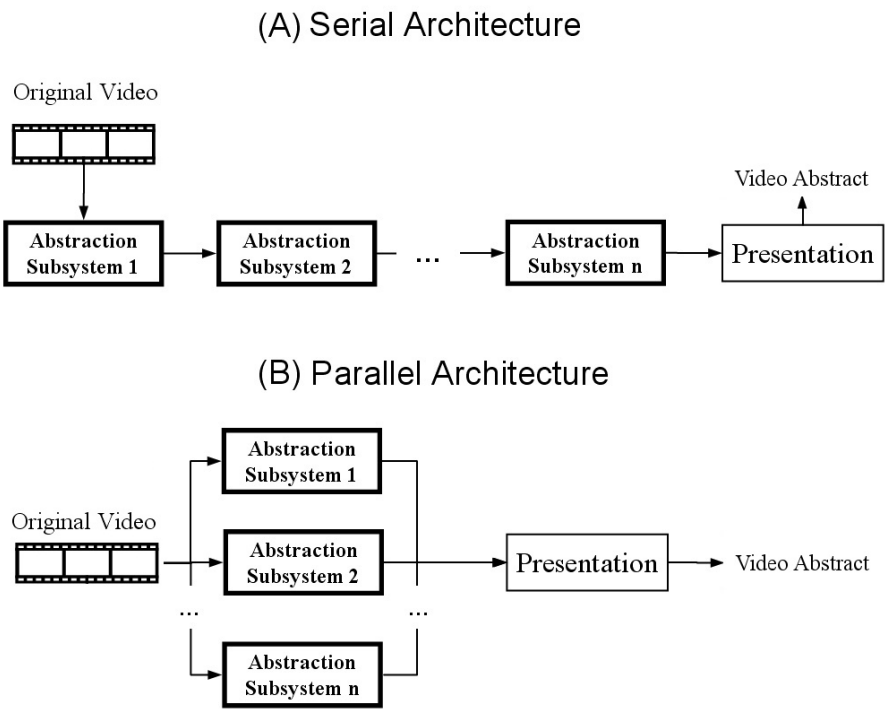


Figure 3.7: Composed Abstraction Architectures

cess.

Figure 3.7 (B) shows the architecture of parallel composed abstraction system. In this case the different abstraction subsystems do not process the results of the previous subsystems and deal with the incoming BUs in different ways. The results of the parallel abstraction processes are combined in order to generate the video abstract. [134] presents a system which can be easily modeled as a composed parallel abstraction system with two subsystems. The abstraction approach generates a video abstract by a frame (BU) subsampling process generating a 25x accelerated output. This subsampling approach can be represented as a simple abstraction subsystem with an unique 'selection' stage in charge of picking one out of every 25 frames. The particularity of the system is that authors consider that a 25x accelerated audio is incomprehensible and, for this reason, complete audio phrases are included in the abstract played at normal speed (the audio and video synchronization is lost). The audio processing can be considered as a parallel abstraction subsystem in which the BUs are composed by audio fragments. Speech recognition and SNR analysis are carried out for the segmentation of the audio in silence bounded phrases. The selection of phrases to be included in the abstract depend on their location (aiming to cover all possible locations), length and word repetition (aiming to avoid repeated sentences which can be found on BBC rushes). The final abstract is composed in the presentation stage by the combination of the independent results of the fast-forward (visual) and phrases selection (audio) abstraction subsystems.

For both of the depicted composed abstraction architectures -serial and parallel- it would be possible to define BUs and metadata flows between the different abstraction subsystems enabling the exchange of any kind of information. The combinations of composed abstraction systems is not limited to the two proposed architectures: any combination of subsystems could be applicable, enabling

the modeling of almost any possible abstraction system (nevertheless we have not found examples of more complex approaches in the literature).

## 3.4 Conclusions

In this chapter, a framework for video abstraction systems analysis and modeling from an operational point of view has been proposed. A taxonomy for classifying the different approaches based on their internal and external characteristics has been firstly depicted together with non-exhaustive classification examples of existing algorithms. The study of the influences and constraints between all the internal and external abstraction modalities can be a complex matter but will be useful for the design of abstraction systems.

Considering the defined taxonomy, an architectural model that allows the development of generic abstraction systems as a sequential processing of the original video BUs has been proposed. This architecture starts from isolating the different possible stages involved in the video abstract generation process, considering each stage as BU processing modules in charge of analyzing, adding information and redirecting the incoming BUs. This separation between the different abstract generation stages will allow the generic study of the abstraction algorithms by dividing the different approaches and studying each part independently. At the same time this division enables the development of generic interchangeable modules for the analysis, scoring, selection and presentation algorithms to be combined in different ways. Once we have a good understanding of video abstraction processes and have a standardized exchange established, the following scenario can be a reality: system A, developed by a video processing group, has a strong Analysis module - if input/output of the module is in standardized format, then system B, developed by a media producer, could borrow that module for its analysis and use system B's fancy presentation module to output a better abstraction.

The proposed approach will ease the task of analyzing the performance and internal/external characteristics of any proposed system in a unified framework applicable for subsequent systems comparison and characteristics specification as well as the classification of the different abstraction approaches attending to their architectural requirements. Additionally, the proposed architecture has allowed to define a set of elemental abstraction models, which are suitable for building almost any of the most spread abstraction approaches found in the literature. The proposed model can be used for evaluating and comparing our methods and systems with the ones from other research groups (e.g., what my group has been focusing so far is actually the 'scoring' stage, while that other group's system has a strength in the 'presentation' stage). Additionally, it is possible to study methodologically the possibilities of modifying a given algorithm with alternative internal or external characteristics. Examples of system modeling and modification of a given system external and internal characteristics can be found in Annex A.

# Part III

# On-Line Video Abstraction

# Chapter 4

# On-Line Video Abstraction Requirements and Implications

## 4.1 Introduction

In chapter 3, a taxonomy for video abstraction systems is presented. It is aimed to classify existing video abstraction technologies from an operational point of view, taking into consideration both their internal and external characteristics. Such approach is an uncommon point of view for the characterization of video abstraction systems if compared with other similar works, like for example [1] or the rest of taxonomies described in chapter 2, more focused in a systematic review of video abstraction techniques from the point of view of the applied abstraction mechanisms, without analyzing each system operational characteristics. Nevertheless, the two main classification categories in which the taxonomy was divided, video abstraction process external and internal characteristics, provide a convenient point of view for dealing with the subject of this work: the study and development of *on-line* video abstraction systems. From the set of defined external characteristics (chapter 3 section 3.2.1), we will focus on the *Performance* and *Generation Delay* which will determine if an abstraction system can be considered as *off-line on-line* or *real-time* (a subset of *on-line* abstraction approaches). The requirements of each possible category will be defined in the following sections.

The proposed taxonomy, together with the generic video abstraction architecture proposed in chapter 3 section 3.3.3, serves as the basic framework for the definition and analysis of the elements and concepts required for the definition of the operational constraints for building both *on-line* and *real-time* video abstraction systems.

The rest of the chapter is organized as follows: in section 4.2 the possible operation modalities are defined together with the terminology applied in the rest of the chapter. Section 4.3 deals with the operational requirements of each abstraction modality. In section 4.4, the practical issues of the *on-line* abstraction modality are discussed. Finally, conclusions are presented in section 4.5.

## 4.2 Definitions

### 4.2.1 *On-Line* and *Real-Time* Abstraction Systems

The target of this work is the development and analysis of *on-line* video abstraction algorithms, considering as well the particular case of *real-time* approaches. Attending to the taxonomy presented

in chapter 3, the kind of systems we will deal with must fulfill specific *Performance* and *Generation Delay* constraints. For the rest of the work, we will mainly focus on two possible types of abstraction approaches:

- *On-Line*: Abstraction systems with *linear* performance and *progressive* generation delay. To fulfill a *linear* performance, the amount of resources required by the abstraction approach must scale linearly with the length of the original video. On the other hand, the *progressive* generation delay implies that the availability of the complete original video is not required to begin the output abstract generation. With the fulfillment of both conditions, the abstract can be generated 'on the fly', as the original video is being broadcasted or recorded, making a video abstract available with a limited delay once the original video finishes (the amount of acceptable delay will depend on the application scenario) and being able to provide partial output during the original video processing.

- *Real-Time*: A *real-time* abstraction system is a particular case of *on-line* abstraction approaches and, therefore, it must fulfill the same requirements as an *on-line* system (*linear* performance and *progressive* generation delay) with the additional constraints of being able to generate the output abstract without pauses and at a high enough rate to enable its *real-time* visualization, consisting on being able to display the video abstract at regular video playing speed.

In section 4.3, the operational constraints of the defined *on-line* and *real-time* abstraction approaches are presented, based on the operational concepts defined in the following subsection 4.2.2.

### 4.2.2 Abstraction Systems Operational Concepts

Figure 4.1 shows an abstraction system depicted as a black box, without considering the stage division proposed (see chapter 3, section 3.3) which will be later taken into account. As discussed in the previous chapter, an abstraction system can be modeled as a BU flow through the different abstraction stages ('analysis', 'scoring' and 'selection') until all the BUs have been either discarded or included in the output abstract. Maintaining the same conceptual point of view, we will consider that the incoming BUs are generated by an external source arriving to the abstraction system at an average rate, $\overline{R_A}$. The abstraction system processes the incoming BUs and outputs them at an average rate $\overline{R_O}$. A fraction of the outputted BUs is selected to be included in the abstract at an average selection rate, $\overline{R_S}$, while the rest of the BUs are discarded at an average discard rate $\overline{R_D}$ (both measures expressed in BUs/second), with $\overline{R_O} = \overline{R_S} + \overline{R_D}$. In most abstraction systems, the BUs are collections of frames (with the associated audio samples) so the arrival rate, $\overline{R_A}$, depends on the incoming frame rate and the size or length of the BUs. For this reason, the input and output rates of the system can be expressed in terms of frames per second or number of BUs per second (according to the number of frames composing each BU). Analogous considerations could be applied if dealing with other kind of BUs such as audio samples instead of frames.

Depending on its characteristics, an abstraction system will process the incoming BUs individually or in groups of BUs -GoBs- and will require a specific amount of time for the processing of each $i^{th}$ incoming GoB, $T_{P_i}$, defined as the time such GoB spends inside the system until it is selected or discarded. The amount of time required for the arrival of the required number, $n$, of BUs composing the GoB must be taken into consideration. Such filling time, denoted as $T_{F_i}$ for the $i^{th}$ received GoB, will depend on the BU arrival rate, $\overline{R_A}$, and for the first GoB it will be defined as $T_{F_o} = \frac{n}{\overline{R_A}}$. Once the system has accumulated the required amount of BUs, it will spend, on average, an amount of time, $\overline{T_{GoBP}}$, in

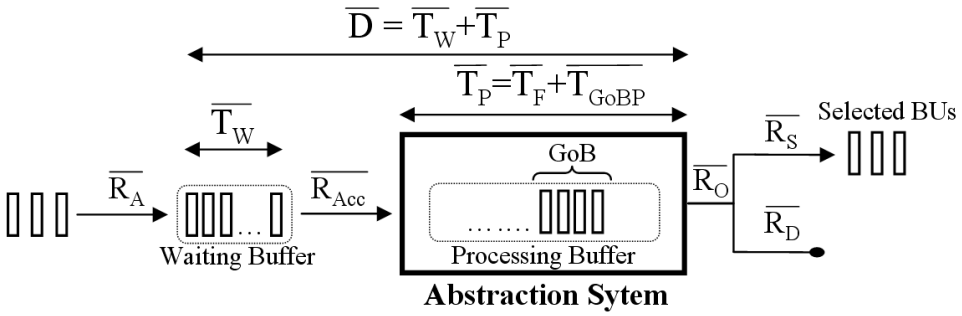


Figure 4.1: Abstraction System Operational Definitions

processing the GoB ($T_{GoBP_o}$ for the processing of first received GoB). Therefore, the total processing time for the first received GoB will be defined as $T_{P_o} = T_{F_o} + T_{GoBP_o}$ while, for the forthcoming GoBs, the filling time will be reduced due to possibility of receiving BUs during the previous GoB processing time lapse. Therefore, the number of BUs left to receive is determined as $n - (\overline{T_{GoBP}} \cdot \overline{R_A})$. For the rest of the processed GoBs, the average filling time, $\overline{T_F}$, will be expressed as

$$\overline{T_F} = max\left(0, \frac{n - \overline{T_{GoBP}} \cdot \overline{R_A}}{\overline{R_A}}\right) = max\left(0, \frac{n}{\overline{R_A}} - \overline{T_{GoBP}}\right) \tag{4.1}$$

resulting in no waiting time, $\overline{T_F} = 0$, if the GoB processing time is greater than the time required to receive a complete GoB, $\overline{T_{GoBP}} \geq \frac{n}{\overline{R_A}}$. The average GoBs processing time for the rest of the GoBs, $\overline{T_P}$, can be expressed as:

$$\overline{T_P} = \overline{T_F} + \overline{T_{GoBP}} = max\left(0, \frac{n}{\overline{R_A}} - \overline{T_{GoBP}}\right) + \overline{T_{GoBP}} = max\left(\overline{T_{GoBP}}, \frac{n}{\overline{R_A}}\right) \tag{4.2}$$

Therefore, given the average required time by the system for the processing of a GoB, $\overline{T_P}$, it is possible to determine the average rate at which the abstraction system is able to accept the received BUs for its processing, $\overline{R_{Acc}}$:

$$\overline{R_{Acc}} = \frac{n}{\overline{T_P}} = \frac{n}{max\left(\overline{T_{GoBP}}, \frac{n}{\overline{R_A}}\right)} = min\left(\frac{n}{\overline{T_{GoBP}}}, \overline{R_A}\right) \tag{4.3}$$

The maximum possible acceptance rate achieved by the system is therefore limited by the average arrival rate, $\overline{R_A}$ (logically, the system can not accept BUs not received yet). In the case of an ideal system with an infinite $\overline{R_A}$, that is, a system where all the BUs are instantly available (e.g. stored in a local repository with negligible reading and transfer times), the average BUs processing rate will be determined only by the GoB processing time, $\overline{T_{GoBP}}$, and the length, $n$, of the GoB.

BUs not directly accepted in the system at their arrival are inserted in a storage buffer (see figure 4.1) during a $T_W$ amount of time until the system is able to process them. Such time depends on the number of previous BUs stored in such buffer, $N_W$, and the average rate in which those BUs leave the buffer, that corresponds with the acceptance rate, $\overline{R_{Acc}}$. For the $i^{th}$ received BU, its waiting time, $T_{W_i}$, can be approximated as:

$$T_{W_i} = \frac{N_{W_i}}{\overline{R_{Acc}}} \tag{4.4}$$

where $N_{W_i}$ specified the number of BUs stored in the buffer when the $i^{th}$ BU arrives. It is straightforward to determine that the number of buffered BUs, $N_W$ will increase when the arrival rate is greater than the rate of acceptance, $\overline{R_A} > \overline{R_{Acc}}$, it will decrease if $\overline{R_A} < \overline{R_{Acc}}$ (eventually implying no waiting time, $T_W = 0$) and it will be constant, in average, in the $\overline{R_A} = \overline{R_{Acc}}$ case. Consequently, an increasing $N_W$ value implies a growing delay in the system due to the increment of the waiting time, $T_W$, yielding to an 'unstable' abstraction system in which the delay will depend on the length of the original video.

The system delay for the $i^{th}$ BU, $D_i$, will be defined as the total elapsed time since the BU arrives in the system (without being necessarily accepted immediately for its processing) until it is selected/discarded. Such time will be determined by the time spent in the waiting buffer, $T_{W_i}$ plus the time required for the $ith$ BU processing, $T_{P_i}$:

$$D_i = T_{W_i} + T_{P_i} \tag{4.5}$$

## 4.3 Operational Constraints

In this section, the operational constraints associated to the development of *on-line* and *real-time* abstraction systems are analyzed (subsections 4.3.1 and 4.3.2) together with the set of constraints that the individual abstraction system stages must fulfill (subsection 4.3.3). The set of most relevant constraints for both types of systems are summarized in table 4.1.

| | |
|---|---|
| *On-Line* | $T_N \leq \frac{N}{n} \cdot c \; ; \; \overline{R_A} \leq \overline{R_{Acc}} \; ; \; \overline{T_P} \leq \frac{n}{R_A}$ |
| *Real-Time* | $\overline{R_S} \geq V_R \; ; \; \overline{T_P} \leq \frac{n \cdot s}{V_R} \; ; \; N_I \geq (t - D_F) \cdot V_R$ |

Table 4.1: *On-Line* and *Real-Time* Abstraction Systems Operational Constraints

### 4.3.1 *On-Line* Systems

The first requirement for an *on-line* system is the *linear* performance, that is, the total amount of time required for the processing of video scales linearly with respect to its length. Considering $T_{P_i}$, the required time for the processing of the $i^{th}$ GoB of a video composed of $n$ BUs, the total time required for the processing of a $N$ GoBs video will be defined as:

$$T_N = \sum_{i=1}^{N/n} T_{Pi} \tag{4.6}$$

the linear performance implies that the condition $T_N \leq c \cdot \frac{N}{n}$ must be fulfilled. Such restriction implies that the total processing time is proportional to the amount of GoBs processed (with a linear relation determined by a constant value $c \in \mathbb{R} > 0$). The most straightforward way to assure such restriction fulfillment is to establish an upper limit for GoB processing time, $T_{P_i} \leq c$, $\forall i > 0$ so the total processing time scales linearly with respect to the number of GoBs

$$T_N = \sum_{i=1}^{N/n} T_{P_i} \diagup T_{P_i} \leq c,\ \forall i > 0 \Rightarrow T_N \leq \frac{N}{n} \cdot c \tag{4.7}$$

Another of the characteristics of an *on-line* abstraction system is its capability for starting the output process, selection or discard of BUs, without a complete availability of the original video $N$ BUs. *On-line* systems must operate with a limited delay, $D$, defined as the total elapsed time since a BU arrives in the system until it is selected/discarded (see equation 4.5). As it was depicted in the previous section, the time spent in by the $i^{th}$ BU in the waiting buffer, $T_{W_i}$, will depend on the number of BUs in such buffer when such BU arrives, $N_{W_i}$. Therefore, for a controlled $T_W$ value, the number of waiting BUs must be always limited and such situation occurs only if the system is able to accept incoming BUs at a rate at least as high as the arrival rate $\overline{R_A} \leq \overline{R_{Acc}}$. As an upper limit for the processing time was established, $T_{P_i} \leq c$, it can be derived that the fulfillment of the condition $\overline{T_P} \leq \frac{n}{\overline{R_A}}$ assures the *on-line* operation.

It is possible to assure that a system will be able to operate *on-line* if it is possible to guarantee that the input rate will be at least equal to the BU arrival rate and the required time to process a GoB of $n$ BUs is limited. The temporary unfulfilment of those conditions does not necessarily imply an *off-line* behavior of the system but, if those conditions are not controlled, it is not possible to assure the *on-line* processing in all situations.

### 4.3.2 *Real-Time* Systems

For the *real-time* operation mode additional conditions to those required for the *on-line* processing must be fulfilled: the average selection rate, $\overline{R_S}$, must be higher than a fixed visualization rate, $V_R$, which determines the minimum speed for continuous visualization of the output abstract

$$\overline{R_S} \geq V_R \tag{4.8}$$

The selection rate of the system, $\overline{R_S}$, will be determined by the total rate of BUs, $\overline{R_O} = \overline{R_S} + \overline{R_D}$, the system is able to output per time unit:

$$\overline{R_O} = \overline{R_{Acc}} = \frac{n}{\overline{T_P}} \tag{4.9}$$

and the fraction, $0 < s < 1$, of the processed BUs which are included in the output abstract:

$$\overline{R_S} = s \cdot \overline{R_O} = s.\overline{R_{Acc}} = \frac{s \cdot n}{\overline{T_P}} \tag{4.10}$$

from equations 4.8 and 4.10 the following limitation can be derived

$$\overline{T_P} \leq \frac{n \cdot s}{V_R} \tag{4.11}$$

establishing the maximum processing time per group of BUs allowed to fulfill the *real-time* operation mode. It should be noted that, apart from a maximum processing time per BU, a *real-time* abstraction system would require a homogeneous selection of BUs: the discard of too many consecutive BUs may produce pauses in the abstract playing. In a given time instant, $t$, the number of BUs included in the output summary, $N_I$, should be

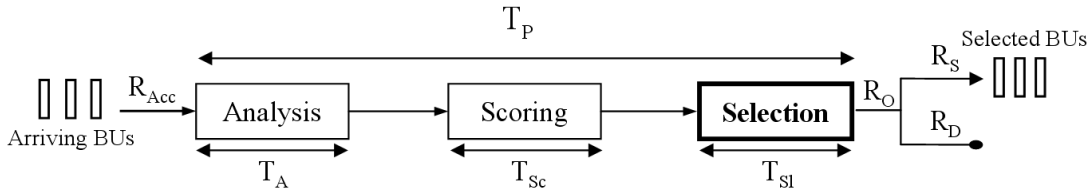


Figure 4.2: Stages Operational Definitions

$$N_I \geq (t - D_F) \cdot V_R \tag{4.12}$$

where $D_F$ represents the elapsed delay until the first BU is included in the abstract. From that time instant, the total selected BUs (included in the abstract) must be at least the number of displayed BUs.

### 4.3.3 Abstraction Stages Constraints

In the previous sections, the general operational constraints for guaranteeing abstraction systems with *on-line* or *real-time* operation modes were defined. This section is devoted to provide a brief analysis of how those global constraints may affect the individual abstraction stages depending on the kind of applied algorithms.

Figure 4.2 presents the three basic stages an abstraction system can contain (as defined in chapter 3), that is, 'analysis', 'scoring' and 'selection' modules. Each stage has an associated BU processing time: $T_A$ for the 'analysis', $T_{Sc}$ associated to the 'scoring' stage and, finally, $T_{Sl}$ for the 'selection' process. The total BU processing time will be determined by the sum of each individual stage BU processing times (for each $i^{th}$ received BU)

$$T_{P_i} = T_{A_i} + T_{Sc_i} + T_{Sl_i} \leq c \; \forall i > 0 \tag{4.13}$$

So, as long as the total processing time is kept under a constant upper limit, $c$, the *on-line* or *real-time* operation modes are assured (with a more or less restrictive $c$ value depending on the desired operation mode). This condition should be easier to control in systems with constant 'analysis', 'scoring' and 'selection' BU processing times, but it is not the natural way in which many existing abstraction techniques work.

A classification of abstraction systems based on their internal characteristics was defined in chapter 3 section 3.2.2. Systems dealing with fixed or variable size BUs were differentiated and two main categories were considered for the 'analysis', 'scoring' and 'selection' stages: *intra-BU* and *inter-BU* approaches. Depending on how an abstraction system is classified among those categories, there will be different ways in which the proposed constraints affect the system.

The problem relies on keeping the overall system BU processing time under a predefined threshold in all possible circumstances and, for this purpose, the individual complexity of each stage in the abstraction chain must be controlled. The difficulty arises when the applied algorithms have a computational efficiency worse than linear performance (e.g. quadratic, exponential) with respect to the amount of processed data. In the way we have represented an abstraction system, the amount of data corresponds to either the size of a BU (e.g. number of frames, audio samples) or the number of BUs needed in the processing (for example, *inter-BU* processes where a feature is extracted from several BUs, or where a number of BUs are compared). In both cases, we will denote the process carried out in

| Function Type | Performance |
|---|---|
| *type 1* | $\mathcal{O}(1), \mathcal{O}(log(n)), \mathcal{O}(n)$ |
| *type 2* | $\mathcal{O}(n \cdot log(n)), \mathcal{O}(n^2), \mathcal{O}(n^3), ..., \mathcal{O}(n^2), \mathcal{O}(n!)$ |

Table 4.2: Common Algorithms Performances

any abstraction stage as $f(n)$, with $n$ being the number of elements to be processed (elements within a BU, or number of BUs). We will differentiate two types of function, *type1* and *type2*, according to a computational efficiency criterion. *Type1* functions are those where the required processing time scales linearly (or better) with respect to $n$ ($f \in \mathcal{O}(n)$). For such kind of functions it is possible to find a constant $c$ satisfying

$$\exists n_0 \in \mathbb{Z}^+, c \in \mathbb{R}^+ \nearrow f(n) \leq c \cdot n \, \forall n > n_0 \tag{4.14}$$

The rest of functions, with a worse performance than *type1* ones, will be denoted as *type 2*. Table 4.2 enumerates typical example of performances for both type of functions[1].

#### *Intra-BU* Systems

We will consider, in a first place, *intra-BU* systems, that is, systems where the 'analysis' and 'scoring' processes work with individual BUs (fixed or variable length).

In the case of applying *type 1* functions, the size of the BU will not affect the overall system's performance: the time required for the processing of a BU will scale (in the worst case) linearly with respect to its size. Nevertheless, higher BU sizes will produce a proportional reduction in the average BU arrival rate, $\overline{R_A}$, resulting in no effect in the total amount of time required for processing a video but in a higher delay in the system, as can be deduced from equations 4.2 and 4.5.

*Type 2* functions require a more restrictive usage because, in this case, the reduction in the BU arrival rate is not sufficient to balance the increment in the individual BU processing time. In the case of a fixed size BU system, the required BU process time will be constant (with independence of the algorithm performance) and the processing times will be easier to control, just by applying fast algorithms, optimizing the algorithms complexity or reducing the fixed BU size when possible. On the other hand, abstraction systems working with variable size BUs (e.g. a system in which each BU corresponds to a shot in the original video and, therefore, the BU size can not be a priori determined) and applying *type 2* functions present more difficulties for the computational performance control: it is not possible to assure that all the incoming BUs are processed in an amount of time under the established limit $c$ (see equation 4.13) as such time depends on the BU size (it would be possible if the application scenario or type of content allows us to determine that it is not possible to deal with oversized BUs). In this case, one of the possibilities is to maintain a variable BU size with an upper limit so that the maximum processing time could be controlled. Another possibility could be to substitute the *type 2* applied functions for *type 1* approximations or change the operation modality to fixed-size BUs.

#### *Inter-BU* systems

All the previous considerations with respect to fixed or variable BU sizes are applicable as well to *inter-BU* abstraction systems. In such systems, together with the size of the BUs, the number of BUs

[1] A performance, $f(n) \in \mathcal{O}(1)$, implies a constant cost function

needed for the processing of each stage must be taken into account. The complexity of the applied algorithms should be controlled in the following situations:

- The complexity of the process depends on the size of the BUs: for example, many algorithms for video fragment comparison perform operations where all the elements (frames) of one of the fragments are compared with all the elements of the second one. When dealing with BUs composed of $n$ frames this kind of comparison are $\mathcal{O}(n^2)$, that is, *type 2* algorithms and, therefore, all the considerations taken for *intra-BU* systems should be taken into account.

- The complexity of the process depends on the number of BUs considered: for example, in case of redundancy elimination approaches, such as clustering approaches, it is quite common to compare a given BU with all the other BUs in the video and, even considering low cost BU comparison functions, a long video could yield to an excessive processing time per BU.

In general terms, in order to control the processing time and delay in *on-line* video abstraction systems, a proper strategy would consist on the implementation of fixed-size BU algorithms while establishing a limit in the number of BUs involved in any kind of processing. The consequences of such limitations in the application scenarios of *on-line*/*real-time* abstraction systems will be subject to study in the following sections.

## 4.4 *On-Line* Abstraction Practical Issues

In this section, some of the operational limitations of *on-line* abstraction systems will be discussed. The previous sections depicted the performance constraints required for the development of *on-line* systems in terms of processing times, algorithms complexity, delay, etc. In this section, we will take into consideration different possibilities for *on-line* abstraction systems and their limitations, derived from the lack of information and required computational performance associated to the *progressive* generation modality. The two main types of practical issues for the implementation of *on-line* abstract generation systems that we have identified are the following:

- Abstract length control: The lack of information about the length of the original video and the characteristics of the incoming BUs (due to the *progressive* generation modality) may difficult the control over the obtained abstract length.

- 'Analysis', 'scoring' and 'selection' stages precision: Depending on the type of implemented abstraction approach, the lack of information about the incoming BUs (as well as the computational complexity constraints) may cause a loss of precision in the 'analysis', 'scoring' and 'selection' stages.

Along the present section, in order to study the defined practical issues in different situations, we will present several types of abstraction systems in incremental complexity order. Most simple approaches are easily implementable but, at the same time, provide a smaller number of functionalities and possible application scenarios. More complex systems will be progressively presented explaining their additional functionalities and limitations. The types of systems considered are the following (based on the taxonomy and generic architecture provided in the chapter 3 framework):

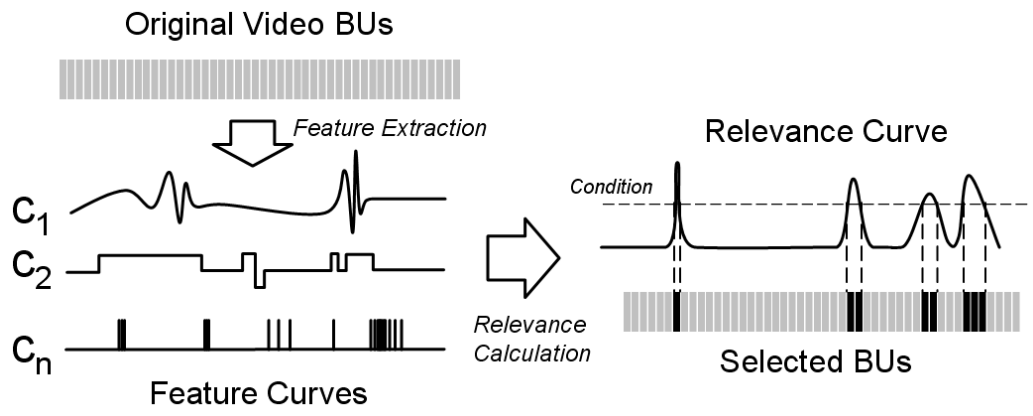


Figure 4.3: Relevance Curve Abstraction Mechanism

- *Unbounded* size, *intra-BU* systems: Abstraction systems where the abstract size is not a priori set and depends only on the abstraction system internal mechanisms. The *intra-BU* operation mode implies that the 'analysis', 'scoring' and 'selection' of a BU depend only on such BU characteristics (no information from other BUs is required).
- *Bounded* size, *intra-BU* systems: Abstraction systems in which the output abstract length is a priori set with *intra-BU* 'analysis', 'scoring' and 'selection' stages.
- *Inter-BU* systems: Abstraction systems containing *inter-BU* stages, that is, one or more of the 'analysis', 'scoring' or 'selection' stages requires information from several BUs.

For the analysis of different types of abstraction approaches and the practical issues associated to their implementation as *on-line* systems, we will consider their modeling as 'relevance curve' approaches. 'Relevance curve' [8, 104] and 'highlight oriented' [101, 102] based approaches are very similar video abstraction techniques relying on the same underlying mechanisms. Video abstracts are composed by the selection of a subset of fragments (e.g. frames, shots) from the original video fulfilling certain conditions. Such conditions could be, for example, the detection of specific events -highlights- (e.g. a goal in a soccer match [97], applause and cheering [135] or specific viewing patterns [136]) or an associated relevance value over a predefined threshold [137] (where the relevance value could be calculated based on the combination of any kind of extracted features: motion activity [8], manual annotations [105], video sequence quality [138], etc.). In principle, any type of abstract could be generated with those techniques as long as the appropriate analysis techniques exist. The generic video abstraction systems architecture proposed in the previous chapter (see chapter 3 section 3.3) demonstrates how different kind of video abstraction approaches can be modeled in such a way.

Figure 4.3 shows the process of a relevance abstraction approach: the original video is analyzed with one or more techniques, generating a number of associated feature curves, $c_1, c_2, .., c_n$. Such curves are processed and combined in some way for obtaining a relevance curve. The relevance curve should not be necessarily a numerical value, although such approach is the most usual, and could be a set of tags or annotations, a multidimensional value or whatever notation required for the specific application of the abstraction system. Finally, the original content fulfilling certain conditions (in figure 4.3, the condition is a simple thresholding) are selected for the composition of the video abstract.

In the following subsections, we will depict the implementation issues associated to each one of the three previously defined types of systems, modeled as 'relevance curve' approaches.

### 4.4.1 *Unbounded* Size *Intra-BU* Systems

Within the 'relevance curve' based abstraction approaches, the most simple systems are the ones in which the generated abstract size is *unbounded*, and which operate with *intra-BU* analysis, scoring & selection mechanisms. An *unbounded* size approach implies that the output abstract length is not known a priori and, therefore, the length of the output abstract will only depend on the amount of original content fulfilling the inclusion condition [50, 75]. The *intra-BU* category includes all those approaches where the analysis, score and selection of a given video fragment depends only on such fragment characteristics [82, 57]. The relevance value of a given BU will have no relation with the inclusion, discard or characteristics of other fragments from the original video.

The described approach is one of the most straightforward ones for its implementation as an *on-line* abstraction system because of the *intra-BU* mechanisms: an independent BU processing mechanism makes it possible to avoid one of the most important disadvantages in *on-line* abstraction system, which is to calculate the relevance of a given BU without information about the characteristics of the forthcoming ones. Another important advantage is that there is no need for the implementation of an output abstract size control.

Figure 4.4 presents the basic architecture for an *on-line*, relevance curve based, abstraction approach. The incoming BUs arrive at the analysis stage which analyzes the BUs individually or in groups. Such analysis stage can include an arbitrary number of analysis functions, $f_1, f_2, ..., f_n$, for the extraction of the same number of analysis values, $a_1, a_2, .., a_n$ which are associated to the processed BUs. The annotations for the BUs are individual, being, in this case, the feature curves the complete set of annotations for all the BUs. Such 'annotated' BUs are then processed by the scoring stage, in charge of assigning a relevance value, $r$, to each BU (or group of BUs) based on their associated annotations. Finally, the selection stage must decide which of the BUs are included in the output abstract and which of them should be discarded. As depicted in figure 4.4, the user preferences could be taken into account in the whole process for guiding the feature extraction, scoring (for example, using different weights for different preferred features) or the selection stage. The only consideration which should be taken into account for an appropriate *on-line* operation in this kind of abstraction system, is to keep each stage execution time under control, as discussed in section 4.3.3. For this reason, the possibility of building a specific type of *on-line* abstraction system with these characteristics (*unbounded size, abstract independent*) will mainly depend on the existence of computationally efficient analysis approaches for the extraction of the required features. In many cases, adapting an existing abstraction system for *on-line* operation mode relies in the amount of possible optimizations that can be applied for getting efficient enough techniques, from a computational point of view, without an excessive reduction in the precision of the obtained results.

### 4.4.2 *Bounded* Size *Intra-BU* Systems

The first restriction that can be found in the development of *on-line* abstraction systems is the implementation of *bounded-size* approaches, where there is an established target for the output abstract length. Such target size, as defined in the proposed external characteristics taxonomy (chapter 3, section 3.2.1), can be a fraction of the original video length [57, 127] or a fixed value [39, 84]. In systems

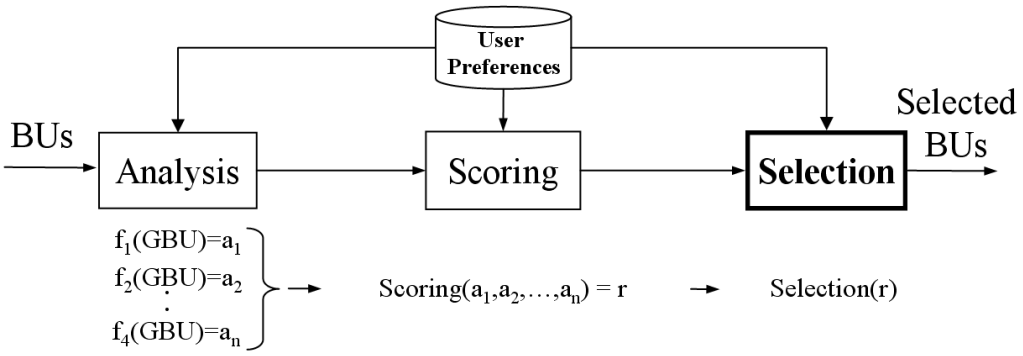


Figure 4.4: Relevance Curve Abstraction Stages Architecture

where the length of the original video is a priori known, the fixed output size approach is equivalent to setting an output abstract target rate with respect to the original video length.

The kind of system considered will be a relevance curve abstraction approach, as it has been defined in previous section, which enables the generation of any kind of abstract if a proper scoring mechanism exists.

**'Binary Relevance' BUs**

We will consider, in a first case, abstraction systems where the BUs from the original video can be classified in a binary way: those 'fulfilling the inclusion condition' and those 'not fulfilling the inclusion condition' (this approach is equivalent to highlight oriented systems where fragments fulfilling certain condition are included in the abstract [85, 97, 135, 136]). The specific set of BUs included in the abstract when the output size is predefined, will usually not be relevant as long as the included BUs are classified as 'fulfilling the inclusion condition'. An *on-line* implementation of such abstraction system should be straightforward: the BUs are sequentially analyzed and scored, and those 'fulfilling the inclusion condition' are included in the output abstract until the target length is reached. Other considerations such as, for example, the way to proceed when not enough 'selectable' BUs to reach the desired abstract size are available, will depend on the specific application scenario. Nevertheless, such problem will affect in an analogous way to an *off-line* abstraction approach.

**'Continuous Relevance' BUs**

The application of *on-line* approaches to scenarios where the BUs can not be binary classified and each BU fulfills the inclusion condition in a different degree presents more difficulties. In such case, some BUs are more relevant or appropriate than others for their inclusion in the output abstract [137, 8, 105, 87]. Given a predefined abstract size, the usual way to proceed is to select the subset of original BUs which maximize the total relevance value. The *off-line* implementation of such solution presents no complications: the original video BUs can be analyzed, rated and sorted in order of relevance and then the desired number of BUs can be just taken from the relevance-sorted list [87, 139]. On the other hand, an *on-line* approach involves more difficulties because, in the instant a BU must be selected or discarded, there is no available information about the characteristics of the forthcoming BUs. Therefore, it is not possible to determine if the selected BUs are the most appropriate to be

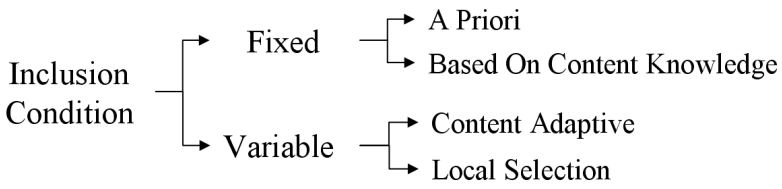


Figure 4.5: Size Control Diagram

included or if, otherwise, BUs with higher relevance values will be received in the future. The possible solutions for this problem and the quality of the BU selection will depend on the application scenario. The kind of original content, the level of knowledge and assumptions that can be made about such content, the available analysis tools, or the type of desired abstract are different conditions that will determine the final results and all of them should be taken into consideration when studying the application of an *on-line* approach to a video abstraction problem.

The main issue to take into consideration is the definition of the inclusion condition, that is, to determine what characteristics a given BU should fulfill in order to be included in the abstract. For example, in relevance curve based system, we should determine the relevance value limits for the selection of BUs. Figure 4.5 shows a diagram summarizing different possible approaches. The most straightforward one is the a priori determination of a fixed inclusion condition without further considerations. Such approach implies many difficulties for obtaining the desired output abstract sizes because such size will only depend on the arbitrary characteristics of the original video content. For this reason, this kind of approach will be, in most of the cases, only useful for *unbounded size* abstraction systems in which the length of the output is not a determinant aspect of the abstraction system. The application of more appropriate inclusion conditions could be feasible in scenarios where there is some previous knowledge about the original video characteristics. For example, if the abstraction system is always applied to the same video genre (e.g. soccer matches), it could be possible to make useful assumptions about the relevance values distribution along the video and, therefore, it could be possible to determine an appropriate initial inclusion condition for achieving output abstract lengths close enough to the target values. The deviation in the obtained lengths with respect to the target values will depend on the original content. If such deviations are acceptable is something that should be determined by the application scenario.

A different approach to deal with the problem is to make use of variable conditions during the video BUs processing. Such approaches are represented in the 'Variable' branch in the figure 4.5. The 'Content Adaptive' category represents abstraction systems which establish adaptive inclusion conditions that can be adapted on-the-fly to the characteristics of the video under process. Such conditions will vary according to the analysis of the already received content. For example, a possibility for setting an adaptive relevance threshold could rely on the analysis of the already received BUs relevance, extracting their statistical distributions and applying then an estimation of an appropriate threshold for the generation of an approximated length abstract. The kind of statistical analysis carried out, applied estimations and assumptions, and other parameters related to the selection process will determine the quality of the selection.

Another approach included in the variable inclusion category is the 'Local Selection' approach. In this case, in the same way as an *off-line* approach accumulates the complete set of the BUs information for the selection of the most relevant ones, a local selection process is carried out over subsets

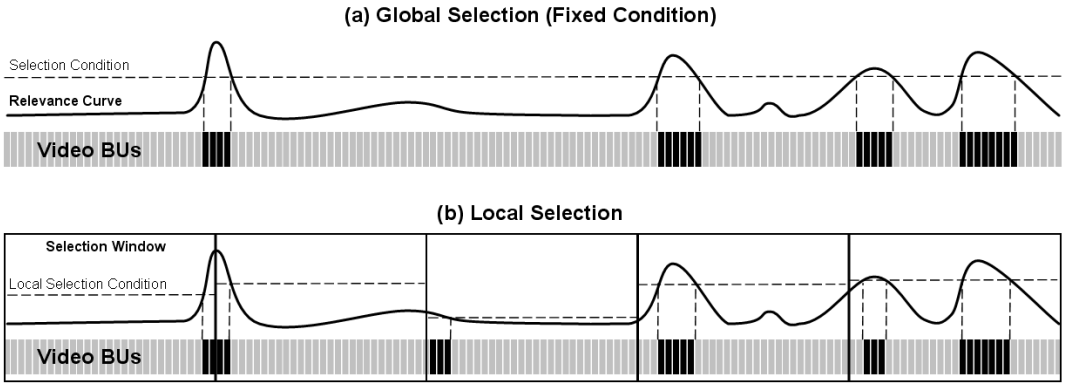


Figure 4.6: Fixed and Variable Inclusion Condition Examples

of the original video BUs. Instead of making instantaneous inclusion or discard decisions, a number of BUs is accumulated and, based on the information provided by the complete subset, those BUs considered relevant enough are selected to be part of the output video abstract. Figure 4.6 (a) shows an example of a selection process carried out over a complete video according to its associated relevance curve where the inclusion condition is fixed along the complete video BUs processing. On the other hand, Figure 4.6 (b) depicts a possible 'local selection' approach where the BUs are accumulated in different selection windows where varying inclusion conditions (different thresholds in the example) are applied. Of course, a local selection approach could be implemented in many different ways (sliding selection window, variable window size, etc.). The basic principle in this kind of selection processes relies on considering that a higher number of available BUs information permits, potentially, to obtain better selection results. At least, the availability of more information should not have a negative influence in the selection process. The drawback is the required accumulation of BUs in the selection stage that, in turn, implies a higher delay in the abstract generation process (see section 4.3), fact that could limit the number of scenarios in which this technique could be applied.

The way in which the selection process is executed is not necessarily limited to the defined categories and it could be possible, for example, to apply a combination of the approaches depicted in figure 4.5 or variations of the proposed techniques. It is possible to implement adaptive inclusion conditions limited by previous knowledge about the original content or, for example, to apply a local selection approach with a selection condition depending on the information acquired from the previously processed content (and not from the current selection window data).

The application of *on-line* selection will be, in the best case, as good as an *off-line* approach operating with the same data (a better selection can be carried out with the availability of the complete video information). The characteristics of the original content and the selected method will determine how close an *on-line* approach can get to an *off-line* process in terms of selection quality. Generally, the best kind of content for the application of *on-line* abstraction is that presenting constant or similar characteristics all along the video length. For example, the processing of a video which last set of BUs have similar characteristics to the first ones will permit the effective application of adaptive selection mechanisms (the statistics extracted from the first fragments of the video would be applicable to the last ones). The homogeneous distribution of relevant content along the video benefits the *on-line* approaches because, given the limited scope of the information handled by such systems, it increases the probability of a selection process where the most relevant content is correctly selected.

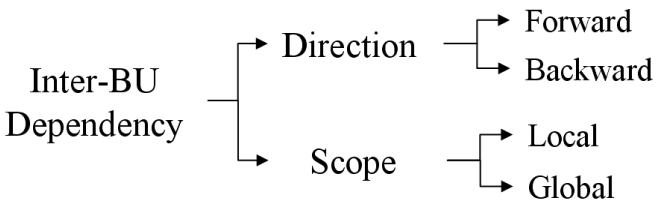


Figure 4.7: *Inter-BU* Dependency Diagram

On the other hand, a very heterogeneous content difficults the selection process: if we consider, for example, a video where the most relevant content is accumulated at the end of the video, an *on-line* system lacking of such information will probably select less relevant content from the beginning of the video given its unawareness about the relevant content accumulation at the end of the video. That means that not every kind of content is suitable to be processed by an *on-line* approach in a proper way and, therefore, each application scenario should be carefully studied before the application of *on-line* techniques.

### 4.4.3 *Inter-BU* Systems

The previous sections focused on the implementation of the most simple application cases for *on-line* abstraction; *unbounded size intra-BU* abstraction systems, which major limitation consists on the need of computationally efficient analysis and scoring algorithms. A further level of complexity is reached when dealing with systems where the selection process is constrained by the unawareness about not yet received content characteristics, discussed in section 4.4.2. In this section, we will take into consideration the implications of *inter-BU* abstraction approaches, that is, systems in which the analysis, scoring or selection of a BU depends on other BUs from the original content set [39, 75, 126].

There are many different ways in which *inter-BU* dependencies can be present in an abstraction system. Figure 4.7 depicts the two main aspects that we will consider for the analysis of *inter-BU* dependencies: direction and scope. Those aspects have been taken into consideration because they have a direct influence in the characteristics of an *on-line* implementation of a given abstraction system. The 'direction' category differentiates the cases in which the analysis, scoring, or selection of a BU depends on already received BUs -backward dependency- or the case in which any of them depend on forthcoming BUs -forward dependency-. In a backward *inter-BU* dependency case, in the instant a BU is received, the required BUs to complete its processing have been previously received so, as long as the previous BU information has been appropriately stored, the system is able to immediately process the current BU. A representative example can be found in 'sufficient content change' abstraction approaches [56, 44, 58], where original content is added to the video abstract only if it differs enough from previously selected content so, in this case, the selection of a given BU only depends on previous content. On the other hand, in case a BU processing depends on 'future' BUs, the system will be forced to wait for the reception of such BUs, implying an increase of the delay in the output abstract generation. Clustering approaches [50, 64, 62] usually require all the original BUs to complete the process: any BU can have both backward and forward dependencies with any other random-positioned BU (in this case the dependency relies in belonging to the same cluster and can affect the selected set of BUs).

The other main aspect about *inter-BU* dependencies to be considered is the 'scope'. Such term

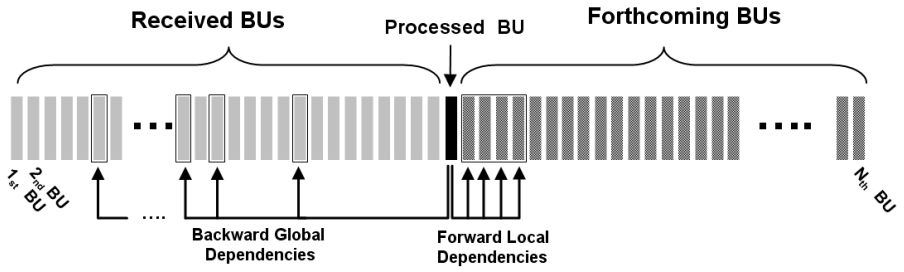


Figure 4.8: BU Inter-Dependencies Example

refers to the distribution of the BUs dependencies along the original video. In a local dependency case, the number of BUs required for the processing and their distances to the dependent BUs is limited. This is a common situation that can be found, for example, in systems where the feature extraction or scoring of a given BU depends on the adjacent or neighboring BUs. For example, in [104], motion activity is extracted from limited length group of frames -GoFs-. In local approaches, whether the dependencies are backward or forward oriented, the amount of accumulated BUs and, therefore, potential delay will be always limited. A global scope dependency will be present when it is not possible to determine the number and distance of the dependencies of the processed BU. The most common global dependency arises in redundancy removal approaches [91, 3, 62], which aim to remove repeated or too redundant events in a video. In such systems, the repeated events are distributed in arbitrary positions along the video and it will be not possible to a priori determine the amount of repeated events and their separation, factors that will determine the relevance of given BU.

Figure 4.8 shows an example of cases of backward global and forward local dependencies. In the first case, backward global dependencies, the current BU processing requires information from already received BUs located in any position in the original video. The second case, forward local dependency, is represented by the links to not yet received BUs which are, in this case, located in close and localized positions of the video. The combination of forward/backward and local/global dependencies will be possible but we will denote the dependencies in a given system with the name of the most limiting present dependencies, namely, the forward and the global ones.

As it has been previously explained, a forward dependency implies a higher delay in the abstraction system because the processing of a BU can not be finished until all the BUs it depends on are available. In any case, in the local dependency case, the amount of time and BUs to be processed are limited. On the other hand, when dealing with global dependencies (in terms of dependencies location or number) different problems which could influence the *on-line* operation of the system arise:

- An excessive amount of dependencies could influence the processing of a BU in one of the abstraction process stages.
- An undetermined number of dependencies in a forward dependency case can produce an unpredictable amount of delay in the system.

The number of BU dependencies in an analysis, scoring or selection process must be carefully considered when implementing *on-line* systems and has been previously analyzed in section 4.3.3, where the computational performance requirements for *on-line* approaches were described. The

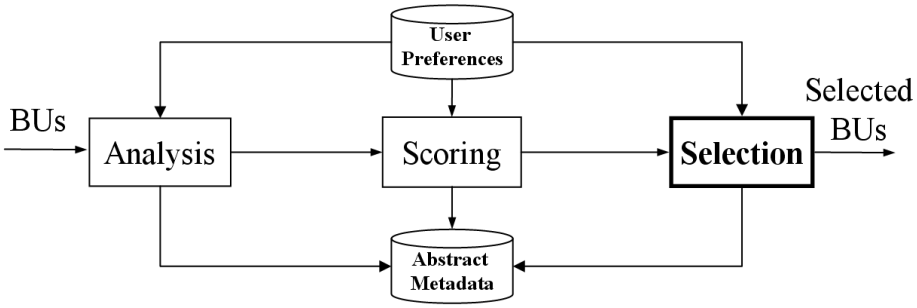


Figure 4.9: *Inter-BU On-Line* Abstraction Architecture

amount of BUs information that is required to store should be taken into account as well, due to the associated memory consumption, which could be limited in a given system. Even considering that the number of dependencies are kept under the required operation limits, the global dependencies may produce, in case of forward dependencies, unfeasible delays. Such kind of dependencies should be avoided in the implementation of *on-line* approaches and, if possible, substituted by alternative methods (some examples for redundancy elimination approaches will be provided in the following chapters).

A special case of *inter-BU* dependency can be found in the abstraction approaches classified as *abstract-dependent* [39, 57]. In such systems, the scoring or selection mechanisms depend on the previously selected or discarded BUs and not only on the original video characteristics: so, a BU processing depends on the generated abstract content. In 'sufficient content change' approaches [38, 75, 128], BUs are included in the output abstract if they differ enough from previously included BUs. Another example can be found in [60], where BUs are iteratively included in the abstract and those similar to those included are removed from the selectable set. Figure 4.9 presents an *on-line* architecture which enables the implementation of *inter-BU* approaches (including *abstract-dependent* ones) adding the Abstract Metadata storage which allows to keep information about previously processed BUs (that could be needed for *backward* dependencies) and generated abstract information.

## 4.5 Conclusions

In this chapter, taking into account the previously defined taxonomy and generic abstraction architecture, the restrictions and requirements of both *on-line* and *real-time* systems have been analyzed and formalized. Starting from the most basic abstraction models to more complex ones, the different algorithmical issues that must be taken into consideration for *on-line* implementations have been discussed. One of the main targets of the work presented in this chapter is to determine under which circumstances it is possible to build equivalent *on-line* or *real-time* approaches to existing video abstraction techniques and what are the limitations of those *on-line* and *real-time* systems.

The most appropriate approaches for *on-line* implementation have been analyzed and the main issues that should be addressed for dealing with more complex systems have been enumerated. The imposed constraints imply that, in the case of developing an *on-line* or *real-time* abstraction approach, techniques commonly applied should be modified or simplified to fulfill the required operational constraints. However, the possibility of applying an *on-line* approach will vary attending to

each specific application scenario, characteristics of the abstraction mechanism and type of content to be processed.

In the following chapters, the analysis carried out in this work is applied for the development of generic content *on-line*/*real-time* video skimming approaches as well as applications which fulfill the described operational constraints and solve the identified limitations.

# Chapter 5

# On-Line Video Skimming Algorithms

## 5.1 Introduction

In previous chapters, the characteristics of *on-line* and *real-time* abstraction systems, as well as the constraints associated to their implementation, have been defined. The main restrictions of the *on-line* approaches rely on the computational efficiency of the applied algorithms and the lack of information about forthcoming video fragments which can limit the precision of the abstraction system 'analysis', 'scoring' and 'selection' stages (as analyzed in previous chapter 4 section 4.4). All the described operational constraints aim to enable the implementation of systems fast enough to process the incoming content 'on-the-fly' and to output results with a limited and controlled delay. For this purpose, it is required to propose alternative solutions to common abstraction techniques, usually designed to provide *off-line* operation modalities. This chapter describes the work carried out for the development of efficient *on-line* algorithms which could be applied for generic content video abstraction. The reasons why *on-line* approaches work for video summarization based on redundancy removal are analyzed and applied.

Two experimental systems have been designed and developed (with preliminary versions presented to the TRECVid 2007 and 2008 campaigns). The purpose of the developed algorithms was the implementation of *on-line* abstraction approaches which could confirm the possibility of achieving results comparable to *off-line* algorithms with the restrictions of the *on-line* perspective. The outcome of the developed systems are video skims, which have the advantage of including audio and motion information although, in this case, the systems development was focused on the visual aspects of the video content (as discussed in chapter 3, the swapping between a video skimming and a keyframe extraction algorithm is usually straightforward).

In the following sections, we will firstly present, in section 5.2, related work in terms of existing abstraction approaches which provide low computational complexity and *progressive* generation functionalities. In section 5.3, the foundations of the redundancy removal approach are presented together with the justification of why such abstraction technique can be successfully applied in an *on-line* way obtaining good results. Afterwards, the developed summarization algorithms are presented in sections 5.4 and 5.5. The first one provides an *on-line* approach, without guarantee of a generating a specific length summary, which was employed as a first experiment demonstrating the possibility of obtaining results similar to other *off-line* abstraction approaches with an *on-line* system. Afterwards, the second summarization algorithm, a binary tree based approach, is presented. Such system represents the evolution and generalization of the previously developed systems for provid-

ing a highly configurable, *on-line* scalable (in terms of computational effort and generated abstract quality) abstraction system. The results of the early version of both approaches in the TRECVid evaluation campaigns can be consulted in section 5.6. Finally, in section 5.7, some conclusions close the chapter.

## 5.2 Related Work

We have defined *on-line* abstraction systems as *linear* computational performance approaches with *progressive* generation delay (see chapter 4). In most cases, abstraction systems with *linear* complexity are those performing local optimization or selection of the original video fragments maintaining a constant analysis and selection complexity. Many abstraction approaches rely on visual redundancy elimination, usually applying costly image and video fragment comparisons. On the other hand, straightforward solutions, like the selection of the first frame of each shot [58], direct video subsampling [64] or limit the number of comparisons to surrounding frames [52], are able to perform linearly and provide *progressive* generation delay. Approaches aimed for their implementation in commercial devices such as personal video recorders (PVRs) pay special attention to the computational performance of the system. Examples can be found in [140], an automatic highlight scene detection system, in [141], which presents a fast-forward abstraction approach relying in the detection of face tracks on the original video, or in [11], a recorded programs browsing system which classifies the content according to the number and position of detected faces. On the other hand, methods dealing with the abstraction problem as an optimization problem [60, 67], maximization of an objective function [126], or clustering based approaches [100], make use of the whole available original content for the abstract generation and require a number of comparisons which heavily increases with respect to the amount of original information, yielding to *non-linear* performance.

With respect to the generation delay, the most common approach is the *off-line* operation modality, that is, the abstraction algorithm requires the complete original data before processing the abstract. Clustering [50, 40, 100], rate-distortion [84] approaches or other methods such as [72], where the complete original video is mapped to a polyline later simplified for the generation of the video abstract, are typical *off-line* solutions. Most of the existing progressive abstraction approaches are reduced to subsampling methods like fast-forward approaches [27, 64] or systems where one keyframe is selected from each incoming shot [58] or group of frames (e.g. in [39] keyframes are extracted from each video segment accumulating a predefined amount of variation). More complex methods include potentially *progressive* adaptive playback approaches [142], *progressive* analysis for the identification of motion acceleration or deceleration points as keyframes [75], or methods based on local analysis of a feature curve extracted from the original video [128].

The *on-line* abstraction modality has not been specifically addressed in the literature and existing *on-line* systems are mainly reduced to basic subsampling [64] approaches and 'sufficient content change' keyframe extraction systems [75, 128, 38]. The first algorithm presented in this chapter (section 5.4) consists on a 'sufficient content change'. Its main novelty lies in being a video skimming system instead of a keyframe extraction approach, implying higher computational requirements and the novel mechanisms to control its computational complexity. Such system served as a first approach to evaluate the possibilities of *on-line* video skimming systems compared with *off-line* approaches. The second approach (section 5.5) constitutes a completely novel approach and no *on-line* systems, similar in terms of provided functionalities, have been found in the literature. Both algorithms have been evaluated, in terms of comparison with other abstraction approaches, within the TRECVid BBC

rushes evaluation campaigns (see section 5.6) and their functionalities are in-depth analyzed in chapter 7.

## 5.3 Redundancy Removal Foundations

Redundancy removal, usually based on visual similarity metrics, is one of the most common applied approaches for video abstract generation. Such mechanism consists in the elimination of fragments from the original content which contain repeated, that is, similar information already selected for its inclusion in the video abstract. The abstracts are generated as a reduced set of fragments from the original video which tend to be visually different from each other. It is usually assumed that the selection of very dissimilar fragments would necessarily provide a wider coverage of the original information. One of the advantages of this technique is the possibility of its application to almost any kind of content and, therefore, no specific domain information is required. This is one of the main reasons why the redundancy removal approach has been applied for the video skimming processes described in this section. Nevertheless, as it will be explained in the following subsections, different abstraction criteria may be integrated in the developed systems.

Clustering is one of the most common techniques applied for redundancy elimination. Such technique consists in the accumulation of similar fragments from the original video (e.g., frames, GoPs, shots) in groups called clusters. The similarity criteria may rely on visual, semantic or whatever features the system is designed to work with (the most common are the visual features). The video abstract is then built by taking representative fragments from the different groups and, in this case, the fragments considered as representative vary from one method to another (e.g. cluster centroid, longest fragment). Many video abstraction methods found in the literature [40, 62, 64, 50, 100, 63] summarize the original videos by the application of those clustering approaches. Nevertheless, there are alternative approaches for redundancy elimination like, for example: [119] which makes use of a graph based optimization method; [143] where authors make use of an 'excerpt shortening approach' (as defined in [1]) for retaining only the essential information for the comprehension of each original video excerpt; and [91] where a similarity matrix is generated, calculating the differences between all the fragments of the video, for a further selection of the most representative non-redundant fragments.

### 5.3.1 Redundancy Removal from an *On-Line* Perspective

From the *on-line* video abstraction perspective studied in this work, the most restrictive limitation of the most common redundancy elimination approaches is the requirement of processing the whole original video information for carrying out the clustering or global optimization process. According to the conditions defined in previous chapter 4, an *on-line* abstraction approach can not depend on the availability of the whole original content for starting the abstract generation process and, therefore, different redundancy elimination techniques must be applied. The most suitable are 'sufficient content change' approaches [75, 128, 38], where Basic Units -BUs-, that is, incoming video fragments of different length depending on the application (see chapter 3), are included in the output abstract only if they differ enough from previously included BUs. This mechanism provides a way to avoid excessive redundancy in the generated abstract and does not require the complete original content which can be, therefore, progressively processed. Nevertheless, some of the problems previously identified (see chapter 4, section 4.4) arise and others continue unsolved. For example, setting an appropri-

ate similarity threshold for BU inclusion without the complete content information will introduce an uncertainty in the output abstract length. On the other hand, one of the problems that must be addressed for the fulfillment of the *on-line* operation mode is to keep the computational complexity of the applied algorithms under the required levels for *on-line* (or *real-time*) processing. In 'sufficient content change' approaches, the amount of selected content is constantly growing as the original video is processed. Consequently, the amount of comparisons required for checking that an incoming video fragment (BU) is not redundant grows as well. Such growing number of comparisons can eventually imply a not *on-line* performance of the system and, hence, must be avoided.

The main approach applied in this work for solving the above identified problems is to limit the number of comparisons to be carried out, forcing the abstraction systems to perform only a reduced number of video fragment similarity checks. The hypothesis in which we rely is to consider that, in most of the video content, like movies or TV series, the visual redundancies are located within limited time lapses, that is, the probability of finding a video fragment similar to a given one is smaller as the temporal distance between both fragments increases. This may not happen in every type of content like, for example, news bulletins or quiz shows, where redundancies, like anchorperson shots, are uniformly distributed along the footage. In movies content it is quite likely, within a given scene, to find very similar shots (same character, same position in stage) but, as the movie and situations evolve, the characters, cloths, light conditions or locations vary and it is more difficult to find repeated content (from a visual point of view, as well as from a narrative one -e.g., the same setting may appear but it will belong to a different scene and may contain relevant semantic information, different from previous one-). If such hypothesis is fulfilled, then it would be possible to establish a time limitation for the comparisons carried out by an *on-line* abstraction system while keeping under control the amount of undetected redundancies and maintaining the computational complexity of the system under acceptable levels. As its has been mentioned, this may not be applicable to every video genre: for example, sport events typically contain visual redundancies homogeneously distributed along the video length which do not represent the same semantic information and a movie could contain as well repeated content in distant positions along its footage. Of course, the temporal window redundancy check will work better in content in which it is possible to assure that a high as possible amount of the redundancies are concentrated in a locality of the video fragment subject to comparison.

**Analysis of Redundancies Distribution in Video Content**

In order to determine the feasibility of the formulated hypothesis, a simple test has been carried out with commercial movies[1] and the TRECVid BBC rushes content[2], a much more redundant content than usual movies and TV series. The main objective of the test was to determine how much redundant content can be detected with the application of common visual distance metrics and to figure out the video timeline evolution of such redundancies. The first stage consisted on establishing a visual similarity distance. In this case, the chosen technique was a combination of the classical color histogram and MPEG-7 Color Layout [144], both of them calculated in the YCbCr color space. The color histogram is a technique commonly used in image retrieval [145] and summarization approaches (see chapter 2, section 2.4) because of its simplicity, speed, and reasonably good results. Nevertheless, such technique lacks of spatial information about the distribution of colors in the image and it has been combined with the Color Layout descriptor, applied as well for image retrieval

[1] 'Pasqualino: Seven Beauties', 'House of Flying Daggers', 'Mediterraneo', 'Hotel Rwanda' and 'Kagemusha'

[2] The complete set of 39 test videos from the 2008 evaluation campaign

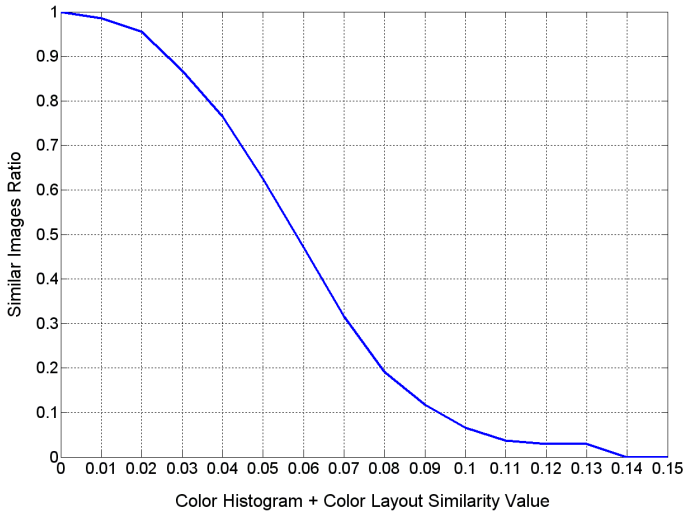


Figure 5.1: Visual Distance Vs. Similar Images Ratio

[146], which includes color spatial distribution information. The obtained distance, averaging the two original distances, is still far from providing the results of the human visual perception, but such circumstance is an inherent characteristic of all existing image comparison techniques (specially the fastest, and therefore simplest, ones). A test was carried out picking random frames (~150 frames) from 4 of the above mentioned movies random fragments (~15 minutes fragments subsampled at 1 second interval). Each frame was displayed together with the most similar frames found in the same fragment, according to the proposed distance. Finally, the distance corresponding to the most distant frame (according to the defined image distance) containing the same semantic information as the original one (from a subjective point of view) was annotated. It was observed that, for images which obtained larger values, it was more likely to find other images not representing the same information as the original one with smaller distance values. The graph shown in figure 5.1, which represents the probability of one image to be perceptually similar to another one depending on their visual distance, was derived from the experiment. It can be observed how, for small values (under 0.03) the probability that two images with such distance represent the same semantic information is above 90%. Such probability drops as the visual distance increases and, for example, for distances above 0.08, the amount of cases in which the two images represent the same 'semantic' information is under 20%.

The main problem related to existing image distances is their impossibility to capture the 'real' semantic meaning of the images. There is always a degree of uncertainty about the perceptual similarity of two images which may share the same color and shape structure or distribution but contain different information. Therefore, it is not possible to determine a fixed threshold for a perfect separation between semantically different or similar images. This limitation introduces an uncertainty in any visual comparison result which, depending on the case, may have even a heavier influence in the redundancy removal imprecisions than the applied fragment selection methodology.

Once we have approximated a degree of confidence in the defined distance metric for the identification of similar images, it is possible to study the distribution of the 'detectable' similarities in video content. For this purpose, we have carried out a test with the previously mentioned commercial

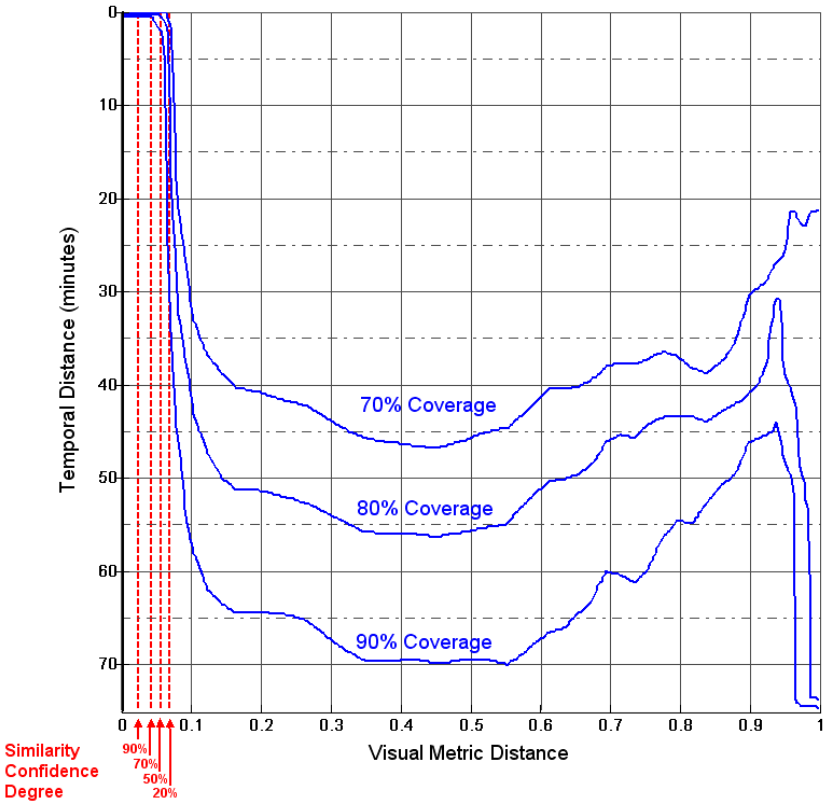


Figure 5.2: Movie Content Visual Redundancies Distribution

movies and the BBC rushes content. The original videos are firstly subsampled extracting 2 frames per second and all the frame cross-comparisons are carried out. The results are accumulated in a 2D histogram where each bin represents how many pairs of frames are separated by certain temporal ($y$ axis) and visual ($x$ axis) distances. For each possible visual distance value, $d$, the cumulative distribution function is calculated. Therefore, each position $(d,t)$ in the obtained distribution represents the probability that, given two frames separated by a visual distance $d$, their temporal separation, $t'$, fulfills $t' \leq t$. With such information we have derived the graphics displayed in figures 5.2 and 5.3. The 70%, 80% and 90% coverage lines represent the temporal margin (in minutes) which includes 70%, 80% and 90% of the pairs of frames with certain visual distance values ($x$ axis).

Figure 5.2 was calculated making use of the 5 complete movies previously identified without considering the first and last 5 minutes in order to avoid distortions caused by the opening/closing credits. Making use of this figure, we can determine that, on average, 80% of the pairs of frames in the studied movies with a visual distance value of 0.3 are separated by less than 54 minutes. The dotted lines in the graph, labeled as 'Similarity Confidence Degree' determine the visual distance values, from the previously defined tests (see figure 5.1), which provide a security of 90%, 70%, 50% and 20% of two images being semantically similar. The obtained graphics provide very significant information about the redundancies distribution in the video, being possible to determine that most of the detectable image redundancies are concentrated in small temporal distances. For example, if we establish an acceptable confidence degree for considering two images being similar as 70%, making use

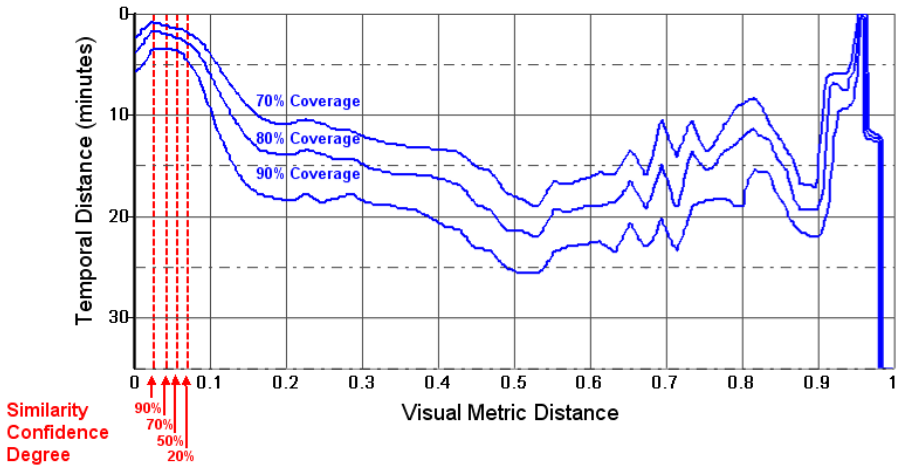


Figure 5.3: BBC Rushes Content Redundancies Distribution

of the data represented in figure 5.1, it is possible to determine that we require a visual distance value below 0.044 in order to reach such level of confidence. If we follow the 90% coverage level line, we can see how it crosses the 70% similarity confidence degree line in about 1 minute amount of time. Such result means that 90% of the pairs of frames with such computed visual distance are separated by less than a minute in the original video. It must be pointed out that the test has been carried out at frame level, without considering higher units (such as segments or shots), subsampled at 1/2 second rate. Without aiming to extract further conclusions, the graph shows a clear tendency of most part of the visually similar frames to be grouped in close temporal positions.

Figure 5.3 shows the obtained graph for the 39 BBC rushes videos leaving 2 minutes margin of unprocessed frames at the beginning and end of the videos to avoid the distortion introduced by the included test patterns. In general terms, the observed behavior is very similar to the values obtained with the movies content. Nevertheless, the temporal distance for similarity confidence degrees above 50% (visual similarity distance below 0.06) is greater than in the previous example due to the high redundancy of the BBC content, which includes repeated takes. In this case, as the videos are shorter (about 30 minutes maximum length), the obtained covering curves do not reach high distance values. It can be observed how the coverage curves present an increment in the distance values when reaching similarity values very close to 0 (for similarity confidence degree levels above 90%). This fact can be explained because of the junk content included in the BBC rushes videos that is mainly composed by blank frames or test patterns randomly distributed along the videos. Such kind of content (for example, blank frames) produces very low visual distance values when compared and it can be found in any position in the videos. However, the obtained graph shows again a clear grouping of most similar frames in very reduced time intervals.

In conclusion, it seems reasonable to consider that the temporal comparison limitation imposed by the *on-line* operation mode, as previously discussed, will not produce a heavy impact in the generated summary quality if considering the time lapses that cover the majority of the 'detectable' redundancies. In the following sections two *on-line* summarization approaches, which results in the TRECVid evaluation campaigns are included in section 5.6, are presented. Such results demonstrate how the performance of *on-line* approaches are comparable to *off-line* ones, which make use of the

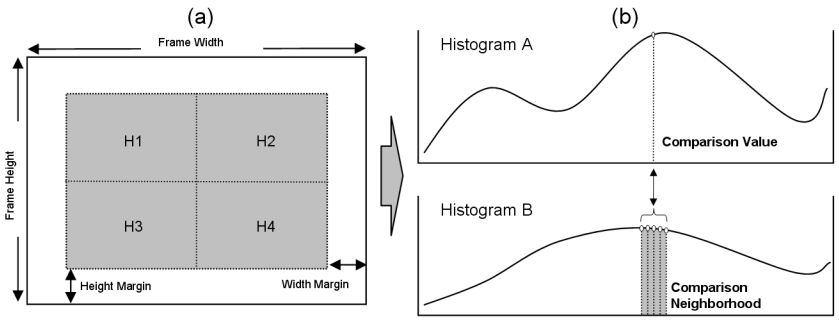


Figure 5.4: (a) Image Quarters Histogram Calculation; (b) Histogram Value Neighborhood Comparison

whole original content.

## 5.4 On-Line Video Skimming Based on Histogram Similarity

In this section, the first developed 'sufficient content change' abstraction approach is described. Such approach was applied in the submission presented to the TRECVID 2007 BBC rushes summarization task [147] (the results obtained are included in section 5.6). The method provides a very fast algorithm with low memory consumption which relies in fast frame and shot similarity measures applied in a redundancy elimination process. The reasons why the *on-line* operation modality can be effectively applied in combination with such redundancy removal techniques have been described in the previous section.

### 5.4.1 Similarity Measures

#### Frame Similarity

The Color histogram is a popular image indexing feature commonly used for image [148, 149] and video segments retrieval and identification [150, 151] due to the advantages it provides: simple implementation, speed of histogram calculation and comparison, and robustness and invariance to rotation and small scale changes. Its main disadvantages are associated to its high sensibility to changes on the image lightning conditions [149] and its impossibility for representing the spatial distribution of the colors in the image. There exist several techniques applied for the reduction or elimination of the illumination [152] or spatial [153] limitations of the color histogram. In this algorithm, in order to reduce these limitations while maintaining a high computational efficiency required for the *on-line* approach, the histogram is not calculated over the complete image but over rectangular areas resultant from the division of the original image in quarters. Such technique is a simple way to add spatial information to the color information provided by the color histograms [154].

As shown in figure 5.4 (a), the complete frame area is not used for the calculation of the histograms and only a central region is considered. Such reduced area is focused in the center of the image, which is considered to contain the most relevant information. Additionally, a smaller area reduces each frame histogram extraction time. Nevertheless, the usage of a too small processing area may produce undesired results, for example, if an object or element covers most of the processing area,

the color histogram results may not represent the whole image. In the experiments carried out, the width and height margins of unused information in each side of the histogram computing area is set to 1/6 of the total width or height respectively, obtaining a reduction in the process time required for the calculation of the color histogram.

Histograms are affected by changes in the illumination in terms of a horizontal displacement of the calculated indexes, principally in the Y channel. In order to reduce such effect, when two histograms are compared, the differences are not calculated directly between corresponding indexes in the histograms but as an average of a given histogram position and a neighborhood of positions in the compared histogram (5.4 (b)). Defining the original image histogram values as $Ho_i$ and the histogram values of the comparison picture as $Hc_i$ (histogram A and B respectively in figure 5.4 (b)) with $0 \leq i \leq 255$ (assuming 256 quantification levels on each color channel), the difference between the two given histograms is defined as:

$$HDiff(Ho, Hc) = \frac{\sum_{i=w}^{255-w} \sum_{j=-w}^{w} \frac{abs(Ho_i - Hc_{i+j})}{(2w+1)}}{(255 - 2 \cdot w)} \tag{5.1}$$

Where $w$ defines the number of values adjacent to the original histogram position (in both sides) which are part of the comparison neighborhood (see figure 5.4 (b)).

Considering $H(Im)_{q,c}$ as the histogram corresponding to image $Im$, quarter $q$ (see figure 5.4 (a)) and channel $c$ (with $1 \leq q \leq 4$ corresponding to the four possible quarters and $1 \leq c \leq 3$ to the 3 possible image color channels Y, U, V, -no color space transformation has been considered in order to avoid time consumption in color space conversions-) the final frame visual difference -*Diff*- value between to images $Im_1$ and $Im_2$ is calculated as the average color histogram differences of each image quarter and color channel:

$$Diff(Im_1, Im_2) = \frac{\sum_{c=1}^{3} \sum_{q=1}^{4} HDiff(H(Im_1)_{q,c}, H(Im_2)_{q,c})}{3 \cdot 4} \tag{5.2}$$

This visual distance measure will be used in the rest of the algorithm steps for different purposes such as shot similarity measure, shot change detection and shot variation metrics.

**Shot Similarity**

Histogram based visual distance metrics are commonly used for video segment retrieval by selecting a specific key frame and using it for comparison, selecting or dropping video segments [23, 155]. In this case, the comparison metric defined in the previous section is applied for the definition of a visual similarity metric between sequences of frames. Given two frame sequences $S_1 = \{f_{1,1}, f_{1,2}, ..., f_{1,a}\}$ and $S_2 = \{f_{2,1}, f_{2,2}, ..., f_{2,b}\}$ composed by $a$ and $b$ frames -$f$- respectively, we define the histogram difference matrix, *HDiffMatrix*, between $S_1$ and $S_2$ where each position $i, j$, with $1 \leq i \leq a$ and $1 \leq j \leq b$, is defined as:

$$HDiffMatrix_{i,j}(S_1, S_2) = Diff(f_{1,i}, f_{2,j}) \tag{5.3}$$

Each position of *HDiffMatrix* is defined as the difference value between the frames on both sequences indexed by the row and column of the matrix. Each row (or column) in the matrix represents the difference of a specific frame in one sequence with all the frames of the other. Figure 5.5 shows several examples of histogram difference matrices calculated for several kind of shots with themselves


Figure 5.5: Intrashot Histogram Difference Matrices

(darker values represent similar frames). The black diagonal on the obtained matrices represent the self comparison of each frame in the sequence.

Shot A (static group of frames) shows low and constant distance values, as all the frames in the shot are similar. In Shot B, a gradual increment in the difference matrix values (from the top left corner) can be observed, caused by the evolution of the shot. Shot C shows a slowly evolving sequence of frames with an abrupt change (corresponding to the white band in the right and bottom sides of the matrix) at the end of the sequence. Shot D shows an example of static camera sequence (dark areas of the difference matrix) in which some people cross the street in front of the camera (the white bands in the matrix). Finally, Shot E includes a shot change example which is clearly visible in the regions shown in the histogram difference matrix.

The *HDiffMatrix*, when self calculated for a given frame sequence, provides information about the evolution of such sequence and can be used to estimate its accumulated variation. Given two sequences $S_1$ and $S_2$ composed by $a$ and $b$ and frames respectively, we will define the mean difference value between them as:

$$MeanDiff(S_1, S_2) = \frac{\sum_{i=1}^{a} \sum_{j=1}^{b} HDiffMatrix_{i,j}(S_1, S_2)}{a \cdot b} \tag{5.4}$$

When self-calculated for a given sequence $S_1$, the $MeanDiff(S_1, S_1)$ will result in low values for static or low movement sequences, and in higher values for sequences with significant changes. Although the distribution of the changes in the sequence is lost, the speed restrictions imposed by the *on-line* operation mode requires to avoid other more complex and computationally expensive descriptors. Regarding the study of similarity between different shots, figure 5.6 shows several examples


Figure 5.6: Intershots Histogram Difference Matrix.

of histogram difference matrix calculation for different shots in each axis. Comparison between shots A and D shows low frame distances while some vertical higher distance bands appear (corresponding with the people crossing in front of the camera –see figure 5.5-).

Comparison between Shots A and E shows a light region to the left and a dark area to the right corresponding to the shot change within Shot E. Comparison between shots C and B shows how two completely different sequences result on an almost white (high distances) difference matrix. While the *MeanDiff* value provides information about the variability of a frame sequence, the intrashot *HDiffMatrix* provides a metric for frame sequence similarity when calculated for two different sequences. As each frame in the sequences is compared with all the frames of the other sequence, the obtained value is invariant to the temporal distribution of the frames in the sequences. Therefore, visually similar events do not need to be time-aligned in both sequences to detect their similarity. When dealing with long sequences, all the particularities included in each sequence, similarities and differences, tend to balance each other. Therefore, the final mean difference value loses significance and better discrimination results are obtained when comparing static sequences or short video segments. In this proposal, it is not considered to extract and apply more sophisticated (and computationally costly) descriptors due to the *on-line* and *real-time* target and, therefore, the approach is to split the original video in small subsegments trying to maintain as much visual coherence in the clips as possible. This approach will allow, not only meaningful comparisons between shots but to reduce the computational cost of the histogram difference matrix calculation ($O(n \times m)$, when comparing two frame sequences of $n$ and $m$ frames).

### 5.4.2 Shot Change Detection and Splitting

The first step in the proposed *on-line* abstraction chain is the detection of shot changes in the original video: the original frames are read one by one and stored in a shot detection buffer where the frame histogram distances are applied for the detection of shot changes [156]. The comparison is performed between several consecutive frames, so it is possible to detect both abrupt and gradual shot changes depending on the size of the shot detection buffer. Nevertheless, such size can affect the overall performance of the system because each decoded frame is compared with all the frames already stored in the shot detection buffer.

Figure 5.7 depicts the data flow in the summarization process. Figure 5.7 (a), the *Shot Detection Buffer*, corresponds with the shot change detection mechanism. When no shot change is detected the read frames are stored in the *Splitting Buffer* -figure 5.7 (b)- where the received frames are grouped and further processed for a subsequent splitting before sending them to the *On-line Shot Selection* step -figure 5.7 (c)-. The way in which the incoming groups of frames are divided in small frame sequences aims to minimize the *HDiffMatrix* calculation time while allowing the *On-line Shot Selection*

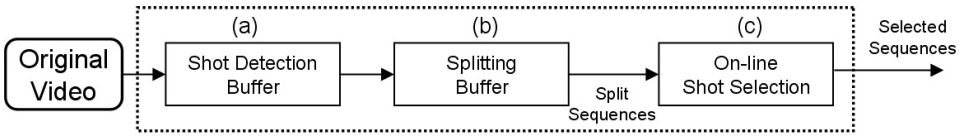


Figure 5.7: On-Line Stage Processing Flow

module to carry out meaningful comparisons (avoiding too long and heterogeneous sequences). For determining the frame in which the sequences are split, given an accumulated sequence $S$ composed of $n$ frames, a '*decay*' value is calculated as:

$$Decay(S) = \frac{MeanDiff(S,S)}{n} \tag{5.5}$$

This *Decay* value tends to 0 as the number of frames in the sequence, $n$, increases, but the decay amount depends as well on the *MeanDiff* value of the video segment which, as previously discussed, is an indicator of its variety and increases as the variety does. The *Decay* value drops faster in static sequences than in high variation ones. Assuming that a user requires a smaller amount of time for the understanding of a static sequence than in the case of a dynamic one, it is convenient to set a mechanism outputting more fragmented sequences in cases of low variation and viceversa. Moreover, higher fragmentation in static sequences will produce a more effective summarization because a high number of very similar video segments will be compared and most of them will be eliminated. For this purpose, the *Splitting Buffer* flushes its content to the *On-line Shot Selection* stage when a shot change is detected in the *Shot Detection Buffer* or when the *Decay* value reaches values below a specific *splitThreshold*, thus controlling the split size of each frame sequence. In the implemented system, the *splitThreshold* parameter has been experimentally set in such a way that the length of the obtained video fragments is about 20 frames length for static content and up to 40 frames for sequences with the high variety. Keeping the length of the segments in small values allows faster segment and shot comparisons.

### 5.4.3 Shot Selection

Once the received video has been split, the obtained segments are processed in order to decide about their discard or their inclusion in the final summary. The proposed approach has been designed to operate *on-line*, generating the summary as the video is being received and analyzed, but it does not allow to strictly control the output summary length, which will depend on the content characteristics.

#### Filtering of Subsequences

Before deciding if a video segment received from the *Splitting Buffer* should be included in the output summary or not, it must fulfill several conditions. The first filtering criteria is related to the length of the segment: too short segments are discarded in order to allow the user to correctly perceive the content and a minimum sequence length of 20 frames is applied in the system. Another implemented filtering mechanism is related to the amount of visual information provided by the video segment. For example, a uniformly colored set of frames does not provide information with enough relevance to be included in the output summary (black or white sequences, TV test patterns) and a mechanism to discard such kind of content has been added to the filtering step. The number of significant colors

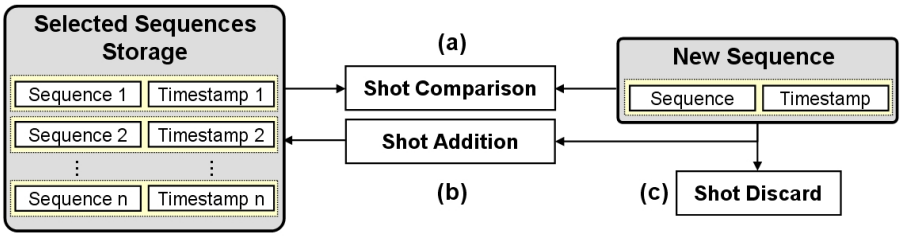


Figure 5.8: On-line Shot Selection Data Flow

on each frame are calculated counting the number of positions in the color histogram with a value above 1/3 of the maximum histogram value (considering each color channel). Those video segments containing too few colors are considered to provide no information and are filtered.

### On-Line Subsegment Selection

Figure 5.8 depicts the *on-line* shot selection process. Each new sequence of frames obtained from the *Splitting Buffer* (see Figure 5.7 -b-) is compared with the set of already selected sequences (Figure 5.8 -a-) obtaining the *MeanDiff* value between the new sequence and all the stored sequences with a timestamp difference below a configurable value. Setting a temporal difference limit for the comparison of video subsegments avoids an excessive and increasing number of comparisons as the video is being generated, improving the computational efficiency of the system without decreasing significantly the final results quality. The temporal structure of the original video is maintained as well, because similar video segments are eliminated only if they are close in time, characteristic which may be interesting in certain abstraction scenarios. Depending on the result of the comparison, the new frame segment is added to the *Selected Sequences Storage* (Figure 5.8 -b-) and written in the summary if it is different enough to the already selected sequences. For this purpose a *MeanDiff* value configurable threshold is defined for the inclusion or discard of the incoming fragments (Figure 5.8 -c-).

Therefore, the length of the obtained summary depends on the redundancy of the original video and the distribution of the similar fragments (similar and close in time video fragments have a higher discard probability), the maximum temporal difference for frame sequences comparison and finally the *MeanDiff* threshold for the inclusion of video segments. The computational efficiency of this method (which mainly relies on the number of comparisons between frame sequences) depends on the same variables: for a low *MeanDiff* threshold more sequences are included in the buffer and more segments comparisons are required for the processing of the original video, what makes the system to get slower. The same happens if the temporal comparison limit is increased, although the number of selected segments may be reduced because each incoming fragment is compared with more distant segments. Such limit should be set attending to the trade-off between the detectable redundancies (see redundancy distribution analysis in section 5.3.1) and the computational efficiency of the system. Section 5.6, where the customizations to rushes are described, includes several performance and generated summaries quality results corresponding to the TRECVid 2007 BBC rushes summarization task submission.

## 5.5 Binary Tree Based On-Line Video Summarization

This section describes the *on-line* video skimming algorithm developed starting from the experience gained in the development and evaluation of the previous system. The new algorithm provides a generic approach for *on-line* video summarization, scalable in terms of computational complexity, abstract generation delay and memory consumption as well as summary quality. The flexibility of the proposed approach enables the application of the abstraction algorithm for devices with different processing capacities or abstraction purposes. Furthermore, the proposed approach provides a flexible *on-line* framework in which different summarization criteria and techniques can be integrated.

The algorithm is based on the dynamic generation of a '*summary tree*' which models the different possibilities for inclusion or exclusion of the incoming video fragments. Based on such tree structure, the algorithm is able to calculate different generable video summaries as the video is being received and, iteratively, the best path in this binary tree is selected. Such path in the tree codifies the selection or discard of each incoming video fragment, characterizing the output video summary.

In the following sections the different parts of the algorithm are described: in section 5.5.1, an overview of the binary trees skimming approach is provided, section 5.5.2 describes the applied frame and shot comparison techniques, while the branch scoring and pruning mechanisms are described in sections 5.5.3 and 5.5.4 respectively.

### 5.5.1 Dynamic Tree Summarization

As it has been discussed in previous chapters, the *on-line* abstraction approach implies several limitations with respect to *off-line* systems (e.g., small and limited delay, progressive generation, lack of complete information about the incoming video) which may produce, given the constraints of the selection process, a negative impact in the results quality. The output summary length control is one of the problems that can be found. For example, in the basic system described in section 5.4, the summary length depends on the characteristics of the original content. Another issue identified in the previous system was related to the potential summary quality loss caused by the instant selection or discard of the incoming video fragments, without any mechanism to check if subsequent fragments could be more appropriate for their inclusion in the summary. The dynamic tree summarization approach described in this section is based on assuming that the usage of a buffer storing $n$ incoming video fragments allows to improve the selection process. Such buffer allows choosing from 1 to $n$ fragments on each segment inclusion decision instead of taking instantly inclusion/discard decisions based on the characteristics of a single incoming fragment. The precision of the fragment selection process will increase with higher $n$ values, enabling to reach an *off-line* approach equivalent precision with $n$ equal to the total number of fragments in the original video. Nevertheless, the increment of the buffer length, $n$, will introduce as well a minimum delay in the summary generation of $n$ video fragments (the video fragment buffer must be filled before starting the selection process, as described in chapter 4) as well as an increment in the complexity of the selection system (up to $2^n$ possible combinations of fragment selection could be evaluated for its inclusion).

#### Summarization Trees

Binary trees have been selected to model the different possibilities for video fragment selection from the buffer. The proposed approach assumes the reception of arbitrary length incoming video fragments (which could correspond to isolated frames, fixed length blocks of frames, shots, etc.) which

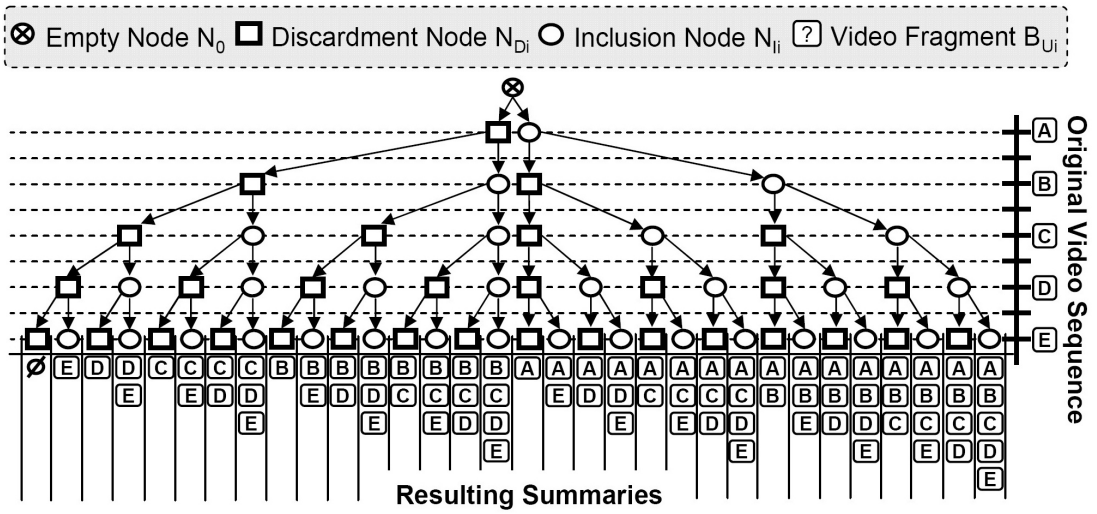


Figure 5.9: Summarization Tree Example

will be denoted as basic units, *BU* (see chapter 3 framework definition). The original video $V = \{BU_1, BU_2, ..., BU_s\}$ will be, therefore, composed by $s$ basic units.

The summarization tree will be constructed starting from an empty root node $N_0$. For each $i_{th}$ received basic unit ($BU_i$) we define two types of nodes: inclusion -$N_{Ii}$- and discard -$N_{Di}$-; which correspond, respectively, to the inclusion or discard of $BU_i$ in the output summary. For each $BU_i$, an instance of the two types of nodes are appended to the existing $i-1$ position nodes (all $N_{I(i-1)}$ and $N_{D(i-1)}$ nodes, or the $N_0$ node in case $i = 1$): we denote this operation as *'expand'*. An example of a generated tree is shown in figure 5.9: each tree branch represents a possible video summary and the type of each level $i$ node indicates the inclusion or discard of the $i_{th}$ video fragment, $BU_i$.

Given a summarization tree starting at level $i = 0$ (initial node $N_0$) with a depth of $d$ *BU*s, all the possible combinations of inclusion and discard of $d$ *BU*s are represented in the different branches of the tree. Each branch ends in a $d$-level terminal node without child nodes that is denoted as a 'leaf' node $L_d^n$ ($n_{th}$ leaf at $d$ depth). Each leaf, $L_d^n$, in the tree represents a possible video summary as the sequence of *BU* inclusions/discards from the initial node $N_0$ to the $L_d^n$ node.

For an original video $V$ composed of $s$ *BU*s, a summarization tree of depth $s$ can be built representing all the possible generable video summaries, each one corresponding to one of the possible $L_s^n$ leaf nodes ($1 <= n <= 2^s$). Assuming that it is possible to compute a score value, $S_s^n$, for each generable video summary, $L_s^n$, according to a given criterion, an optimal summarization process would consist on the selection of the tree branch corresponding to the $L_s^n$ with higher $S_s^n$ value. As all the generable video summaries are evaluated, the generation of the best summary, according to the selected criterion, is assured.

Nonetheless, the computational requirements for the generation of a $s$-depth summarization tree requires the evaluation of $2^s$ video summaries which, even for very efficient scoring algorithms, is computationally unfeasible for high $s$ values (and even more for *real-time* requirements). Moreover, such tree structure would require memory capacity for the $2^{s+1}-1$ total nodes composing the tree (the memory required for storing each node will depend on the needed information storage per node which will, in turn, depend on the summary scoring algorithm). Therefore, a more computationally efficient approach, enabling *on-line* and *real-time* operation modes, is required and is described in

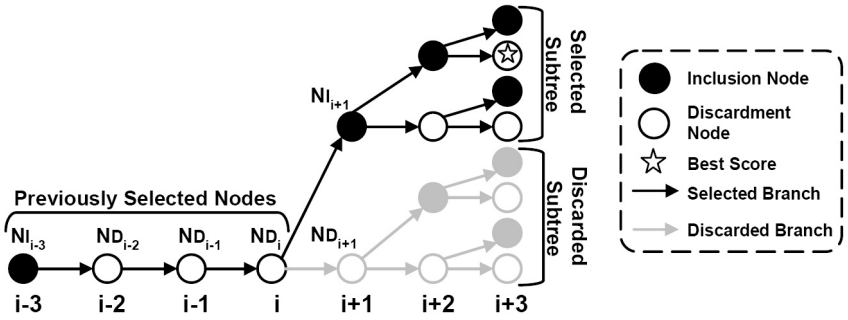


Figure 5.10: Dynamic Tree Example

the following sections.

**Dynamic Sub-Trees**

The *on-line* summary generation is achieved by the generation of partial summarization trees: given a required maximum delay $D$ (that is, the period between the reception of a $BU$ in the system and the decision about its inclusion or discard in the abstract -see chapter 4 section 4.2-) the algorithm builds an initial summarization tree of depth $d = D$. At this point, in order to keep the maximum allowed delay, the $BU_1$ (the first basic unit inserted in the system) must be either selected to be included or discarded for the summary. In order to take such decision, all the scores $S_D^n$ corresponding to the tree leafs are calculated and the subtree including the $L_D^n$ with higher score is selected (alternative branch selection mechanisms such as, for example, the selection of the subtree with higher average leaves score, could be considered). The selected subtree will necessarily start with either the node $N_{I1}$ or the node $N_{D1}$; the type of such node indicates the inclusion or discard of $BU_1$. Hereafter, the complete subtree with $N_0$ as root, which does not include the leaf with best score, is discarded and eliminated from the system. The first node of the selected subtree ($N_{I1}$ or $N_{D1}$) is then set as new tree root. The resulting structure is a subtree of depth $d = D-1$ (starting at level 1 and ending at level $D$) ready to be expanded with the next incoming basic unit $BU_{D+1}$ and to iteratively repeat the process by discarding the subtree with the lowest score and by expanding the one with the highest score.

Figure 5.10 shows an example of a tree with depth $d = 3$ when $BU_{i+3}$ has been just received and the tree has been expanded. A decision about the inclusion of all the BUs from $BU_1$ to $BU_i$ has already been taken and the next step consists on the decision about the inclusion of $BU_{i+1}$ in the summary. The scores corresponding to all the tree leafs are calculated and, in this example, result in a maximum score associated to a leaf node located in the $N_{Ii+1}$ subtree. Such subtree is selected, while the $N_{Di+1}$ one is discarded. As the root of the selected subtree is an inclusion-class node, $N_{Ii+1}$, the basic unit $BU_{i+1}$ is included in the summary. Finally, the new tree root is updated and the dynamic tree is ready for the next iteration.

The described mechanism allows the progressive summary generation by building partial subtrees as the video is being received. The tree size, and hence the summarization process delay, is maintained by an expansion-reduction mechanism corresponding to the subtree selection (reduction) and to the new $BU$ instance nodes addition (expansion).

**Summarization Tree Scalability**

The application of the abstraction tree approach provides a generic mechanism for *on-line* video summarization in which three main aspects of the system performance may be easily controlled varying the tree generation parameters (tree depth and maximum number of tree branches):

- *Abstraction Delay*: The maximum summarization delay $D$ can be controlled with the maximum depth $d$ of the generated sub-trees. Higher $d$ values imply more time between the instant a *BU* enters in the system and the instant in which it is selected or discarded. On the other hand, in the case of applying higher $d$ values, a better selection process can be expected (due to the higher number of *BU*s and, hence, tree branches evaluated). Finally, if an unbounded $d$ value is applied, the complete summarization tree will be generated and the process will be equivalent to an *off-line* approach (the complete original information must be received before the selection of the best path in the tree).

- *Computational Performance*: The computational performance of the summarization process will depend on the number of tree branches that must be scored as well as each branch evaluation cost. The maximum possible number of summaries evaluation on each iteration (that is, the number of tree branches or leafs), $2^d$, grows exponentially with respect to the subtree depth $d$. This growth may imply the evaluation of too many branches, preventing the system from reaching *real-time* performance. In order to deal with this limitation, the number of branches must be limited by considering low $d$ values (which imply low delay and higher speed but reduced summarization accuracy) and/or by the application of tree pruning algorithms (i.e. elimination of low-scored branches) avoiding the expansion of too many branches.

- *Memory consumption*: The memory consumption of the abstraction algorithm will directly depend on the number of stored nodes and the amount of information associated to each node. The branch scoring algorithm will determine which information each node must keep for its containing branch evaluation and, in any case, the amount of nodes will be determinant in the memory resources consumption. As it has been previously stated, the number of nodes will depend on the tree depth and the number of branches it is composed of and, therefore, the memory consumption will be directly related to the summarization delay and performance.

Although for most applications the generation parameters, tree depth and maximum number of branches, may be set as fixed values, the proposed system is particularly suitable for the implementation of adaptive systems where the depth of the tree and the number of evaluated branches could dynamically vary in order to achieve specific performance results, such as fixed abstract generation rate or delay with varying computational resources (e.g. a summary could be generated in a system with low memory or CPU speed at the same rate than in a faster system by computing lower depth trees or evaluating a fewer number of branches, generating a lower quality summary). A *real-time* summary generation application making use of such scalability functionalities is presented in chapter 8.

### 5.5.2 Frame and Segment Similarity

In our first *on-line* system (see section 5.4), the applied visual similarity measure was based on the calculation of color histograms of the frames in the video sequences. As it has been commented, the color histogram performs well for measuring the similarity between two images in quantitative terms

but fails in the differentiation of the spatial distribution of the colors in the image. A division of the original images in four quarters with a subsequent calculation of the color histogram of each quarter was applied, obtaining better results than with a simple histogram but still limited. In the improved system, the color histogram was substituted by the MPEG-7 Color Layout descriptor [144], specifically designed to be a fast, resolution independent descriptor aimed for image retrieval, indexing and video shot identification. Therefore, it fulfills the *real-time* requirements of the proposed approach while capturing the spatial distribution of the colors in the image.

Once the Color Layout is computed for each frame within a video segment, it is necessary to define a sequence comparison mechanism. Given two sequences $S_1 = \{f_{1,1}, f_{1,2}, ..., f_{1,a}\}$ and $S_2 = \{f_{2,1}, f_{2,2}, ..., f_{2,b}\}$ (constituted by $a$ and $b$ frames respectively), the first step is to compute the similarity between all the frames in both sequences, obtaining the *similarityMatrix* between $S_1$and $S_2$. In this matrix position $i$, $j$ is defined as

$$similarityMatrix_{i,j} = CLDiff(f_{1,i}, f_{2,j}) \tag{5.6}$$

where *CLDiff* corresponds to the Color Layout difference between two frames (as defined in [144]). The computational cost of the *similarityMatrix* calculation is $\mathcal{O}(N^2)$, being $N$ the length, in frames, of the sequences. The frame sequence length limitation applied in the proposed algorithm assures a constant and limited complexity in the calculation of the difference matrix for visual segment comparison.

For each row $r$ in the *similarityMatrix* we define

$$minRowValues_n(S_1, S_2, r) = \{mv_{r1}, mv_{r2}, ..., mv_{rn}\} \tag{5.7}$$

as the subset of the $n$ minimum values in row $r$ of the $similarityMatrix(S_1, S_2)$. With such values we define the distance measure

$$minDistance_n(S_1, S_2) = \frac{\sum_{i=1}^{a} \sum_{j=1}^{n} mv_{ij}}{a \cdot n} \tag{5.8}$$

being $a$ the number of frames in $S_1$. As this measure is not symmetric ($minDistance_n(S_1, S_2) \neq minDistance_n(S_2, S_1)$) we finally define the similarity measure as

$$distance_n(S_1, S_2) = \frac{minDistance_n(S_1, S_2) + minDistance_n(S_2, S_1)}{2} \tag{5.9}$$

In case $n = a = b$, with $a$ and $b$ being the number of frames in $S_1$and $S_2$, the $distance_n$ value obtained is equivalent to the *similarityMatrix* average value, a visual segment similarity measure which fails to represent differences in cases such as the comparison between shots A and B in figure 5.11 (video fragments from TRECVID 2007 content set) where, although several frames from each video segment are very similar, the average difference will be negatively penalized by the rest of dissimilar frames.

On the other hand, in the case $n = 1$, the proposed metric represents the average between the minimal differences between all the frames in $S_1$and $S_2$. Figure 5.11 shows the example of shots A and C, which are very different with the exception of the last fragment of shot C: a comparison with $n = 1$ will cause a high similarity value. The application of the $n$-most similar frames comparison method with intermediate $n$ values requires a minimum number of similar frames between two sequences in order to produce high similarity values.


Figure 5.11: Shot Comparison Examples

### 5.5.3 Branch Scoring

When considering the characteristics of the generated video abstract, the branch scoring mechanism is the core of the proposed summarization approach. The branch scores are applied for the selection of the summarization sub-trees kept during the tree branch reduction process applied to control the computational complexity of the approach. Therefore, such scores will determine the characteristics of the generated video summary.

The proposed summarization scoring system aims to generate video summaries with the following characteristics:

- Controllable summary length with independence of the characteristics of the original content.
- Lowest possible visual redundancy: including fragments as different as possible from a visual point of view will reduce the perceived summary redundancy and will increase the probability of including a higher number of different events in the output summary.
- High continuity values: smoother summaries will positively influence the perceived summary quality [126].
- Inclusion of high activity segments: including the fragments with higher activity values aims to maximize the probability of including events in the output summary.

In the developed scoring system, an independent metric has been extracted for the measurement of each of the defined characteristics and later combined.

#### Summary Length Score

The summary length control has been implemented by determining a target summarization rate, $0 < target < 1$. In the developed algorithm, each $d$ level node, $N_d$, associated to $BU_d$, contains information about the number of frames of $BU_d$. Such number of frames will be denoted as $nF(BU_d)$ while the total number of possible and included frames in the tree branch containing $BU_d$ will be denoted as $totalF(BU_d)$ and $includedF(BU_d)$ respectively. The summarization ratio is calculated based on those values as follows:

$$sumRatio(BU_d) = \frac{includedF(BU_d)}{totalF(BU_d) \cdot target} \tag{5.10}$$

And the size score is defined as:

$$sizeScore(BU_d) = \begin{cases} \frac{1}{sumRatio(BU_d)} & sumRatioBU_d > 1 \\ sqrt(sumRatio(BU_d)) & sumRatio(BU_d) \leq 1 \end{cases} \tag{5.11}$$

**Continuity Score**

In this work, it is considered that the output summary will be perceptually more pleasant if the number of discontinuities is reduced as much as possible. As the proposed system splits the original video in small fragments, it is possible that the output summary could contain unpleasant 'cuts'. To avoid such effect, we introduce in the system a mechanism for rating summaries with less discontinuities with higher scores. A *continuity* will be present in a node when both such node and its parent node are included nodes. We define $continuities(N_d)$ as the total number of *continuities* in the summarization tree branch ending in node $N_d$ and $includedN(N_d)$ as the total number of inclusion nodes contained in such path (that is, the number of included *BU*s in the summary represented by the tree branch). The continuity score is defined as the ratio between the included nodes and the number of *continuities* in the tree branch:

$$continuityScore(BU_d) = \frac{continuities(N_d)}{includedN(N_d)} \tag{5.12}$$

**Redundancy Score**

As it has been discussed in the introductory section (see section 5.3), the reduction of the visual redundancy of the original video is the core of the presented summarization approach. Such mechanism has been applied because it is potentially applicable for almost any kind of video content without requiring a priori knowledge about such content characteristics. For this redundancy removal purpose, the target of the system will be the maximization of the output summary coverage, avoiding the inclusion of too similar fragments in the output summary. Considering two nodes $N_a$, $N_b$ and their corresponding basic units $BU_a$ and $BU_b$, we define the visual distance between such nodes as

$$distanceN(N_a, N_b) = distance_n(BU_a, BU_b) \tag{5.13}$$

being $distance_n(BU_a, BU_b)$ the shot distance (described in section 5.5.2) if both $N_a$, $N_b$ are inclusion nodes, or 0 otherwise. Given a specific leaf node at level $d$, the mean distance between $N_d$ and all the nodes included in the tree branch ending in $N_d$ will be calculated as:

$$meanDistance(N_d) = \frac{\sum_{i=0}^{d-1} distanceN(N_d, N_i)}{includedN(N_d)} \tag{5.14}$$

It should be noted that the tree nodes corresponding already selected or discarded elements are included as well in the comparisons. Therefore, the computational complexity required for the calculation of the *meanDistance* value will grow with the number of processed (and selected) video fragments. In order to keep the performance of the process under controlled levels and taking into consideration the video content redundancy distribution issues described in section 5.3.1, the node comparison is carried out with a maximal amount of $p$ previous nodes, calculating the mean and minimum distance values as:

$$meanDistance_p(N_d) = \frac{\sum_{i=d-p}^{d-1} distanceN(N_d, N_i)}{includedN(N_d)} \tag{5.15}$$

$$minDistance_p(N_d) = min\{distanceN(N_d, N_i)\},\ d-p \leq i \leq d-1 \tag{5.16}$$

Both *meanDistance* and *minDistance* values are combined for calculating the final node redundancy value:

$$nodeDistance_p(N_d) = \frac{meanDistance_p(N_d) + minDistance_p(N_d)}{2} \tag{5.17}$$

Such distance, $nodeDistance_p$, takes into account the average visual distance of the current node, $N_d$, with all the previous nodes included in the summary but includes as well the minimum distance value found. In this way, in case a very similar node is already included in the summary, it will have an increased influence in the final node score.

Finally, the redundancy score for a given tree branch ending in node $N_d$ is iteratively calculated based on the parent node $N_{d-1}$ score as

$$parentR(N_d) = rScore(N_{d-1}) \cdot includedF(BU_{d-1}) \tag{5.18}$$

$$currentR(N_d) = nodeDistance_p(N_d) \cdot nF(BU_d) \tag{5.19}$$

$$rScore(N_d) = \frac{parentR(N_d) + currentR(N_d)}{includedF(BU_d)} \tag{5.20}$$

if $N_d$ is an inclusion node or $rScore(N_d) = rScore(N_{d-1})$ otherwise. Higher redundancy scores are obtained for summaries composed of dissimilar fragments, as the score calculation is based on the similarity distance between all the fragments included in the summary.

**Activity Score**

The previously depicted redundancy score aims to generate video summaries composed by varied fragments, from a visual point of view, trying to capture as many different events from the original video as possible. In this case, the activity score models the amount of variation within a specific video fragment, aiming to include high activity fragments in the video output (we consider that high activity fragments are more likely to contain events than static video segments). Given a tree node $N_d$ with an associated basic unit composed by $n$ frames $BU_d = \{f_1, f_2, ..., f_n\}$ the amount of activity for the node is defined as

$$activity(BU_d) = \frac{\sum_{i=1}^{n-1} CLDiff(f_i, f_{i+1})}{n} \tag{5.21}$$

and the activity score is iteratively calculated based on the score of the parent node $N_{d-1}$ as

$$parentA(N_d) = aScore(N_{d-1}) \cdot includedF(BU_{d-1}) \tag{5.22}$$

$$currentA(N_d) = activity(N_d) \cdot nF(BU_d) \tag{5.23}$$

$$aScore(N_d) = \frac{parentA(N_d) + currentA(N_d)}{includedF(BU_d)} \tag{5.24}$$

**Score Combination and Normalization**

Once the different scores are calculated, they must be combined in order to obtain the score for each possible summary (i.e. summarization tree branch). The scoring model implemented in the summarization tree algorithm allows to define independent weights for each individual score considered. Assuming that $m$ different scores $\{s_{d1}, s_{d2}, ..., s_{dm}\}$ are calculated for each tree leaf ($L_d^n$) and that they must be combined for the calculation of the summary score, $m$ weights should be considered $\{w_1, w_2, ..., w_m\} \in [0,1]$ (in this case, considering size, redundancy, continuity and activity scores, $m = 4$).

On each iteration over the summarization tree, the maximum and minimum values for each individual score $s_i$, $max(s_i)$ and $min(s_i)$, are calculated for the complete set of tree leafs $L_d^i$ and are applied for normalizing each leaf score. For every leaf contained in the tree, its final score will be calculated as

$$score(L_d) = \sum_{i=1}^{m} w_i \cdot \frac{s_{di} - min(s_i)}{max(s_i) - min(s_i)} \tag{5.25}$$

obtaining the final score that will be applied for the tree branch selection and pruning processes.

The proposed weighted scoring provides a mechanism for applying an arbitrary number of combined individual scoring measures, each one with the desired associated weighting. As the different scores are individually normalized on each iteration, there is no need to control their boundaries, which could vary with different types of original content, and the proposed scoring mechanism adapts itself to the particular characteristics of each processed video.

### 5.5.4 Branch Pruning

The branch pruning step consists on the elimination (or not expansion) of specific sub-trees included in the generated summarization tree. In this case, the branch pruning is applied in two situations: for limiting the number of branches evaluated in the constructed partial summarization trees and for content filtering purposes. The number of branches control aims to speed-up the processing, avoiding to evaluate too many paths in the summarization tree. Such process relies on the previously defined scoring algorithm: on each tree iteration, all the leafs of the current summarization partial tree are scored and, given a branch limit, $l$, the $l$ leafs with higher scores are kept. All the non selected leafs and their corresponding branches are eliminated.

On the other hand, a branch pruning approach can be applied as well for the elimination of those branches containing inclusion nodes corresponding to non desired *BU*s (that is, the summary represented by the tree branch includes non desired video fragments). For this purpose, every time a new *BU* is added to the tree, it is analyzed and in case it is considered as undesired no inclusion nodes are appended to the tree for such BU, including discard nodes only (see the tree generation mechanism on section 5.5.1). This mechanism yields to a tree generation process in which no branches including undesired fragments are generated. An example of tree pruning is described in the submission presented to the TRECVid 2008 BBC rushes summarization task (see section 5.6 for more details).

## 5.6 Results of the TRECVid BBC Rushes Summarization Tasks

This section details submissions to the TRECVID 2007 & 2008 BBC Rushes Summarization Tasks [147, 133] . The TRECVid BBC Rushes Summarization task consisted on the generation of videos

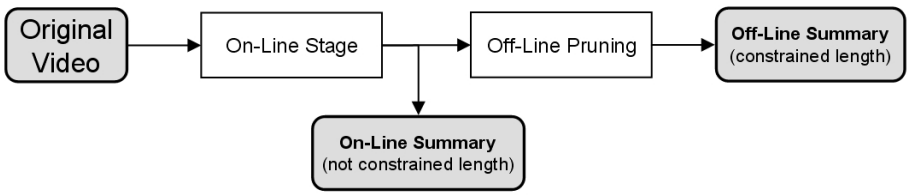


Figure 5.12: Overview of the TRECVid 2007 Abstraction System

summaries of unedited TV shows video content, rushes, which are characterized by containing a high amount of repeated takes as well as junk content, such as blank frames or test patterns. Each participant submitted summaries for the complete set of test videos (42 for the 2007 campaign and 40 for the 2008 one) which were evaluated by human assessors which scored several characteristics of the summaries such as the number of included events, the tempo and rhythm of the summaries or the perception of redundancy in the video (see chapter 6 for more details). The TRECVid campaigns did not consider the audio for the evaluation of the summaries.

In the following subsections, details about the presented submissions (generated with the *on-line* algorithms described in sections 5.4 and 5.5) together with the obtained evaluation results are provided.

### 5.6.1 BBC Rushes 2007 Submission

The algorithm presented to the 2007 BBC rushes evaluation campaign [147] is based on the algorithm described in section 5.4, a 'sufficient content change' approach originally published in [57]. The proposed approach was initially developed a completely *on-line* abstract generation system which provided a fast way to generate a base video abstract. Nevertheless, given the abstract length limitation requirement of the contest (the summaries length limit was 4% of the original video length), and the difficulties to control the output abstract length with the proposed *on-line* approach, the method was combined with a final abstract reduction step. Such step was applied in those cases in which the target summary length exceeded the limit. In both steps the abstraction mechanism rely on the reduction of the original content visual redundancy with methods based on color similarity features. The combination of these two-step summarization processes provided a very fast method with low memory usage for the creation of the video abstract. Although the application of the second stage makes the abstraction system, originally implemented as an *on-line* approach, to become *off-line* in several cases, the pruning stage was applied only in 8 out of the 42 videos evaluated and only for a small reduction of the resulting abstract length. Therefore, the obtained results provided useful information about the characteristics of *on-line* generated summaries when compared with the rest of the participants (*off-line* approaches).

Figure 5.12 shows an overview of the proposed summarization system which is mainly divided in a first on-line stage, where the video summary is generated on the fly while it is being read based on a sufficient content change approach, and an optional *off-line* pruning stage aimed to reduce the size of the on-line generated summary in case it exceeds the requested limitations. Both stages are based in frame and shot similarity measures defined in section 5.4.1. The details of the *on-line* stage are those described section 5.4.

For those cases in which the summary generated in the *on-line* stage exceeds the length limit, a pruning process is carried out by calculating all the *MeanDiff* values (see definition in equation

| *MeanDiff* Threshold | On-Line Time | Off-Line Time | On-Line Length | Off-Line Length |
|---|---|---|---|---|
| 11 | 35.61 | 0 | 2.91% | 2.91% |
| 9 | 37.12 | 0 | 2.74% | 2.74% |
| 7 | 42.01 | 5.12 | 5.3% | 3.95% |
| 5 | 102.86 | 215.25 | 34.25% | 3.9% |
| 3 | 182.9 | 1062.29 | 83.44% | 3.97% |

Table 5.1: Summarization Stage Times and Output Lengths.

| | DU | XD | TT | VT | IN | EA | RE | Effort |
|---|---|---|---|---|---|---|---|---|
| Average | 50.5 | 9.3 | 93.1 | 51.8 | 0.48 | 3.1 | 3.65 | 3291.8 seconds. |
| UAM | 46.6 | 13.2 | 92.3 | 50 | 0.47 | 3.3 | 3.7 | 53.3 seconds. |

Table 5.2: TRECVid 2007 BBC Rushes Summarization Evaluation Results.

5.4) between all the subsegments selected by the *on-line* stage. Iteratively, each pair of subsegments with the lowest *MeanDiff* (i.e. more similar) are selected and their intra *MeanDiff* value is calculated. The subsegment with the lower value (the one with less variability and, hence, the less informative one) from each selected pair is eliminated. This process is iteratively repeated until the selected segments total length is below the established limit. This step introduces an *off-line* processing stage and, therefore, in the cases in which it must be applied, the system can not be considered an *on-line* approach. Nevertheless only 8 from the 42 test videos required the *off-line* pruning stage.

**Performance & Results**

Table 5.1 shows an example of different *MeanDiff* threshold values applied to one of the TRECVid 2007 BBC Rushes Summarization Task content set videos, the required *on-line* processing time and the *off-line* pruning time in case it is needed (without considering video coding and decoding times). The obtained summary length produced by each stage is depicted as well. Lower threshold values involve a higher number of included segments in the *on-line* stage and, therefore, a greater summary reduction must be carried out in the *off-line* pruning stage except for those cases where the summary length after the *on-line* processing is already below the 4% limit. The *off-line* pruning stage has a heavy computational complexity, specially when compared with the *on-line* stage. This can be observed in the required *off-line* processing times when the *on-line* stage outputs too long summaries.

Table 5.2 shows a comparison between the proposed approach and the average evaluation results of all participant's submissions to the TRECVID 2007 BBC rushes evaluation [147].

The results are obtained for the 42 test BBC rushes by setting a *MeanDiff* threshold in the *on-line* stage trying to approximate the output length to the 4% limit. In this case, the knowledge about the original content characteristics, extracted from the training set, allows to experimentally set such parameter. Only 8 of the test videos exceeded the 4% limit after the *on-line* stage and required the subsequent *off-line* pruning. Nevertheless, one of the consequences of setting conservative parameters in the *on-line* stage to reduce the number of summaries exceeding the length limit, is the low duration of the resulting summaries. As can be observed in table 5.2 –DU-, the duration of the summaries is below the average values with a significant difference between the obtained and target summary durations (table 5.2 –XD-).

The rest of the evaluation results, total time judging, total time video play and inclusion rate (TT,

VT, and IN) are very close but slightly under the average values obtained by the rest of participants, while the easiness for understanding the video (EA) and the duplicate video indicator (RE) are slightly above the average. The rate of inclusion in the summary results could be easily improved by setting less restrictive *on-line* stage parameters which would produce a higher number of summaries exceeding the 4% summary length limit and, after the *off-line* pruning, a slightly under 4% length summaries. Nevertheless, the purpose of the proposed approach was to demonstrate the feasibility of applying an *on-line* summarization approach to obtain results comparable to *off-line* methods. Even with the restrictions imposed to the *on-line* system, the obtained results are very close to the average values obtained by the other participants in every category, while the effort of the proposed system (seconds required for the generation of the video summaries) is clearly below other participants results (see table5.2 –Effort-). The performance difference of the proposed method with respect to other participants is huge. When the system is individually compared with the fastests participants, a very similar efficiency is found while the proposed *on-line* abstraction approach obtained better evaluation results. On the other hand, the systems which obtained better evaluation scores required a processing time several orders of magnitude above the proposed method one.

### 5.6.2 BBC Rushes 2008 Submission

The results obtained in the TRECVid 2007 BBC rushes summarization task (see previous section) demonstrated the possibility of generating high efficiency *on-line* video summaries with results comparable to *off-line* approaches. Nevertheless, the proposed system was not completely *on-line*, requiring an *off-line* pruning stage for avoiding to exceed the 4% limit (in the TRECVid 2008 evaluation campaign the length limit was reduced to 2%). Building a completely *on-line* system with a more sophisticated summary length control was one of the main objectives after the TRECVid 2007 participation. Several aspects of the abstraction process were identified as key issues to be improved in further developments: image and shot comparison mechanisms, summary smoothness control mechanisms (the generated summaries included unpleasant discontinuities) and junk removal mechanisms. The proposed algorithm relies on the summary trees generation algorithm, described in section 5.5. Such algorithm provides mechanisms for controlling the length, redundancy, continuity an activity of the generated video summaries as well as filtering mechanisms based on branch pruning, applied for junk removal, eliminating test patterns and clapboard fragments from the original content.

Figure 5.13 depicts the summarization approach overview. The first summarization step consists on the split of the original video in fixed size BUs (25 frames -1 second-, usually considered as the minimal length needed by an human to recognize visual content [64]). Afterwards, the *BU*s are processed by an initial redundancy reduction step: an adaptive frame dropping mechanism aimed to reduce the number of segments in the video with very small variation between consecutive frames (i.e. static sequences) while keeping the original *BU* length for high variation segments. The maximum drop rate is set to 1/2 per *BU* and hence a maximum 2-times speed increment can be achieved with this approach in case of dealing with low variation sequences.

Once the original BUs are accelerated, they are iteratively inserted in a summarization tree with a maximum $d = 90$ depth. Therefore, the established generation delay is 90 seconds (1-second BUs are used) and a maximum of 1000 or 1500 leaf nodes (depending on the submitted run) are kept on each iteration. The number of subtree branches is controlled by the selection of those paths within the generated tree with higher scores. In the submitted approach the scoring mechanism described in section 5.5.3 was applied with the exception of the redundancy metric described in equation 5.17,

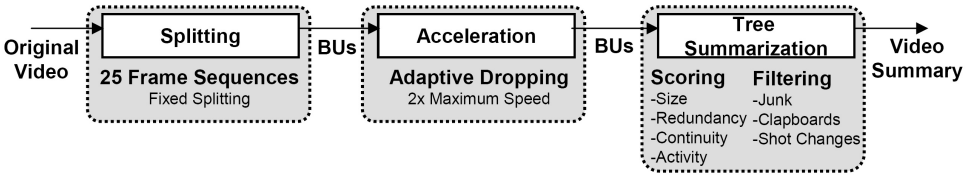


Figure 5.13: Overview of the TRECVid 2008 Abstraction System

$nodeDistance_p(N_d)$, which is in this case defined as:

$$nodeDistance_p(N_d) = meanDistance_p(N_d) \tag{5.26}$$

The branch pruning mechanism designed for filtering undesired video fragments (see section 5.5.4) was applied in the TRECVID BBC rushes summarization task by including in the system three different content filters:

- *Junk Detection*: For the detection of video segments constituted by blank frames and test patterns, a fast variation measure has been defined. It makes use of the 8x8 thumbnail used for the calculation of the MPEG-7 Color Layout of each received frame. The squared differences of each thumbnail pixel and the surrounding pixels are calculated and averaged. It has been experimentally checked that the obtained values for plain color frames are significatively higher than in the case of natural images while test patterns produce the highest values. These experimental observations allowed to set two thresholds, namely, maximum and minimum values for considering an image as 'natural' content.

- *Clapboard Detection*: For the detection of clapboards on video frames two Haar cascade object detectors [157] for white and black clapboards were trained with about 1500 example frames; detection rates over 95% were achieved with the training set. Nevertheless, the high variability in the position, size, rotation and illumination of the clapboards in the test content set would surely produce a non quantified reduction of the detectors' precision. Although the detection rates were not formally evaluated, the junk removal results obtained in the evaluation, described in the following subsection, validate the applied mechanism.

- *Shot Change*: A basic shot change detection filter was added to the system in order to avoid the inclusion of video *BU*s including shot changes (which could produce unpleasant cuts in the output summaries); a simple shot detection was implemented by the calculation of the maximum variation between consecutive frames in a video sequence and the variation between the beginning and end of the video fragment (making use of the Color Histogram distance measure).

**Performance & Results**

Two runs were sent for evaluation in the TRECVID 2008 BBC rushes summarization task [133] with different summarization parameters. The machine used was a Pentium Xeon @ 3.7 GHz with 3 Mb of RAM. Both runs, *Run1* and *Run2* (submitted as GTI_UAM.1 and GTI_UAM.2), were generated with a maximum tree depth of 90 nodes and a maximum of 1500 nodes by iteration for *Run1* and of 1000 for *Run2*. In both cases the summarization performance (120s. on average for *Run1* and 99s. for *Run2*)

outperformed most of the other systems (4879s. effort on average) and only one of the baselines (cmubase3 -17,2 s.-), based on a sumbsampling approach, was faster than the proposed system.

| Run | DU | XD | TT | VT | IN | JU | RE | TE |
|---|---|---|---|---|---|---|---|---|
| Run1 | 31.2 | 0.5 | 45.1 | 33.1 | 0.55 | 3.27 | 2.97 | 2.71 |
| Run2 | 31.1 | 0.5 | 47.7 | 33.5 | 0.56 | 3.32 | 2.96 | 2.62 |
| Avg. | 27 | 4.5 | 41.4 | 29.0 | 0.46 | 3.15 | 3.2 | 2.72 |

Table 5.3: TRECVid 2008 BBC Rushes Summarization Evaluation Results.

With respect to the summarization score weights, *run1* was configured for a balanced summary generation -*wSize= 0.475, wRedundancy= 0.21, wContinuity= 0.120, wVariation= 0.195*- while *run2* was configured focusing in the size and redundancy control -*wSize= 0.6, wRedundancy= 0.350, wContinuity= 0.050, wVariation= 0*-. Table 5.3 depicts the results obtained for the two runs and the average values for the complete set of participants including the baseline summaries. The duration -DU- and time difference with the target -XD- metrics show that the output size control incorporated in the proposed system works very well (very close to the 2% length target) which is a very significative result considering a fully *on-line* approach without information about the length of the incoming video. The obtained lengths are higher than average probably because the 2% may have been considered as a limit and not a target length in many of the submitted runs. The metrics related to the judging time -TT, VT- are slightly higher than the average, an expectable result as the obtained summary lengths are higher than the average and there is a strong correlation between the DU/TT (correlation coefficient = 0,84) and DU/VT (corr. coeff.=0,98) parameters. Regarding the junk metric -JU- the obtained results are over the average proving that the mechanisms for junk shots filtering incorporated in the system works well. Both runs obtained very good results in the inclusion rate -IN-, which are clearly over the average results, paid back in the perceived redundancy -RE- and tempo -TE- of the summaries which are slightly under the average results. Figure 5.14 shows the high correlation between IN/RE and IN/TE measures for all the participants' results and the proposed algorithm results. It should be considered that the video fragment acceleration would probably had positive impact in the IN rate but negative in the summary tempo/rhythm RE.

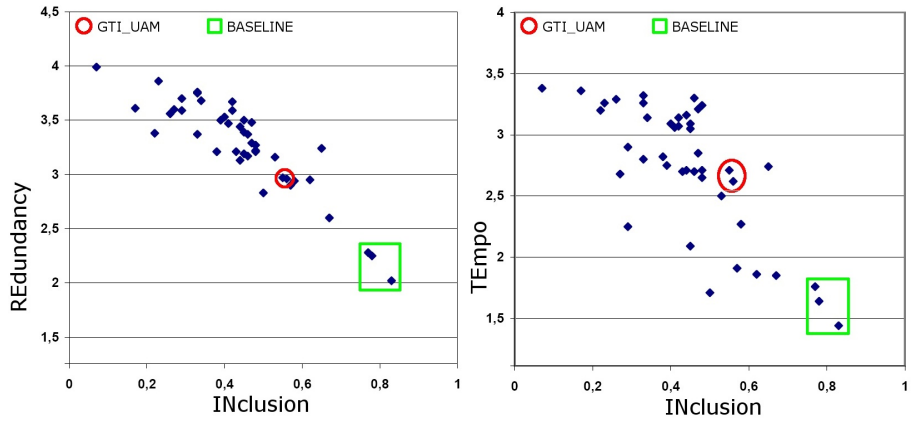


Figure 5.14: Inclusion vs. Redundancy/Tempo

## 5.7 Conclusions

In the first part of this chapter, the foundations of the *on-line* visual redundancy removal approach have been presented, explaining the reasons to consider the *on-line* redundancy removal abstraction approaches as potentially comparable to *off-line* systems, due to the distribution of the visual redundancies in typical video content.

In sections 5.4 and 5.5, two *on-line* abstraction approaches, fulfilling the operative constraints described in chapter 4, have been described in increasing order of complexity and included features. The first approach, presented to TRECVID 2007 BBC rushes evaluation task consists on a 'sufficient content change' approach (section 5.4). Although such system had some limitations, such as the summary length control mechanism, it demonstrated the feasibility of applying an *on-line* solution obtaining results comparable to *off-line* approaches in the TRECVid 2007 summary evaluation campaign.

The experience accumulated with the development and testing of the first abstraction approach served as the basis for the development of the binary tree based summarization algorithm (section 5.5), an innovative and generic *on-line* summarization approach which allows the combination of any summarization criteria with the application of different scoring functions. The described algorithm (based on progressive summarization sub-tree generation) provides a generic mechanism for *on-line* video summarization in which the performance of the summarization process, in terms of summary generation delay and processing time, can be controlled. It is possible to configure the algorithm for faster and lower memory consumption runs (considering smaller trees and lower number of evaluated branches) or for more accurate summary generation modes (deeper trees with higher number of branches).

The participation details in both TRECVid BBC Rushes Summarization Task in 2007 and 2008 are described in section 5.6, including implementation details of the complete systems, developed based on the proposed *on-line* abstraction algorithms, as well as the obtained evaluations. Such results prove that the proposed algorithms are able to obtain results comparable to other participants' proposals while applying efficient and customizable *on-line* approaches.

In the following chapters a framework for the automatic evaluation of video summarization (chapter 6) systems will be described and the two algorithms presented in this chapter will be in-depth analyzed and evaluated (chapter 7).

# Part IV

# Evaluation

# Chapter 6

# Automatic Evaluation of Video Summaries

## 6.1 Introduction

In 2007 and 2008 the TRECVid BBC Rushes evaluation campaigns were carried out as the first attempt of a high scale evaluation of automatic video summarization methods. Such campaigns permitted the evaluation and, hence, the comparison of different approaches for video summarization. Nevertheless, once the evaluation campaigns are over, it is not possible to compare the new developments with previous works. In this chapter, we will describe an automatic system for the evaluation of video summaries which aims to emulate the results provided by the TRECVid BBC rushes evaluation campaigns, making use of the same data sets and evaluation measures (including both subjective and objective evaluations). The proposed approach makes use of the 2008 original participants' submissions and their corresponding results for training different predictors for each considered evaluation measure.

This chapter is organized as follows: the TRECVid BBC Rushes Evaluation Campaigns are briefly described in section 6.2. Section 6.3 describes the feature extraction (subsection 6.3.2) and predictors training and results (subsection 6.3.3). Finally, conclusions are drawn in section 6.4.

## 6.2 TRECVid BBC Rushes Evaluation Campaigns

The TRECVid 2007 and 2008 BBC rushes evaluation campaigns [133, 147] represented the first attempt for carrying out a large-scale evaluation of video summarization systems. Several previous works on video summary evaluation are depicted in [147] where the approaches are classified as extrinsic or intrinsic: *'Some are extrinsic, i.e., in terms of how a summary helps in some task, rather than intrinsic i.e., direct evaluations [...]´*. The enumerated extrinsic approaches include, among others, [158] which evaluated slideshow summaries and [159] evaluating video skims for fact-finding and gisting tasks. The intrinsic approaches included approaches like [160], where 'neutral observers' determined the number of missing or redundant frames on video summaries, [9] which deals with the evaluation of soccer content video summaries, and [161] which included both extrinsic and intrinsic evaluations. However, existing works in the literature are generally applied to a reduced content set and based on the developments of a single research group. The TRECVid BBC Rushes Summarization Tasks in 2007 and 2008 provided a large video database, an uniform method for creating the ground truth and a uniform scoring mechanism. In this chapter, we will focus on the 2008 campaign [133],

where the experiences learnt in 2007 were applied, and contains several differences in the evaluation measures with respect to the previous year campaign.

The task proposed to the participants was to generate a video summary of the original content by removing redundant or unclear footage from BBC unedited footage (rushes), shot for five different drama series. Given the high redundancy of the original video content as well as the amount of junk it includes (e.g., test patterns, clapboards), it was established that the summaries should be no longer than 2% the original video duration and should be presented in a MPEG-1 file (without any specific encoding conditions), to be displayed during the evaluation using the original video frame rate. Participants had to generate video summaries for 40 original rushes videos and their quality was evaluated by 3 human assessors hired for such purpose, with the following subjective and objective measures:

- Objective:
  - *Assessment Time*: Time taken by the assessors to determine the presence/absence of desired fragments.
  - *Duration* (DU): Duration of the summary relative to the 2% target length.
  - *Effort*: Elapsed time for summary creation.
- Subjective:
  - *Inclusion* (IN): A groundtruth was created by the organization identifying video segments from the original video which should be included in a good summary: the decision about such inclusion was based on the events in the segments. The inclusion measure indicates the percentage of such segments (and therefore, events) the assessors consider to be included in the output summary.
  - *Junk* (JU): Subjective perception by the assessors of the amount of junk such as color bars, clapboards or empty frames, included in the summary in a scale from 1 to 5 (5-point Likert scale indicating the assessor's agreement with the statement '*This summary contains many color bars, clapboards, all black or all white frames*').
  - *Redundancy* (RE): Amount of redundancy (in terms of nearly identical fragments included) perceived in the summary in a 1 to 5 scale (5-point Likert scale indicating the assessor's agreement with the statement '*This summary contains many nearly identical segments*').
  - *Tempo* (TE): Satisfaction in the tempo and rhythm of the presentation in a 1 to 5 scale (5-point Likert scale indicating the assessor's agreement with the statement '*This summary is presented in a pleasant tempo and rhythm*').

In this work, we focus in the subjective measures redundancy -RE- and tempo -TE- as well as the groundtruth inclusion metric -IN- which, although based in a groundtruth event list, is influenced by the assessors subjectivity. Such three measures have been selected because they can reasonably model the quality of a summary and, although two of them are clearly subjective, we hypothesized that they depend on quantifiable characteristics of the video summaries. The rest of the measures have not been considered as some of them are impossible to reproduce (assessment time), straightforward to extract (summary lengths), too domain specific (junk perception), or require running the original summarization algorithms (effort). However, all of those metrics, except the assessment time,

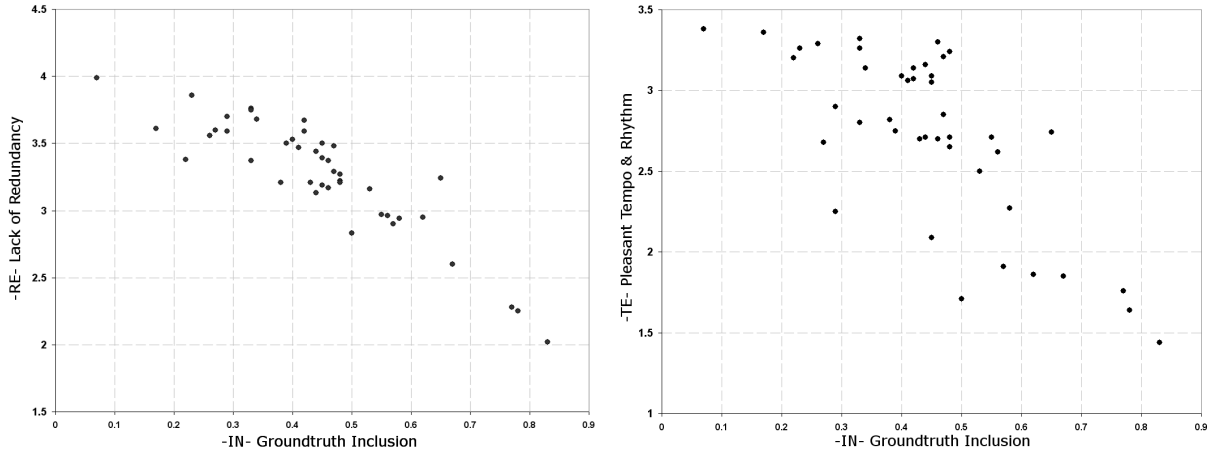


Figure 6.1: IN - RE - TE Comparison

can be extracted with different techniques and included, if needed, in an automatic evaluation approach.

31 teams submitted summaries of the 40 videos, being possible to submit up to two different runs. The total amount of submissions was 43, including three baselines by CMU [134]. With respect to the three measures selected for our study (inclusion -IN-, redundancy -RE- and tempo -TE-), it was found that, in all the proposed approaches, such measures presented a high correlation. The obtained per run average values for IN ranged from 0.07 to 0.83, the RE average results ranged from 2.02 to 3.99 and, finally, the TE mean results ranged from 1.44 to 3.38. Figure 6.1 shows the representation of the average obtained IN values with respect to the RE ones (correlation coefficient -0.88) and with respect to the average TE values (correlation coefficient -0.74). With independence of the applied approaches, the results show how it seems difficult to obtain high IN values while keeping high RE and TE scores and vice versa, and the differences between the presented approaches may rely in analyzing the equilibrium between such measures. More details about such differences, evaluation process and the presented approaches can be found in [133, 147].

## 6.3 Automatic Summary Evaluation

### 6.3.1 Introduction

The 2007 and 2008 TRECVid summary evaluation campaigns established a common scenario for the evaluation and comparison of different video summarization techniques. Although the type of content was very specific, it was very appropriate for the evaluation of automatic video summarization approaches given its high redundancy. Moreover, the summarization task target, with its restrictive target length, was tough enough to measure the performance of the presented systems under very constraining conditions. Nevertheless, once the evaluation campaigns have finished, it is not possible to compare the results of improved or newly developed techniques with existing (and previously evaluated) approaches.

A possibility for further evaluations and comparison of new methods is to reproduce the evaluation process carried out in the TRECVid campaigns. In [162], DCU researchers presented a rushes

summarization system together with an evaluation carried out by re-running the TRECVid 2007 summary evaluation process. The two baselines used in the TRECVid 2007 were newly evaluated, obtaining results coherent with the original TRECVid measures, so authors were able to validate the re-evaluation process and measure the improvements of the new proposed summarization algorithm. Nevertheless, such kind of human evaluation process requires a big amount of time and effort, especially if researchers aim to carry out the evaluation process with the TRECVid 40 original videos and several summarization approaches or different runs.

In order to deal with such limitation, the development of automatic evaluation approaches is highly desirable. Several TRECVid participants carried out efforts for the automation of part of the evaluation process, focusing on the event inclusion -IN- measure. As it has been previously described, such measure aims to determine the amount of events, defined in a manually annotated groundtruth, included in the video summaries. CMU researchers, in charge of developing the baselines for the evaluation campaigns in 2007 and 2008, carried out a manual annotation of the starting and ending times of every groundtruth event and estimated the IN measure by checking the overlapping between the segments included in the summary and the annotated ones [64] with an established minimal length of the video summary segments of one second. The estimated IN values and finally obtained results presented a correlation coefficient of 0.67. In [95], authors carried out an experiment with one of the videos of the 2007 campaign for estimating the required overlapping between a groundtruth event and a summary segment to consider an event as included (the optimal values were found between 2 and 22 frames). In [163] and [164] authors continued their work on automatic IN prediction by extending the original approach. In this case, the method for determining if a groundtruth event was included in the output summary was improved by adding more characteristics, apart from the start and end times, to the set of groundtruth events. Such extended information included, for example, the length or activity of the different events. The number and length of the segments included in the video summary corresponding to every original event are computed as well, completing a feature vector which is used for determining if the events were included or not. Different machine learning approaches were tested using 8 original TRECVid videos and 10 summarization systems for training and evaluation, and they obtained a maximum IN measure correlation of 0.88. The depicted approaches focus in the individual video estimation of the IN measure only, relying on the manual annotation of the original videos for matching the generated summaries included segments with the annotated groundtruth events starting and end times. A different approach can be found in [165], where an automatic summary evaluation method is described. Such method extracts the so-called *coverage*, *conciseness*, *coherence* and *context* metrics, related to the IN, RE and TE measures of the TRECVid campaigns, by the automatic comparison of generated summaries with a manually created reference summary. Authors' carried out a test manually re-evaluating fragments of part of the TRECVid 2007 original videos and submissions and obtained interesting correlations between the automatically extracted and subjective metrics.

In this work, we propose a novel approach for the training and application of automatic IN, RE and TE predictors for complete summarization systems and it is differentiated from previous works for different reasons:

- The system aims to approximate the evaluation results of complete evaluation systems and not only individual videos.

- The chosen measures include, apart from the IN measure, the RE and TE measures which are very relevant to determine the quality of the summarization approach. Although such mea-

sures are quite subjective, in this work it will be demonstrated that they can be predicted.

- The prediction system relies on visual analysis techniques only, extracting features for a further machine learning process, and does not require manual annotation of the original video events or creation of reference summaries.

The proposed approach for the prediction of the IN values consists in the automatic comparison of the video summaries with their corresponding original videos, by means of image distance metrics, and the extraction of different statistics from such comparisons for a further training of the predictors. For the prediction of the RE and TE values information extracted only from the summaries is taken into account. The summaries fragments are analyzed and compared for the extraction of significant features for the estimation of their redundancy and tempo. Once the different set of features are computed, they are used together with the evaluations obtained in the TRECVid 2008 task for training different predictors for each feature. In the following sections the features extraction, training and testing processes will be described.

### 6.3.2 Feature Extraction

The complete data set used for the development, training and validation of the automatic predictors is composed by the 39 test videos from the TRECVid BBC rushes summarization task (one of the videos from the original 40 videos set was eliminated from the evaluation process by the organization due to problems for defining a groundtruth) together with the summaries submitted by 30 participants (including the baselines), some of the submitting two runs. From the complete set of submissions, some of them were discarded for the analysis and training processes because of their specific characteristics. DCU [107] and EURECOM [166], which summaries presentation was composed by multiple windows and overlay information, were eliminated from the content set for avoiding distortions that such characteristics could introduce in some of the extracted features which, as described later, rely on image comparison techniques. The second run from Joanneum Research approach [108] and the Tokyotech submissions [112] were discarded because they included several one and two second summaries, including blank frames ones, a too small amount of information for extracting significative statistics from the summaries.

Some other submissions contained small texts overprinted in the image, but they were kept as they were considered to introduce a negligible distortion in the comparisons. The selected final content set contained 38 different submissions.

From the set of original videos and corresponding video summaries, a number of different features were extracted for the training of the individual predictors. Figure 6.2 shows an overview of the feature extraction process: in a first stage several feature matrices were directly extracted from the original content while the final set of features applied for the individual training of the IN, RE, and TE predictors were extracted from those previously calculated feature matrices. Both feature extraction steps are described in the following subsections.

#### Directly Extracted Features

The average original video was 26.6 minutes length, that is, an average amount of almost 40000 frames per video. Considering that the target length of the summaries was 2% the original video length, it can be estimated that they could reach an average of 800 frames each. As it will be later described, part of the feature extraction relies on a comparison between the frames included in the summary and

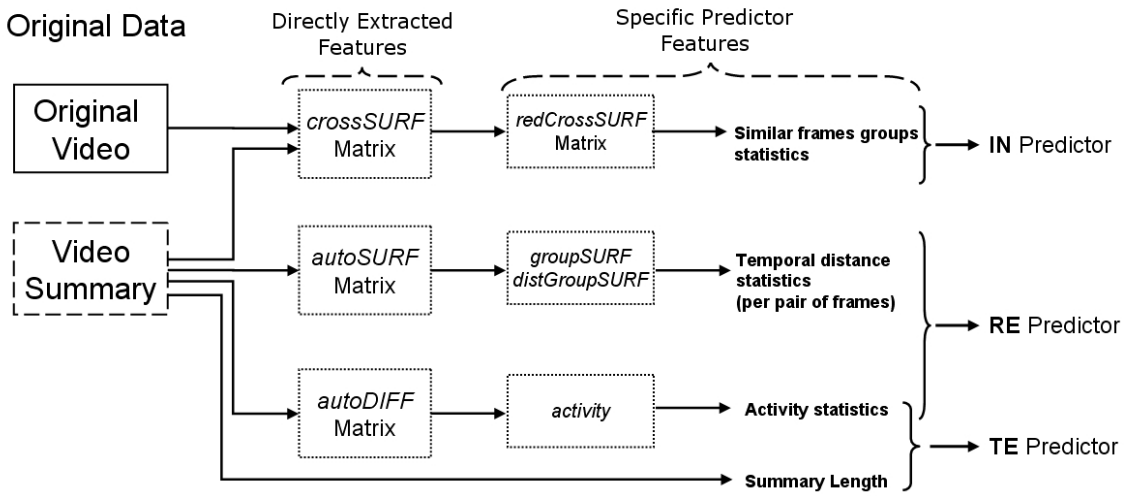


Figure 6.2: Feature Extraction Overview

the original video length and for that reason, in order to speed up the feature extraction process, the amount of information was reduced: the frames of the original videos and summaries were extracted and resized to 80x80 thumbnails (from the original 352x288 resolution). During the extraction process, the original videos were subsampled at 3 frames per second (averaging about 4800 extracted frames per original video) while the complete set of frames of each summary was kept.

The next step was to compute difference matrices between the original and summarized videos. We will define the video difference matrix between two videos $V_1$ and $V_2$ as *vDiffMatrix*, where each position $i, j$ of the matrix is calculated as

$$vDiffMatrix(V_1, V_2)_{i,j} = imDifference(V_1(i), V_2(j)) \tag{6.1}$$

where $v_x(i)$ corresponds to the $ith$ extracted frame from video $x$ and *imDifference* corresponds to the selected (among the suitable ones) image distance metric. In this case, we aimed to compare heterogeneous summaries from different research groups with the corresponding original videos. It was observed that the coding process or parameters varied from one group to another, producing color and quality variations in the videos. For this reason, the applied distance metric was chosen targeting to reduce such differences as much as possible. The huge amount of data to process was a restricting issue, therefore a fast as possible comparison method was required. The applied image distance metric was based on Speeded Up Robust Features -SURF-[167], a scale and rotation invariant interest points detector. The SURF features extraction was carried out on every extracted frame from the original videos and summaries. For each image $i$ of a video, $V_x$, the SURF extraction process, $SURF(V_x(i))$, generates a number, $n$, of interest points from the luminance channel of the image,

$$SURF(V_x(i)) = \{IP_{i,1}, IP_{i,2}, ...IP_{i,n}\} \tag{6.2}$$

and each interest point, $IP$, is defined by a vector composed of a fixed number, $c$, of coefficients (which may vary depending on the configuration of the SURF extraction process)

$$IP_{i,n} = \{f_1, f_2, ..., f_c\} \tag{6.3}$$

We will denote the euclidean distance between two feature vectors $IP_A$ and $IP_B$ as $d(IP_A, IP_B)$. For the comparison of two images, $i$ and $j$, corresponding to videos $V_1$ and $V_2$, we compute the distances between all the interest points from image $V_1(i)$ and image $V_2(j)$, so each position $(a,b)$ in an interest point differences matrix, *IPDiffMatrix*, is defined as

$$IPDiffMatrix_{a,b}(V_1(i), V_2(j)) = d(IP_{i,a}, IP_{j,b}) \tag{6.4}$$

Assuming that *IPDiffMatrix* is composed by $n$ rows and $m$ columns (that is, $n$ and $m$ interest points have been found for images $i$ and $j$ respectively), we define $minRow$ as the distances vector corresponding to the minimum values found on each *IPDiffMatrix* row and the analogous $minColumn$ value for the columns

$$minRow_k(V_1(i), V_2(j)) = \min_{\forall l, 1 \leq l \leq m} \{IPDiffMatrix_{k,l}\} \tag{6.5}$$

$$minColumn_l(V_1(j), V_2(j)) = \min_{\forall k, 1 \leq k \leq n} \{IPDiffMatrix_{k,l}\} \tag{6.6}$$

The final distance metric applied, $imDifference_{SURF}$, is defined as follows

$$imDifference_{SURF}(V_1(i), V_2(j)) = \frac{\sum_{k=1}^{n} minRow_k(V_1(i), V_2(j)) + \sum_{l=1}^{m} minColumn_l(V_1(i), V_2(j))}{n+m} \tag{6.7}$$

With such defined distance metric, given a video summary, $V_{summary}$, and its corresponding original video, $V_{original}$, we denote $crossSURF$ as the difference matrix (see equation 6.1) between both videos calculated with the $imDifference_{SURF}$ image distance,

$$crossSURF(V_{original}, V_{summary}) = vDiffMatrix_{SURF}(V_{original}, V_{summary}) \tag{6.8}$$

and *autoSIFT* as the self difference matrix of a given summary video ,

$$autoSURF(V_{summary}) = vDiffMatrix_{SURF}(V_{summary}, V_{summary}) \tag{6.9}$$

Both matrices provide information for determining the amount of content from the original video represented in the summary (IN measure) and the redundancy of the summary video itself, that is, how much repeated information it contains (related with the RE measure). The image comparison based on interest points, invariant to rotation and scale, is very convenient for a comparison of a type of content like the BBC rushes with many shot repetitions with small changes in camera position, zoom or angle. Nevertheless, there are some characteristics that such image comparison techniques are not able to properly capture. For example, it was observed that, in the TRECVid campaigns, the video acceleration negatively influence the tempo -TE- perception of the evaluators. It was also noticed, that the proposed SURF based comparison is not able to 'capture', in many cases, such acceleration effects because the interest points are compared based on their vector of characteristics only and not based on their position and displacement in the images. For this reason, it was convenient to extract additionally alternative descriptors, aiming to capture the described effects. In order to provide more information for the training and prediction processes, an additional measure, focused on capturing the differences between consecutive frames, is calculated. Given to images, $I_1$ and $I_2$, with

three color planes each ( $I_1^Y, I_1^{Cb}, I_1^{Cr}$ and $I_2^Y, I_2^{Cb}, I_2^{Cr}$), with pixel values between 0 and 255, the image plane difference is calculated as

$$diffPlane(I_1, I_2) = \frac{abs(I_1^Y - I_2^Y) + abs(I_1^{Cb} - I_2^{Cb}) + abs(I_1^{Cr} - I_2^{Cr})}{3 \cdot 255} \tag{6.10}$$

We will denote $histogram_{20}$ as the calculation of a single plane image 20 bins histogram (the number of bins was selected according to experimental results). Given a video, $V$, composed of $n$ frames $V(i)$, $1 \leq i \leq n$, each position $i$ of the finally calculated measure, *consecutiveDiffs*, is defined as the 20-bin histogram of the difference plane between consecutive frames

$$consecutiveDiffs(V)_i = histogram_{20}(diffPlane(V(i), V(i+1))) \; \forall i < n \tag{6.11}$$

Such measure captures the information about the activity between consecutive frames in the video. Instead of accumulating each single pixel variation, the histogram keeps the information about the distribution of such differences in levels of variation (how many pixels in the image have specific levels of variation). We will define the final extracted measure from every processed video summary as *autoDIFF*, which consists on a vector of difference histograms:

$$autoDIFF(V_{summary}) = consecutiveDiffs(V_{summary}) \tag{6.12}$$

The final set of extracted data applied for the training and prediction processes will consist of the $crossSURF(V_{original}, V_{summary})$ (equation 6.8), $autoSURF(V_{summary})$ (equation 6.9) matrices, as well as the just defined $autoDIFF(V_{summary})$ measure.

**Specific Predictor Features**

The feature matrices extracted from the original videos and summaries (*crossSURF*, *autoSURF*, and *autoDIFF* matrices, described in the previous section) represent the basic content information available for the further predictor training process. Nevertheless, the dimensionality of the data is too high to enable the practical application of machine learning techniques. For this reason, the information must be processed to extract the more significant features for its application in the different predictors training processes. In this case, as individual predictors for the IN, RE and TE measures have been developed, and given the different characteristics of the summaries such measures are related to, the applied features will necessarily be different for each predictor. In the following subsections, the final features computed from the previously extracted data are described. The selected features are based on rational reasons, although there were different feature extraction attempts which did not produce the desired results. For this reason, the final set of applied features is the result of a heuristic trial and error process, developed and tested in parallel with the testing and development of the the applied machine learning approaches. The chosen prediction mechanism, training and validation results are described in the next section.

**Inclusion -IN- Features** As previously defined (see section 6.2), the inclusion measure -IN- aims to determine the amount of groundtruth events, from the original video, included on each summary. For the automatic prediction of IN values, it will be necessary to compare the original and summary videos making use, in this case, of the *crossSURF* matrix. Such matrix includes the comparison value of the subsampled original video frames with all the frames in the summary. In an ideal case, the

distance between an original frame and the same frame included in the summary should be 0. Nevertheless, the coding format used by the participants for the submitted summaries is heterogeneous and the condition is not always fulfilled. Moreover, the original video frame set does not contain every original frame (original videos are subsampled at 3 frames per second). However, even with such restrictions, for those frames from the original video contained in the summary, it is always possible to find very low distance values in the *crossSURF* matrix. Therefore, assuming that it is possible to determine which frames from the original video are included in the summary, it is necessary a mechanism to compute how much different information such frames cover: as the original video is very redundant, it is not enough to determine how many frames are included, but to determine how many of them are different and what fraction of the total information contained in the original video they represent.

The first step was to reduce the size of the *crossSURF* matrix, eliminating positions corresponding to almost equal images. For this purpose the self difference matrix of the original video, that is $autoDIFF(V_{original})$, is used for checking every position,discarding those corresponding to frames very similar to already selected ones. An experimental $imDifference_{SURF}$ (see equation 6.7) threshold value *0.1* demonstrated to obtain good results. The *crossSURF* matrix is sequentially processed for the generation of a reduced version, *redCrossSURF*, appending every row $crossSURF_i$ to the *redCrossSURF* if it does not already contain a row $crossSURF_j$ with $j < i$ so $autoDIFF(V_{original})_{i,j} < 0.1$. Finally, the minimum value for each original video frame comparison is kept, producing the *minCrossSURF* vector.

$$minCrossSURF_i = \min_{\forall j} redCrossSURF_{i,j} \tag{6.13}$$

In other words, the *minCrossSURF* vector contains the distance between every non-repeated original video frame and its most similar summary frame.

The final step consists on counting the runs, defined as sets of consecutive values below a given threshold, included in the *minCrossSURF* vector. In this way, it is possible to determine how many positions are below a given threshold but also if such positions are grouped (located in adjacent positions) and, if so, the length of such groups. The runs are calculated for 20 different possible thresholds

$$threshold_i = (i \cdot 0.02)^2,\ 1 \leq i \leq 20 \Rightarrow 0.004 \leq threshold_i \leq 0.16 \tag{6.14}$$

The applied thresholds represent small image distance values (from 0.004 to 0.16), which correspond to very similar images. In this way the extracted runs statistics provide information about how much information, from the original video, is represented in the summary and how such information is distributed. For each different applied threshold the number, average length and variance of the obtained runs are appended are stored as features for the IN predictor. Such features aim to model what fraction of the original information is covered, but also how much of it is distributed in consecutive positions as long runs will be more likely to represent complete events from the original video.

**Redundancy -RE- Features** The redundancy measure -RE- is related to the subjective perception of repeated content in the generated summaries. In this case, we have made use of the *autoSURF* matrix, which codes the visual distance (computed with the $imDifference_{SURF}$ metric, equation 6.7) between all the frames in the summary, and the *autoDIFF* matrix (equation 6.12), that codes information about consecutive frames differences in the summary video. A highly redundant summary will

necessarily contain a higher amount of visually similar frames, and such information is contained in the described matrices.

As in the IN case, the amount of data is too huge for a direct treatment and must be reduced. In this case, making use of the *autoSURF* matrix, all the possible pairwise comparisons between frames in the summary are grouped according to their *autoSURF* matrix normalized value considering 20 bins (the number of bins was selected according to experimental results), providing a fine differentiation between the possible similarity levels :

$$groupSURF_k = \{autoSURF_{i,j} | \frac{k-1}{20} < autoSURF_{i,j} \leq \frac{k}{20}\},\ 1 \leq k \leq 20 \tag{6.15}$$

The temporal distance of all the pairs of frames which distance is included on every group is calculated and the average and variance of such distances are calculated for every possible group, $distGroupSURF_k$.

$$distGroupSURF_k = \{abs(i-j) \mid \forall autoSURF_{i,j} \in groupSURF_k\},\ 1 \leq k \leq 20 \tag{6.16}$$

In this way, the average distance between groups of frames with different levels of similarity is extracted. For a non redundant summary, $distGroupSURF_k$ will necessarily tend to: 1) not contain many values with high similarity, and 2) not contain very similar frames in distant positions; and such information is coded in the extracted features.

The *autoDIFF* matrix contains the histograms of pixel differences between consecutive frames in the summary and therefore it provides additional information: many frames with almost no variation represent static shots and very long static shots in a summary may be perceived as redundant. In this case, an activity metric is computed for every *autoDiff* matrix position from the *autoDIFF* histograms by calculating the ratio between the amount pixels included in the upper half levels of the histogram (that is, high variation pixels) with respect to the total pixels of the image. If we denote each histogram level as $l$ in a given position, $i$, of the *autoDIFF* vector as $autoDIFF_{i,l}$, we will define the activity metric as

$$activity_i = \frac{\sum_{l=11}^{20} autoDIFF_{i,l}}{\sum_{l=1}^{20} autoDIFF_{i,l}} \tag{6.17}$$

From the obtained curve, the average and variance values are extracted and added as features for the predictor training process. Apart from the average and variance values, some different parameters of the activity curve were tested in order to feed the predictor with additional information about the video summary activity distribution. The selected features were the average of the activity values above and below the mean which were included in the predictor feature set.

$$globalAverage = average(\{activity_i, \forall i\}) \tag{6.18}$$

$$supAverage = average(\{activity_i | activity_i > globalAverage\}) \tag{6.19}$$

$$infAverage = average(\{activity_i | activity_i < globalAverage\}) \tag{6.20}$$

$$globalVar = variance(\{activity_i, \forall i\}) \tag{6.21}$$

**Tempo -TE- Features** In the TRECVid participants submissions, it is possible to observe that the tempo & rhythm measure -TE- seems to be related with two main factors: the activity in the summary, and the number and rate in which shot changes appear in the summary. High variation/activity shots can be found in video segments capturing actions or, for example, in cases of abrupt camera movement (not common in edited video but easily found in the rushes content), as well as in many of the participants approaches who applied video acceleration aiming to increase the -IN- scores. Summaries composed by very short shots (including, therefore, many shot changes or cuts) tend to obtain lower RE scores as well. As in the case of the RE extracted features, it is possible to make use of the *autoDIFF* information, which codes the histogram of pixel differences between consecutive frames, for the measurement of such summary characteristics. The activity metric calculated as part of the RE features processing (see equation 6.17 on previous subsection) is applied as well in this case, including as features for the TE predictor the values obtained from equations 6.18 to 6.21 together with the length of the generated summary as an additional feature to determine relations between the length and activity statistics of the summary and the perceived tempo and rhythm.

### 6.3.3 Predictor Training and Results

In order to keep the different measure estimators independent, individual predictors have been developed for each one of them. The predictors are feed with a different set of features each (see previous section 6.3.2) and, to avoid any possible influence of the existing correlation among IN, RE and TE measures (described in section 6.2), they have been trained separately. Different learning mechanisms (neural networks and SVMs) were tested during the development of the system and, finally, the selected technique was the application of regression trees [168] (standard Matlab implementation). The same technique was employed for the three different measures, each of them with their respective set of features.

The complete available content set consists on 38 different runs composed by 39 different summaries each, totaling 1482 video summaries and the corresponding assessors evaluation results. We denote each available run as $Run_i$ and each individual summary included in a given run is denoted as $sm_{i,j}$. Therefore, the complete set of 38 available runs is denoted as

$$Run_i = \{sm_{i,1}, sm_{i,2}, ..., sm_{i,39}\}\ 1 \leq i \leq 38 \tag{6.22}$$

The target of the prediction process consists in approximating the average measures for IN, RE and TE obtained by each submission, with the perspective of a future usage of the trained predictors for automatic evaluation of new summarization methods or runs.

The first step consists in the individual processing of the video summaries, aiming to train predictors for their obtained measure scores.

This process was carried out following a leave-one-out cross-validation approach, considering each complete submission, composed by 39 video summaries, as a individual validation unit. In this way, a training process is carried out with the normalized data of all the participants submissions except a complete submission kept apart for measuring the performance of the prediction system. Such approach, by completely keeping a given submission out of the training process, prevents the system from learning specific characteristics from summaries generated with the same technique as those used for validation. Given a summary, $sm_{i,j}$, we denote its original obtained measures as $IN_{i,j}$, $RE_{i,j}$ and $TE_{i,j}$ and its obtained measure predictions are denoted as $predIN_{i,j}$, $predRE_{i,j}$ and $predTE_{i,j}$. For the prediction of any of the three possible measures for a given summary, $sm_{i,j}$ belonging to run

$i$, a predictor trained with the all the available data from every summary not included in the same run, $sm_{k,l}$, $\forall k,l\ k \neq i$, is applied.

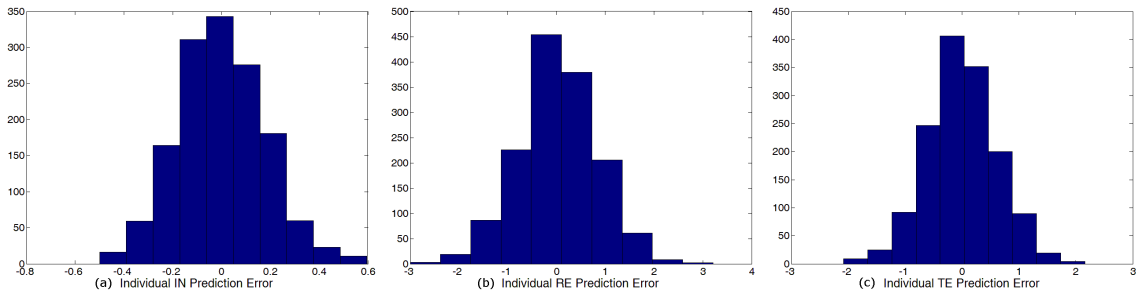


Figure 6.3: IN, RE and TE Individual Predictions Error Distributions

Figure 6.3 (a) shows the individual summaries IN measure targets and predictions error distribution ($error = IN_{i,j} - predIN_{i,j}$), obtained by applying the leave-one-out process for the 38 different available submissions. The correlation between the $IN_{i,j}$ and $predIN_{i,j}$ distributions is 0.62 (close to the 0.67 obtained in [64] with the application of manual groundtruth segment annotations). A limitation was observed in the predicted results, which are unable to reach as high or low scores as the target ones. This is caused because, from the original submissions, only one is able to reach the maximum IN scores and only one reaches the lower scores. When such approaches are left out for the leave-one-out validation, there are not available examples in the content set for so high or low IN score summaries and, therefore, the regression tree is not able to predict such values. Nevertheless, it could be expected that a tree trained with the whole set of submissions would be able to reach those values.

Although the individual predictions provide a limited utility, the target of this work, and what should be determined, is if such 'weak' predictions are useful for a global estimation of a summarization system performance. Comparing the average IN measure values for a complete run $i$, $\overline{IN_i}$, and the average predictions value for the same run, $\overline{predIN_i}$, both shown in figure 6.4, a correlation coefficient of 0.92 is obtained. The predictions present a reduced error, with the highest error found for the lowest IN values predictions, cased by the previously mentioned lack of training examples.

In figure 6.3 (b) and (c), the individual summary predictions error for RE and TE measures are shown. As in the case of the IN measure, the correlation of the individual summary predictions with respect to the assessors evaluations is limited: 0.52 in the case of RE and 0.53 for the TE. However, if considering again average values per complete submissions original measures ($\overline{RE_i}$, $\overline{TE_i}$) and predictions ($\overline{predRE_i}$, $\overline{predTE_i}$), the results are very good. Figure 6.5 shows the average prediction results per submissions for the RE measure and figure 6.6 shows the same results for the TE measure. In both cases, the obtained correlations are very high with a correlation coefficient value of 0.94.

Apart from the high correlation obtained for the three measures (IN, RE and TE), the absolute error of the predictions with respect to the real evaluations obtained by different summarization approaches is quite small and the highest errors are located in maximum or minimum values, caused by the leave-one-out experiments and the lack of examples with very high or very low values. In general terms, the obtained estimation produces very good results and apart from the comparison of new approaches with existing ones, the high correlation between the predicted per-submission IN, RE and TE values with the assessors scores will enable the usage of the developed predictors for relative comparisons between newly developed summarization approaches.

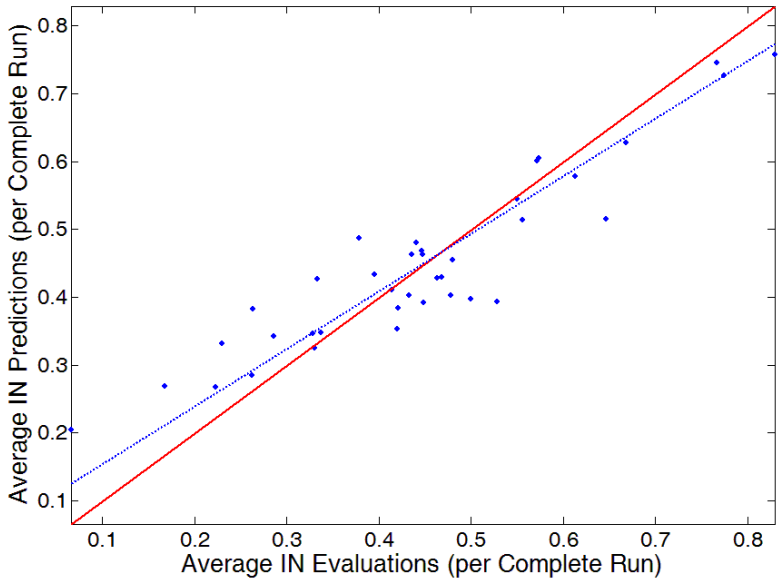


Figure 6.4: Average IN Evaluation Measures and Predictions per Run.

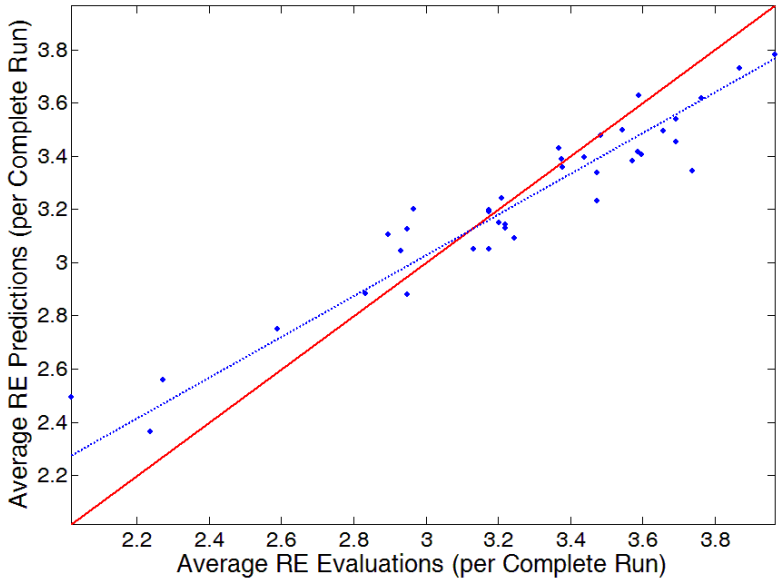


Figure 6.5: Average RE Evaluation Measures and Predictions per Run.

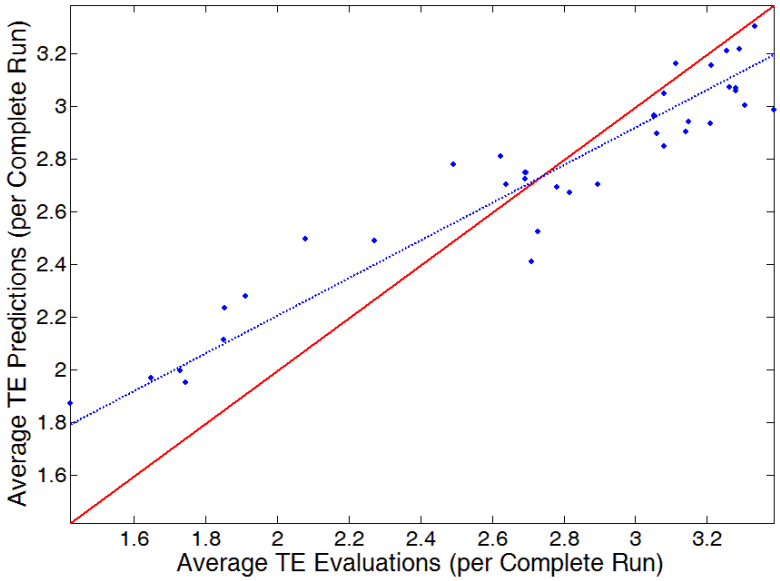


Figure 6.6: Average TE Evaluation Measures and Predictions per Run.

## 6.4 Conclusions

In this chapter, we have described the development of an automatic summary evaluation prediction system. With respect to previous works in the field, the proposed approach focuses not only in the estimation of inclusion, IN, measure scores, but in the estimation of redundancy, RE, and tempo & rhythm, TE, measures as well. Such measures are of the highest importance for the evaluation of a summarization system because, apart from the IN value, a system is defined by the relation between such IN score and the RE and TE characteristics it provides.

The results of the proposed automatic evaluation system demonstrate that, with a large scale evaluation set of data (like the TRECVid BBC Rushes Evaluation Campaigns submissions and results), it is possible to train automatic evaluation systems for the prediction of video summaries characteristics, a very difficult task given the high effort and time required for such evaluations. This system will enable the fast and easier estimation of results for newly developed abstraction approaches. Moreover, the fundamentals in which the approach relies do not make use of any specific annotation or characteristics from the specific content used (BBC rushes) and it is, in principle, applicable for different evaluations if properly trained (for example, different type of content, summarization targets, etc.).

The obtained results demonstrate that it is possible to automatically predict different subjective evaluation measures based on objective characteristics extracted from the summaries, by approaching the problem from a 'whole submission' evaluation point of view and not trying to focus on the evaluation of the individual videos. A precise prediction of assesment results for individual videos is very difficult to obtain, given the high subjectivity of the task and the high influence of the particular characteristics of each video. However, the trained predictors properly model general tendencies and, for complete submissions, the prediction errors of each individual video seem mutually balanced. The obtained results encourages the research on what those objective characteristics are and how they can be adjusted for generating higher quality video summaries.

# Chapter 7

# On-Line Video Skimming Systems Evaluation

## 7.1 Introduction

This chapter focuses in the analysis and evaluation of the two *on-line* video skimming algorithms described in chapter 5 using the automatic evaluation framework described in chapter 6. Summaries generated with both algorithms were submitted to, respectively, the 2007 [57] and 2008 [96] TRECVid BBC Rushes Summarization tasks, obtaining competitive results, specially considering that the rest of the participants submitted *off-line* approaches. However, the two TRECVid evaluation campaigns relied in slightly different evaluation measures as well as a different summary target length and, given the limit in the number of submissions, it was not possible to carry out an exhaustive study of the possibilities of the algorithms. In this chapter, the evaluation measures estimation system proposed in chapter 6 is applied for the evaluation and comparison of both systems under the same conditions. Moreover, a study of the different types of video summaries that the proposed approaches are able to generate and of how the set of configurable summarization parameters affect such summaries characteristics is presented. In this case, by making use of the TRECVid 2008 evaluation measures estimators, described in the previous chapter, which are able to 'predict' the inclusion -IN-, perception of redundancy -RE- and pleasant tempo & rhythm -TE- measures for a given summarization system, it will be possible to experiment with different summary generation possibilities and measure their impact in the summaries quality.

The following sections are organized as follows: in section 7.2 the *on-line* 'sufficient content change' skimming approach (described in chapter 5 section 5.4) is analyzed, discussing the obtained IN, RE and TE predictions for different algorithm configurations. The binary tree based approach (chapter 5 section 5.5), which provides more configuration possibilities, is analyzed in section 7.3 including a discussion about the different types and qualities of the generable summaries (sections 7.3.2 and 7.3.3). Finally, section 7.4 summarizes the presented work and conclusions.

## 7.2 'Sufficient Content Change' Approach Evaluation

In this section, the results obtained by the 'sufficient content change' approach -SCC- described in chapter 5 section 5.4 are described. In the experiments, we have taken into consideration different

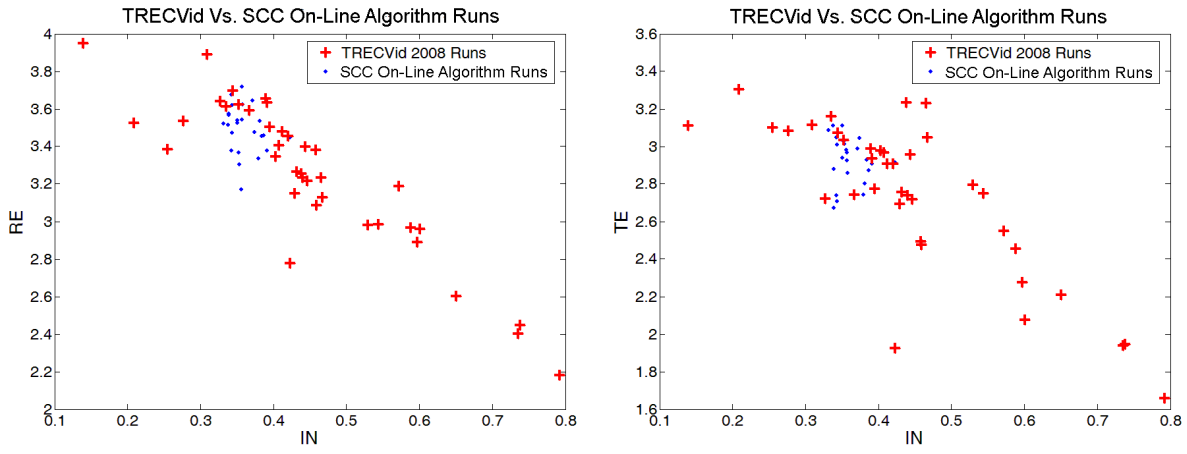


Figure 7.1: SCC On-Line Runs / TRECVid 2008 Submissions

combinations of the three possible configurable parameters of the algorithm: *splitThreshold*, *additionThreshold* and *minLength*. The *splitThreshold* was applied in the first step of the algorithm for partitioning the original video in variable size BUs according to their visual variety (see chapter 5 section 5.4.2). Higher *splitThreshold* values produce shorter video segments ( that is, BUs) and vice versa. The *additionThreshold* defines the minimum visual distance an incoming BU must keep with all the already selected ones for being included in the summary. Finally, the *minLength* parameter sets a minimum length for the processed BUs for filtering too small generated BUs.

The original algorithm was evaluated in the TRECVid 2007 BBC Rushes Summarization Task (see chapter 5, section 5.6.1) but, in this case, we will consider the evaluation measures and conditions defined in the TRECVid 2008 campaign. Several runs of the proposed algorithms were generated with all the possible combinations of the following parameters values: $splitThreshold \in \{0.1, 0.15, 0.20\}$, $additionThreshold \in \{8, 10, 12, 14\}$ and $minLength \in \{5, 25\}$. The possible combinations of parameters result on 24 complete runs which were evaluated with the IN, RE and TE measure predictors described in the chapter 6.

Figure 7.1 shows the comparison between the IN, RE and TE predicted values for the 38 original participants submissions to the TRECVid 2008 campaign and the proposed *on-line* skimming algorithm. For the generated runs the measures predictions ranged $IN \in [0.33, 0.42]$, $RE \in [3.17, 3.72]$ and $TE \in [2.67, 3.11]$, while the rest of participants' runs ranged $IN \in [0.14, 0.79]$, $RE \in [2.17, 3.94]$ and $TE \in [1.65, 3.30]$. The proposed algorithm was able to reach combinations of scores similar to several of the original *off-line* submissions but the different combinations of generation parameters did not produce great variations in the characteristics of the generated summaries (in terms of IN, RE and TE predicted values).

Figure 7.2 shows the average output rates of the *on-line* stage of the algorithm (that is, the rate between the lengths -durations- of the original video and the abstract generated by the *on-line* stage) in relation with the *addThreshold* value and considering two possible *splitThreshold* values. As the generated summary length depends of the characteristics of the original video, in case of exceeding the 2% target limit, the *off-line* stage is executed for the reduction of the final summary length (see chapter 5 section 5.6.1).

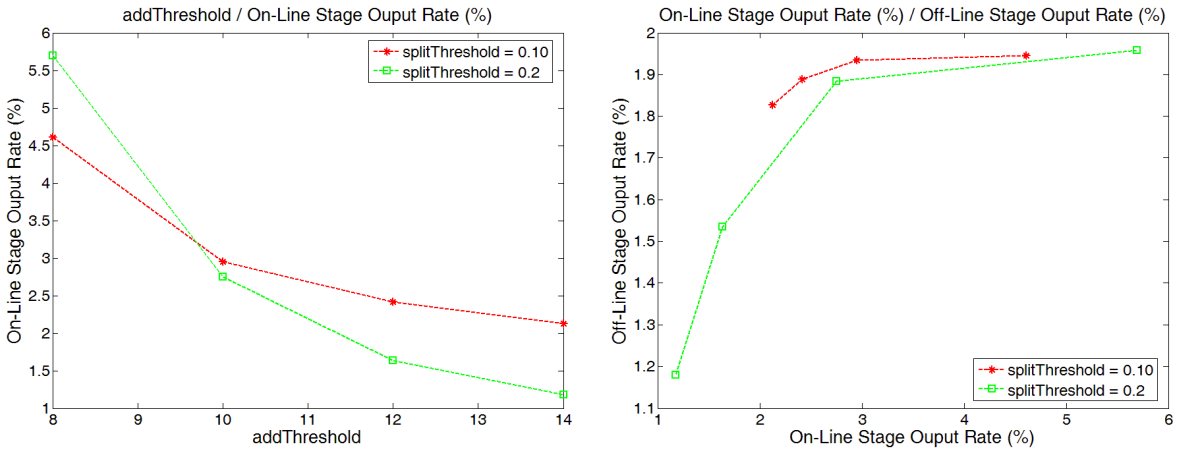


Figure 7.2: On-Line Stage Output Rates

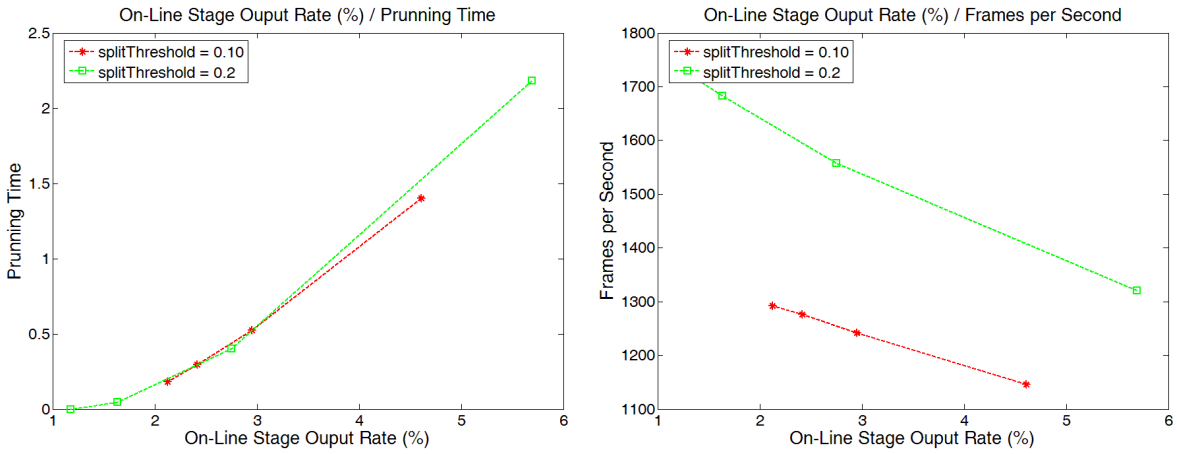


Figure 7.3: Off-Line Stage Pruning Time / Frames per Second Processing

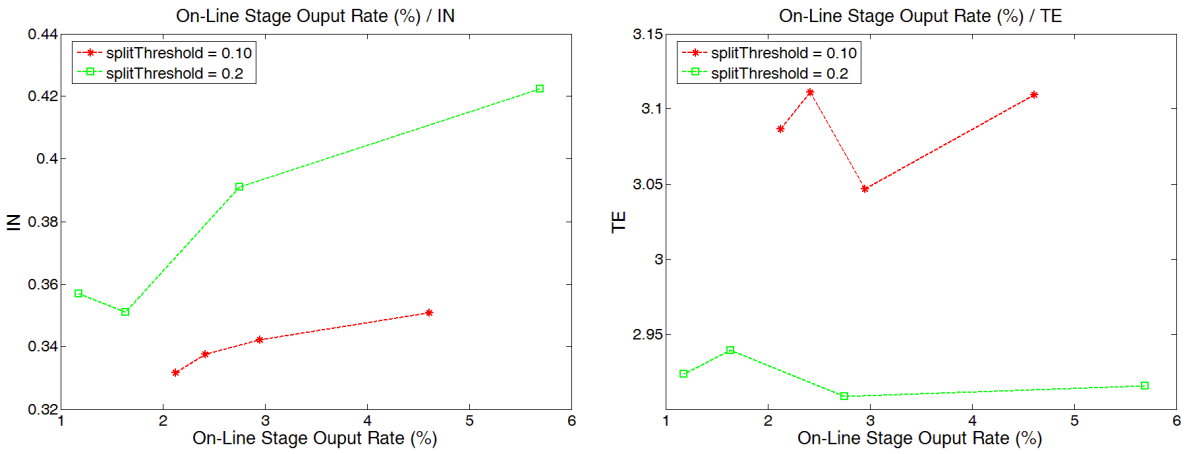


Figure 7.4: On-Line Stage Output Rate / IN and TE Results

Figure 7.3 shows the required time for the *off-line* processing stage[1] (named 'Pruning Time') as well as the average frames per second processed by the algorithm in relation with the *on-line* stage output rate. Although the *off-line* stage times are kept in very small values, the amount of processing grows quadratically with respect to the amount of data. With respect to the effects of the *off-line* stage in the summary characteristics, figure 7.4 shows a comparison between the *on-line* stage output rate and the predicted IN values which shows an increment in the summary IN value when a higher amount of data is processed by the *off-line* stage (it must be noted that in all cases the final summary length never exceeds 2% the original video length). On the other hand, such processing does not seem to produce a reduction in the RE or TE measures and results may invite to think that the *off-line* stage results can offer better quality than those obtained only by the *on-line* stage. Such quality increment seems reasonable because the *off-line* stage reduces the *on-line* generated abstract length with the complete abstract information available, while the *on-line* stage is forced to take instantaneous threshold-based decisions about the inclusion or discard of BUs (the binary tree approach, analyzed in the next section 7.3, provides mechanisms for a more precise BU selection based on the accumulation of several BUs before their selection or discard).

Observing the results shown in figure 7.4, its is possible to extract some interesting conclusions about the effects of the *splitThreshold* value in the output summary characteristics. As we have previously explained, the *splitThreshold* controls the length of the BUs processed by the system: a higher *splitThreshold* generates smaller BUs producing an increment in the IN value due to the higher amount of different BUs which can be selected. On the other hand, smaller BUs result on a reduction in the output summary TE value, probably caused by the increment in the number and frequency of abrupt changes in the generated summary.

The results shown in this section demonstrate the possibilities provided by the proposed SCC algorithm for the *on-line* generation of video summaries with quality levels comparable to *off-line* approaches. However, for the generation of limited length summaries, the proposed approach requires a priori knowledge about the characteristics of the original content for setting an appropriate *addThreshold* value that could allow to avoid the application of the *off-line* summary length reduction stage. Moreover, the characteristics of the generated summaries are fixed in a small range of possible

[1] Hardware platform: Intel Xeon @2.83GHz with 24GB of RAM.

IN, RE and TE values and, as it is shown in the carried out experiments, the variations on the generation parameters do not have heavy effects in the characteristics of the generated video abstracts. In the next section, the characteristics of the binary-tree based *on-line* summarization approach, developed for the dealing with the drawbacks of the SCC algorithm, providing additional features and configuration capabilities, is in-depth analyzed.

## 7.3 Binary Tree Approach Evaluation

The binary-tree -BT- based summarization approach provides many different summarization possibilities by means of a higher number of configurable parameters. A first set of parameters is related to the operational characteristics of the system: summarization tree depth, number of branches, BU (basic unit) length and application of acceleration to the original video (all of them explained in chapter 5 section 5.5). The second set of parameters will determine the characteristics of the generated summaries by applying different scoring weights in the tree generation and branch selection processes. Such weights control different characteristics of the video summaries such as summary length, visual redundancy, selection of contiguous BUs from the original video or the selection of high visual variation BUs. The combination of the different weighting and operational parameters will determine the type of summary that will be generated and its quality level. The analysis carried out in this section shows the relationships between the generation parameters and characteristics of the produced summaries.

### 7.3.1 Overall Performance

This section is focused on analyzing how the proposed *on-line* summarization approach compares with the results obtained by the *off-line* summarization approaches submitted to the TRECVid 2008 BBC Rushes Summarization task. Figure 7.5 shows the comparison between the IN, RE and TE predictions for the 38 original runs submitted to the TRECVid summarization task and 367 generated *on-line* runs. The *on-line* runs were generated with different configurations of parameters (some of them will be later analyzed) trying to determine the different ranges of IN, RE and TE values that could be covered by the BT *on-line* approach and how such measures values were combined. For the summaries generation, BUs from 1 to 50 frames, trees with 5 to 5000 nodes and 6 to 300 levels depth were applied together with different combination of summary scoring weights. It should be noted that not every possible parameter configuration was tested, carrying out a combination of random parameters runs, manually selected ones and several grid parameters tests.

Results shown in Figure 7.5 show how the *on-line* approach is able to reach measure predictions covering almost any measure values obtained by the *off-line* approaches submitted to the TRECVid 2008 campaign. For the BT generated runs, the measures predictions ranged $IN \in [0.18, 0.72]$, $RE \in [2.45, 3.72]$ and $TE \in [1.65, 3.24]$, while the *off-line* runs were able to reach $IN \in [0.14, 0.79]$, $RE \in [2.17, 3.94]$ and $TE \in [1.65, 3.30]$ values. The maximum and minimum IN, RE and TE measures values obtained with the *on-line* approach were slightly under the original submissions limits. However, the generated runs aimed to determine the measure covering capabilities of the *on-line* algorithm without focusing on the maximization or minimization of each possible measure. Moreover, the predicted measures must be considered as indicators of a summary quality and not as exact scores. The obtained results demonstrate the capability of the proposed system for generating different types of summaries.

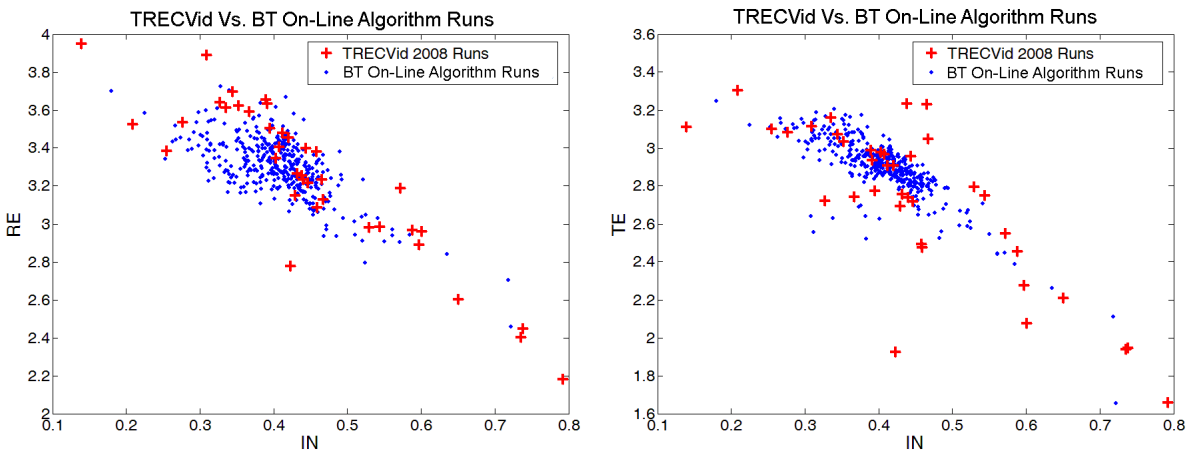


Figure 7.5: BT On-Line Runs - TRECVid 2008 submissions comparison

Another issue which must be taken into account is how the obtained IN, RE and TE results combine in a given summarization system. It can be considered that, given the inverse correlation between the IN, RE and TE measures, those summarization approaches able to keep simultaneously high scores in the IN, RE and TE values will be of better quality than others which, for example, could get high scores in one of the measures but are not able to keep high results on the others. In the results shown in Figure 7.5, it is possible to check how for both IN/RE and IN/TE relations the *on-line* approach is able to perform very well, reaching most of the measure combinations corresponding to *off-line* approaches.

It can be observed how, in the set of obtained results, only a few runs are able to perform above the 0.5 IN value limit. Such high IN values were only obtained when small length BUs (below 25 frames) were applied during the summarization process. In other words, such high IN scores were only reachable when splitting the original video in very small fragments. Figure 7.6 depicts the relationship found between a group of runs with a BU length ranging from 1 to 40 frames. The plots show two differentiated runs: a first group with BU lengths below 25 frames (commonly considered as the minimum length for the perception of a video fragment) and a second group with values above 25 frames. Both runs were configured to behave as a subsampling approach by only considering the size weighting, *wSize,* in the summarization system (see scoring details in chapter 5 section 5.5.3) while keeping the rest of the weights in null values. It can be observed how smaller BU lengths allow to compose a summary including frames from a higher number of different positions in the original video and, therefore, it is possible to obtain a higher event inclusion rate, that is, IN scores. Nevertheless, such small BU length produces negative effects in both RE and TE scores because the frame repetition perception is increased (specially if dealing as a subsampling approach) and the rhythm and tempo of the summaries are not adequate for a comfortable watching. Figure 7.7 shows the effects on the IN, RE, and TE predicted scores caused by the change of the BU length on the depicted examples.

Another feature included in the proposed *on-line* summarization approach is the adaptive segment acceleration mechanism, described in chapter 5 section 5.6.2, in charge of dropping consecutive frames in case they are too similar (by applying a configurable similarity threshold). The maximum applied acceleration is set to 2x in order to avoid an excessive, and too unpleasant, acceleration in the summaries that could prevent the users from perceiving included events. In the TRECVid 2008

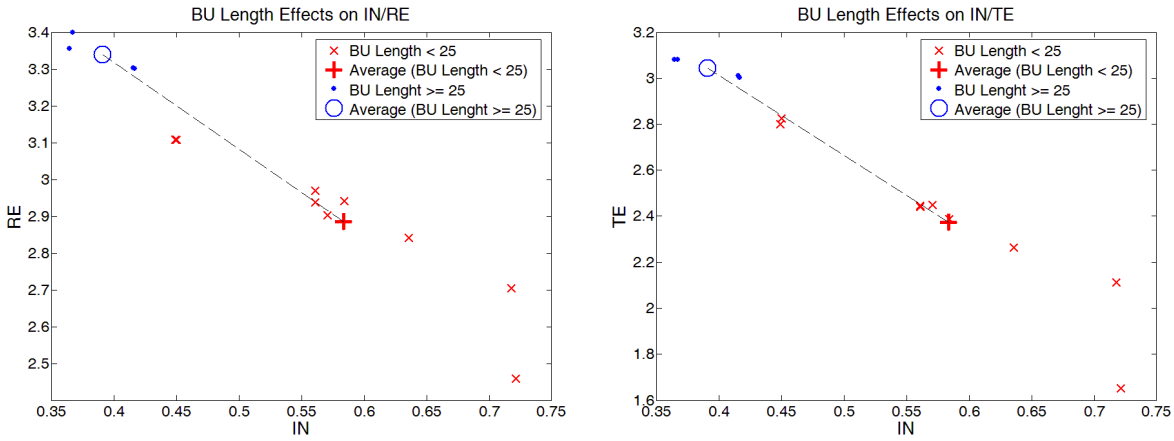


Figure 7.6: BU Length - IN/RE/TE Comparison

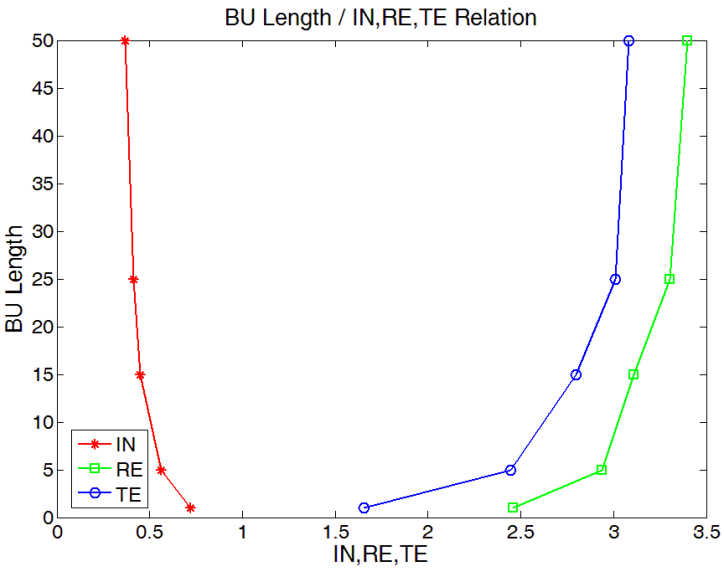


Figure 7.7: BU Length Effects on IN/RE/TE Scores

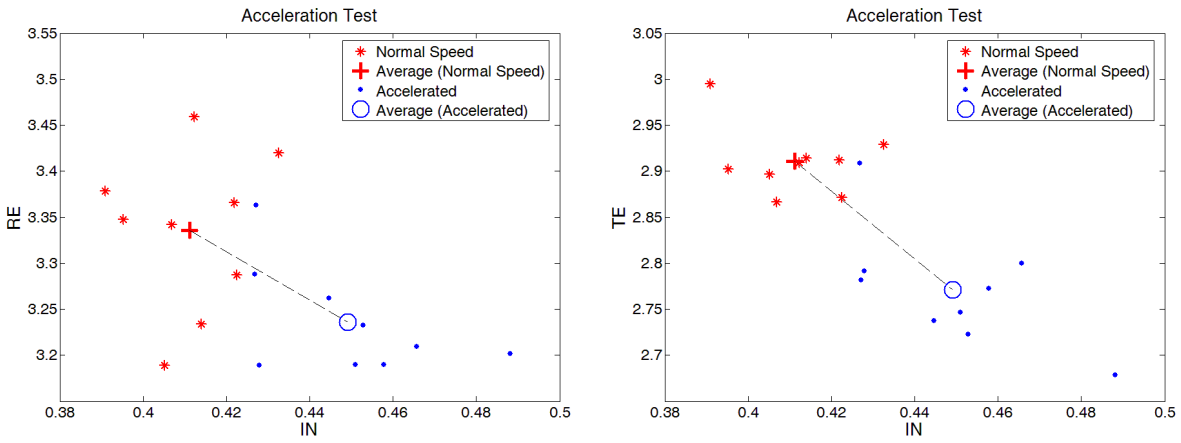


Figure 7.8: Acceleration Effects on IN/RE/TE Scores

evaluation campaign, it was observed how systems which implemented acceleration mechanisms obtained higher IN rates, but also smaller RE and TE results. Figure 7.8 shows measure predictions for two *on-line* runs with identical generation parameters except for the acceleration, which was applied only in one of the approaches. The plots show a behavior analogous to the TRECVid evaluation, where those runs where acceleration was applied obtained a better IN result but reduced TE and RE predicted scores. The explanation for such effect is straightforward: accelerated video permits to store more information in the same output summary length but also a higher amount of potentially included repeated content and, of course, the tempo and rhythm of the summary is reduced with respect to a summary played at normal speed.

### 7.3.2 Control of Summary Type

After describing how the *on-line* summarization system is able to cover the complete range of IN, RE and TE predicted scores, this section focuses on analyzing how the proposed summarization approach allows to control the balance between such measures by making use of the defined weights applied for the branch scoring and selection mechanisms. As it has been previously described, apart from the depth of the summarization tree, maximum allowed branches, BU length and whether to apply acceleration or not, the characteristics of the generated summaries are mainly guided by the scoring weights (a complete description about the scoring and weighting mechanisms can be found in chapter 5, section 5.5.3):

- *wSize:* Defining the weight of the generated summary size score, *scSize*, in the final score.
- *wRedundancy:* Defining the weight in the final score of the redundancy score, *scRedundancy*, which measures the amount of repeated content in a generated summary.
- *wContinuity:* Determining the influence in the final score of the continuity score, *scContinuity*, which measures the amount of consecutive BUs from the original video selected as part of the video summary.
- *wVariation:* The variation score, *scVariation*, measures the amount of internal variation within a given BU.

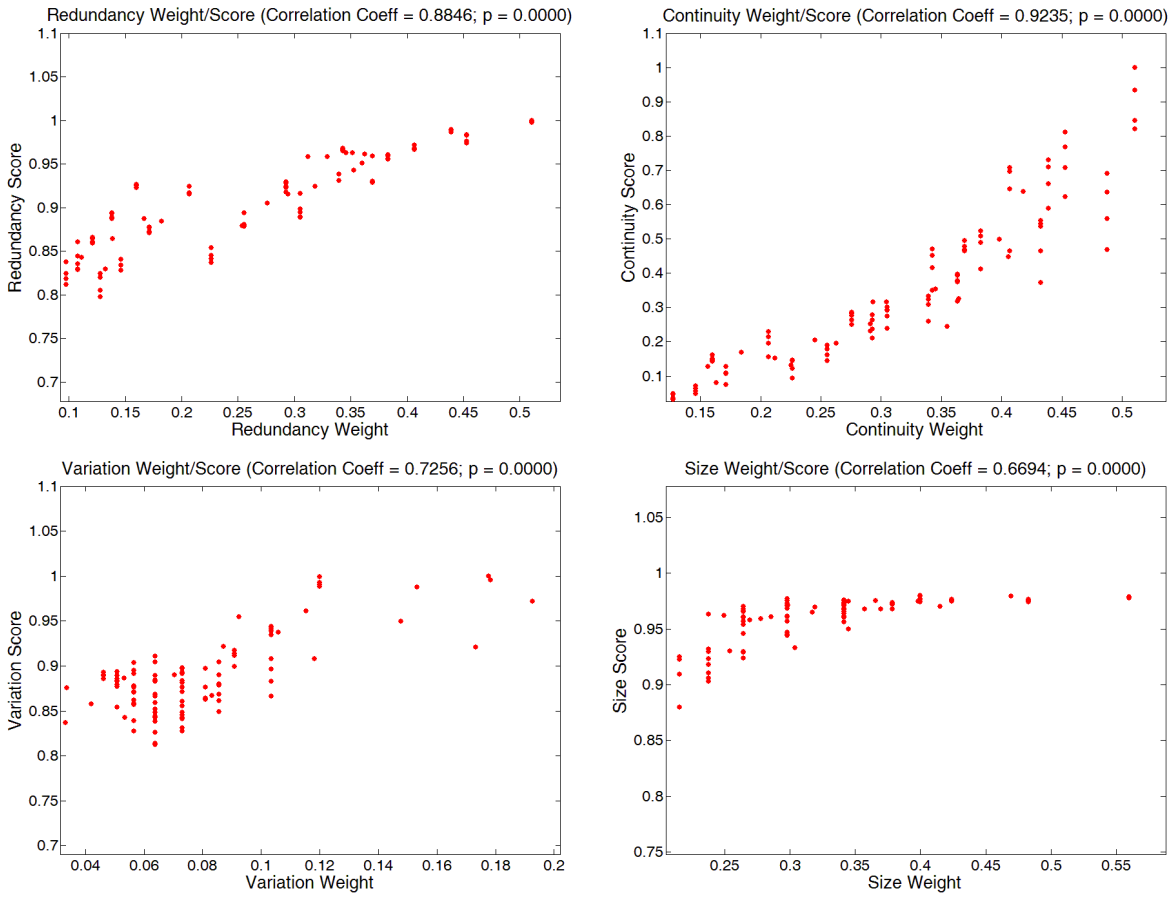


Figure 7.9: Score Weighting and Results

It will be always considered that the weighting values are normalized, that is, the sum of all the weights is equal to 1 ($wSize + wRedundancy + wContinuity + wVariation = 1$) .

In the first place, it should be determined how well the defined weights (*wSize*, *wRedundancy*, *wContinuity* and *wVariation)* can control the actual scores (*scSize*, *scRedundancy*, *scContinuity* and *scVariation*) obtained by the generated summaries. For such purpose, 100 *BT on-line* runs were generated with a fixed BU length of 20 frames without the application of video acceleration. The rest of the parameters were randomly generated, keeping a maximum tree depth of 200 levels and number of branches of 1500.

Figure 7.9 shows how the normalized scores of the generated summaries vary with respect to their corresponding weighting factors. The correlation between weights and obtained scores is clear: 0.88 for *wRedundancy* and *scRedundancy*, 0.92 for *wContinuity* and *scContinuity*, 0.72 in the case of *wVariation* and *scVariation* and, finally, 0.67 for the size weight, *wSize*, and score, *scSize*. The results demonstrate how the summarization tree approach carries out a proper branch selection process which produces an increment in the summary characteristics with higher score weights.

The last step for determining how the proposed summarization approach can be configured for obtaining different types of summaries, requires to analyze how the summaries scores, controlled by the defined scoring weights, affect the IN, RE and TE measures. Table 7.1 summarizes the correlations

| Scores / Measures | IN | RE | TE |
|---|---|---|---|
| *scRedundancy* | 0.79 (p=0) | -0.41 (p=0) | -0.90 (p=0) |
| *scContinuity* | -0.94 (p=0) | 0.40 (p=0) | 0.89 (p=0) |
| *scVariation* | 0.25 (p=0.012) | 0.02 (p=0.81) | -0.22 (p=0.02) |

Table 7.1: Summary Scores and Predicted Measures Correlations

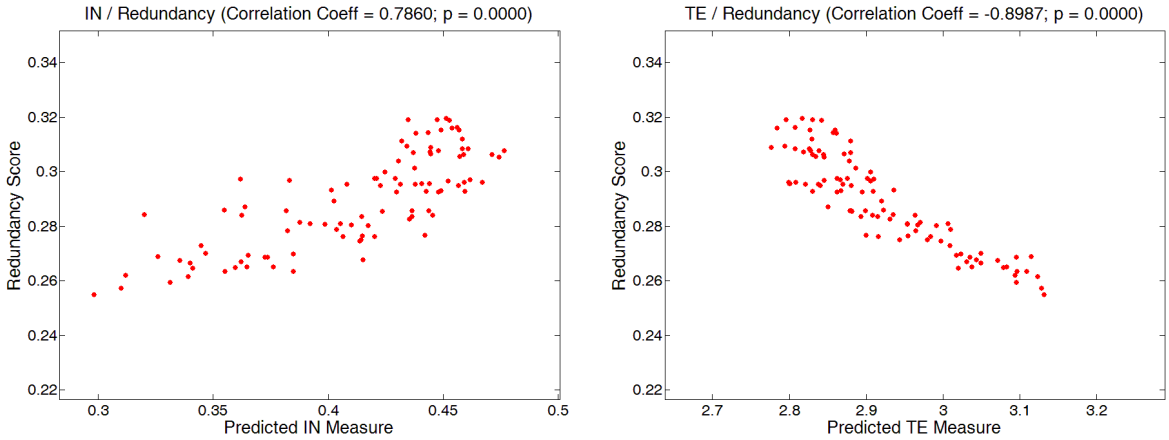


Figure 7.10: Redundancy Score and IN/TE Measures Relation

found between the score weights (*scRedundancy*, *scContinuity* and *scVariation)* and the predicted summaries measures (IN, RE and TE) together with their associated p-values (probability of getting a correlation as large as the observed by random chance).

As expected, it has been found that the redundancy score, *scRedundancy*, is heavily related to the IN measure (0.79 correlation) in a direct way, and to the TE measure (-0.9 correlation) in an inverse relation. On the other hand, the obtained continuity score, *scContinuity*, presents the opposite behavior with negative correlation with the IN measure (-0.94 correlation) and positive with the summaries TE measure (0.89 correlation). Figures 7.10 and 7.11 show graphically the relation between the IN and TE measures and the *scRedundancy* and *scContinuity* scores.

As it can be noticed in Table 7.1, the relations of the RE measure are not as straightforward as those for the IN and TE measures (results are graphically shown in Figure 7.12). The RE scores are slightly correlated to *scContinuity* values (0.40 correlation). This fact can be explained because a summary with a high continuity rate (contains more fragments located consecutively in the original video) is more unlikely to contain repeated content in separated positions, condition that produces a strong redundancy perception. The same reasons can be applied for explaining the RE measure and *scRedundancy* score correlation (-0,41). Although a direct correlation could be expected between such measures, high *scCorrelation* scores imply higher differences between all the BUs composing a summary. Such conditions imply that, in many cases, consecutive BUs are not included in the summary because being too similar; and this fact produces negative effects on the RE and IN measures.

According to Table 7.1, the effects of the *scVariation* score are not as strong as the *scRedundancy* or *scContinuity* ones. The selection of high variation BUs in the video summary implies a slight increment of the IN measure (0.25 correlation) while, on the other hand, produces a reduction of the

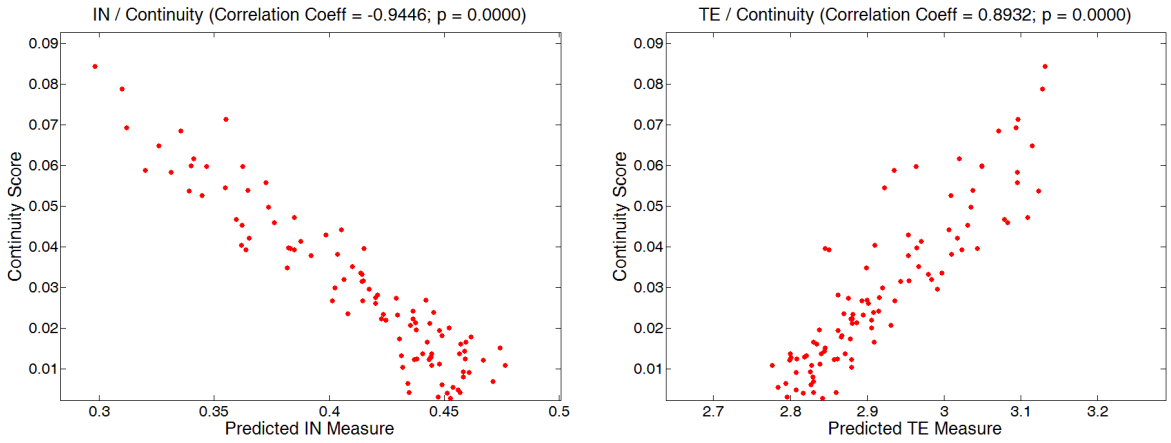


Figure 7.11: Continuity Score and IN/TE Measures Relation

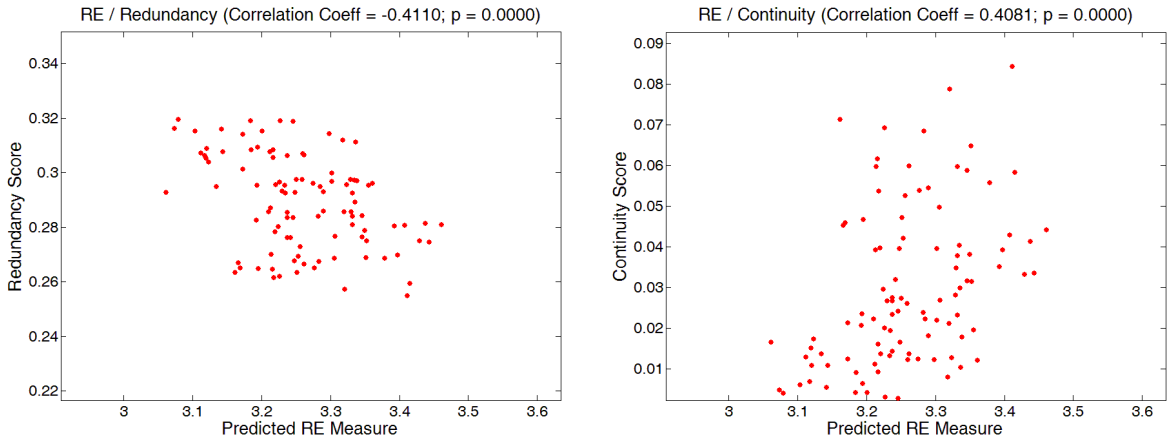


Figure 7.12: Redundancy Measure and Scores Relation

| Generation Parameters | | | | | Predicted Scores | | |
|---|---|---|---|---|---|---|---|
| *wSize* | *wRedundancy* | *wContinuity* | *wVariation* | BU Length | IN | RE | TE |
| 1.0 | 0.0 | 0.0 | 0.0 | 1 | 0.72 | 2.45 | 1.65 |
| 1.0 | 0.0 | 0.0 | 0.0 | 5 | 0.56 | 2.96 | 2.44 |
| 1.0 | 0.0 | 0.0 | 0.0 | 15 | 0.45 | 3.11 | 2.82 |
| 1.0 | 0.0 | 0.0 | 0.0 | 25 | 0.41 | 3.30 | 3.00 |
| 0.3 | 0.4 | 0.025 | 0.0 | 25 | 0.38 | 3.47 | 2.96 |
| 0.5 | 0.3 | 0.10 | 0.0 | 25 | 0.33 | 3.42 | 3.05 |
| 0.2 | 0.4 | 0.15 | 0.0 | 25 | 0.29 | 3.53 | 3.15 |
| 0.4 | 0.0 | 0.25 | 0.0 | 25 | 0.18 | 3.69 | 3.24 |

Table 7.2: Examples of Summarization Parameters and Obtained Scores

TE measure (-0.22 correlation). The results are coherent with what was expected: high variation BUs include more information and, therefore, the IN measure is favored. Nevertheless, high activity fragments produce a reduction in the summary TE values, probably because too many, and maybe short, high activity consecutive fragments may produce unpleasant perception in the viewer.

Table 7.2 shows several examples of different combinations of generation parameters together with the predicted scores for the corresponding generated video summaries. It can be observed how the variation in the scoring weights as well as the BU length (no acceleration nor *wVariation* weight are applied) produces video skims with different characteristics, that can be applicable for different user preferences or scenarios.

### 7.3.3 Control of Summarization Quality

Apart from the possibility of covering the complete range of IN, RE and TE measures provided by the proposed *on-line* summarization approach, the quality of the summaries should be taken into consideration. As it has been previously discussed, a summary can be considered 'better' than other as long as the combined value of its IN, RE and TE measures improves the second summary results. For example, given two summaries with equal IN and RE values, it seems reasonable to determine that the summary with higher TE measure will be the best one. Of course, depending of the summary application scenario, it could be possible to establish different priorities for the IN, RE and TE measures. For example, in the case of a user browsing Internet videos in a portal like You Tube, summaries with a balance between IN, RE and TE could be appropriate. On the other hand, in an application scenario such as the checking of surveillance recordings, the IN measure could be prioritized, as the inclusion of all the relevant events in the summary would be probably more relevant than the redundancy or rhythm of the summary.

In this section, we will analyze the scalability capabilities of the proposed BT *on-line* summarization approach. The generation of video summaries by the application of summarization trees permits to control the performance of the process, in terms of generation delay and processing speed, by controlling the maximum tree depth and number of branches. For determining the quality of the generated summaries, we define two possible scores: *internalScore* and *qualityScore*.

The *internalScore* is based in the scores obtained from the summarization process which were applied in the branch selection process. In the general case of $i$ possible scores with their corresponding weights, the *internalScore* value is defined as:

$$internalScore = \frac{\sum w_i \cdot score_i}{w_i} \tag{7.1}$$

were $w_i$ and $score_i$ correspond, respectively, to the *ith* weight and score. In the present case, only four previously defined scores (*scRedundancy*, *scContinuity*, *scVariation* and *scSize*) and corresponding weights are considered, and the *internalScore* value represents how adequately the summarization system was able to generate a summary prioritizing the different scores according to the defined weights. As the applied scores are represented in different scales, all the values in the following experiments are normalized prior to the calculation of the *internalScore* value.

The *qualityScore* will determine the 'subjective' quality of the generated video summaries by making use of the IN, RE and TE predicted measures. Although it could be discussed how to combine the three measures for generating a single quality measure, in this case, the *qualityScore* will be defined by the average of IN, RE and TE normalized values.

In this case, 20 summarization runs were used for carrying out the experiments, keeping fixed the score weights, not using video acceleration, selecting BUs of 30 frames, and using summarization tree depth values ranging from 10 to 100 levels combined with a maximum amount of 100 to 10000 nodes. For the experiments carried out, the individual internal scores and predicted measures are normalized taking into consideration only the values obtained with the 20 summaries of the test set.

Figure 7.13, shows the effect of the tree depth variation on the *internalScore* and *qualityScore* values. In the case of the *internalScore,* it can be clearly observed how the increase in the depth of the summarization tree increases the "quality" of the summary for the different number of nodes runs except when reaching 100 levels depth. Such score reduction is caused because very deep trees imply a huge amount of possible branches and, as in this case the number of branches is limited, too sparse trees are generated. The experiments carried out show that this situation produces a negative effect in the summary quality unless a high enough number of nodes is applied. Graphs for the *qualityScore* show results with higher variations. Such variations may be caused for several reasons, probably associated to the error margin that the IN, RE TE prediction system implies. On the other hand, the correlation between the *internalScore* and *qualityScore* measures is not perfect and, of course, increasing *internalScore* measures do not imply the same behavior in the *qualityScore.* However, observing the *qualityScore* average values, it can be noticed how the evolution of such values is quite similar to the *qualityScore* ones, with a constant increment of the scores until the maximum depth level is reached, and the summary quality decreases.

Figure 7.14 shows the runs information depicting, in this case, the evolution of the summaries quality with respect to the number of nodes in the tree. The *internalScore* values clearly show a constant increment as the number of nodes in the summarization tree increases. It should be noted how the curve corresponding to the 100 levels depth tree maintains quality levels below the tree with 70 levels depth. The 10 levels tree curve stalls with values above 1000 nodes, effect caused because a 10-levels tree may have a maximum of $2^{10} = 1024$ nodes and, therefore, all executions with a higher number of nodes produce the same result. Once more, the *qualityScore* graphs show a much higher variation and unclear results. However, the average curve shows a slight increment as the number of applied nodes grow.

The obtained results demonstrate how the implemented summarization algorithm provides a functional scalability mechanism for the generation of higher quality summaries when increasing the depth and number of nodes in the summarization tree, showing a clear increment of the *internalScore* values when the tree depth and number of nodes increase, until certain depth values are

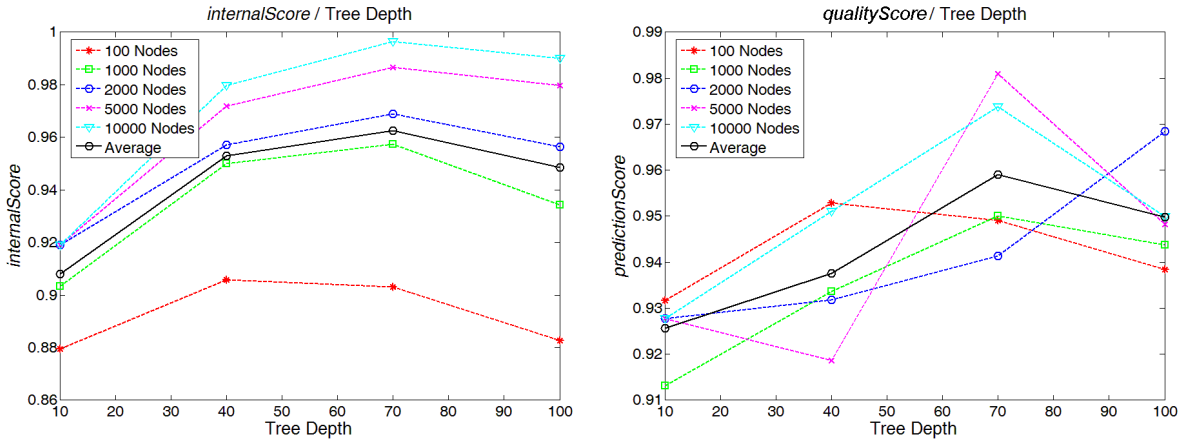


Figure 7.13: Effects of the Summarization Tree Depth on Summary Quality

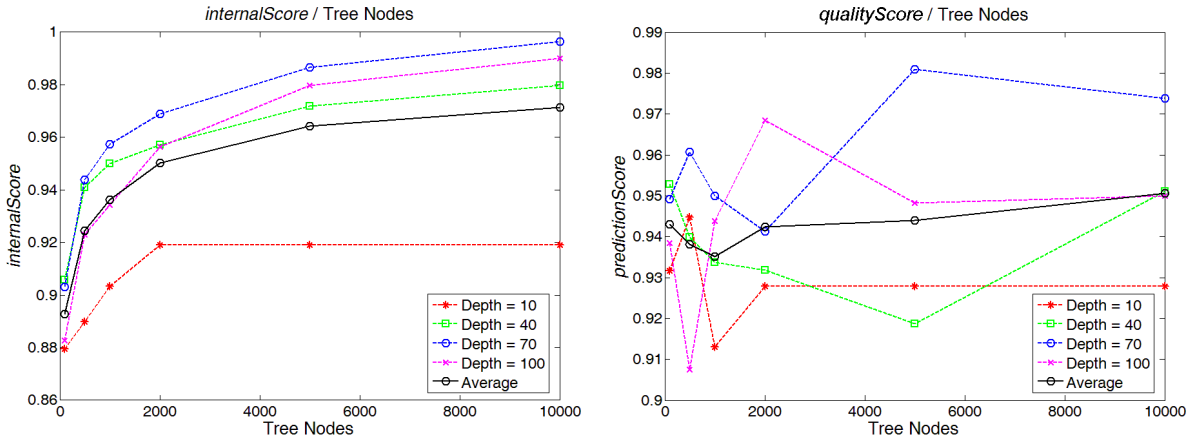


Figure 7.14: Summarization Tree Node Limit Effects on Summary Quality

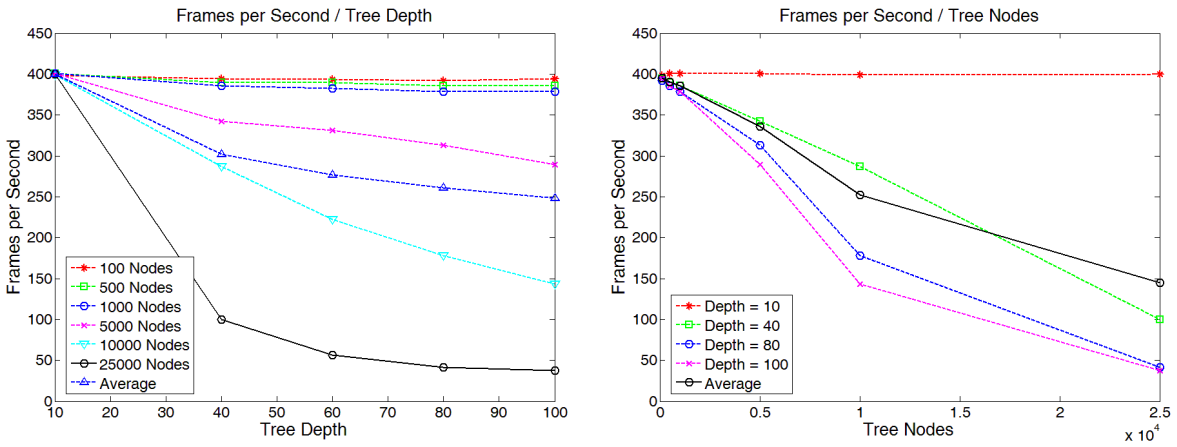


Figure 7.15: Summarization Tree Nodes and Depth Effects on Processing Rate

reached; this show that a large number of nodes is required to obtain high quality results. The results obtained with the predicted IN, RE and TE measures, with a similar evolution to the internal scores results, allows to hypothesize that the *internalScore* increment implies higher *qualityScore* values. In any case, the proposed summarization approach allows the implementation of any scoring measures and, therefore, the application of new branch scoring mechanisms for the generation of a different type or higher quality summaries is possible.

Of course, the increment in the tree complexity (in terms of number and depth of the generated branches) implies an increment in the algorithm computation complexity: more processing time and a higher delay is required for the generation of a video summary. Figure 7.15 shows the evolution of the average number frames per second the system is able to process for the different combinations of tree depth and number of nodes. All the executed runs are able to perform at least at 25 fps[2] , maintaining the *on-line* requirements of the system.

## 7.4 Conclusions

In this section, a complete evaluation of the *on-line* approaches described in chapter 5 has been carried out making use of the automatic evaluation framework described in chapter 6. Different aspects related to the characteristics and quality of the generated summaries, as well as operative aspects of the summarization algorithms, have been considered.

With respect to the first proposed algorithm, a 'sufficient content change' approach, it has been shown how the quality of the generated summaries is comparable to several *off-line* approaches. Nevertheless, in case of dealing with constrained length summaries, the approach requires an *off-line* processing stage when the *on-line* generated summary exceeds the length limit. Additionally, the variability in the characteristics of the summaries that can be generated with such approach are very limited.

In the case of the second algorithm, based on the generation of binary trees, the results show how the proposed system is able to generate video summaries with different types of characteristics, that

[2]Hardware platform: Intel Xeon @2.83GHz with 24GB of RAM.

is, with different combinations of IN, RE and TE predicted measures. The generated summaries cover almost the complete range of IN, RE and TE values, and their possible combinations, obtained by the *off-line* systems presented to the TRECVid 2008 BBC rushes evaluation campaign.

The different generation parameters for controlling the acceleration of the video and BUs length have been analyzed, describing the effects of such parameters in the characteristics (IN, RE and TE measures) of the generated summaries. Moreover, it has been demonstrated how the configurable scoring weights guide the branch selection mechanism allowing to generate summaries with different combinations of IN, RE and TE measures.

Finally, the scalability properties of the binary tree based summarization system were analyzed, demonstrating how the increment in the number of summarization tree levels and nodes yields, as expected, to the generation of higher quality summaries, as well as to an increase in the required processing time and generation delay. Such scalability properties will allow the configuration of the summarization system for different combinations of speed/delay and output quality according to specific application scenarios. Such possibility, together with the mechanisms for generating different kind of summaries, constitutes a highly customizable *on-line* summarization approach with results comparable to *off-line* approaches.

# Part V

# Applications

# Chapter 8

# On-line Video Abstract Generation of Multimedia News

## 8.1 Introduction

The work presented in this chapter focuses on the *on-line* generation of news bulletins abstracts. The process consists of the combination of news stories segmentation, video skimming and composition techniques operating as the original video is being broadcasted. The result of the process is a video abstract in which, for each story found in the news bulletin, a visual composition is generated combining the anchorperson introduction and a video skim of the visual segments of the story. The proposed system is able to generate the video abstract, not just in an efficient but also in a progressive way: at any given time instant in the news bulletin, the proposed method provides a partial abstract of the already received news stories, without the need of the complete bulletin content. The system is able to work with news bulletins composed by an arbitrary number of stories and without assumptions about their length or shot composition. The *on-line* generation capability provided by the module is specially interesting for this kind of content because the fast availability of the news information is of highest relevance for both professional users, such as journalists, or regular users interested in access to the latest news. In order to fulfill the requirements of this *on-line* generation operation mode, a number of techniques for shot classification and video skimming, problems traditionally solved with *off-line* techniques, have been implemented in an *on-line* way.

The chapter is structured as follows: after this introduction, section 8.2 presents the state of the art on existing news content abstraction approaches. Section 8.3 gives an overview of the proposed abstraction system which combines different stages for the incoming content classification, video skimming and abstract composition. Section 8.4 presents an overview of the characteristics of news video bulletins (the developed techniques for *on-line* news segments classification are further detailed in appendix B). Section 8.5 details the orchestration of the different modules composing the system. The obtained results, in terms of objective and subjective evaluations, are summarized in appendix C. Finally, in section 8.6, future work is foreseen and conclusions are drawn.

## 8.2 Related Work

News content has been a popular subject of research interest in the last years, particularly, the techniques for its analysis, understanding and abstraction. This is probably due to the huge amount of available news broadcast video and the derived necessity of an easier access.

When considering the application of abstraction techniques for news content, one of the main problems to deal with is the identification of story boundaries. In the TRECVid 2003 story segmentation task, participants employed a wide variety of effective techniques, including text-based (the original videos were provided with closed caption text) and audiovisual approaches. In [169] several of the presented techniques are compared. The results show that the best results were obtained when applying audiovisual or a combination of audiovisual+text (up to 0.77 $F_1$ scores) techniques, while the text-only based approaches obtained worse results. Some of the participants [170, 171] obtained up to 80% accuracy in the detection of anchorperson and in [169] it is stated that just with a correct anchorperson detection rate close to 100% it would be possible to achieve a $F_1$ measure of 0.62 in story segmentation. The correct anchorperson detection is, therefore, of the utmost relevance for news abstraction. Face detection techniques have been a commonly applied for such purpose: for example, in [172] a list of major casts (including anchorperson in news content) is generated by a clustering of the content based on face detection and audio features. In [11] a system for news content browsing making use of the same face detection technique [173] as this work is presented. [174] includes, as part of the extracted feature set for story segmentation, a face detection algorithm based on flesh color detection followed by a shape analysis. Such work assumes that each story begins with an anchorperson followed by a more detailed report. The video bulletin is divided into shots clustered using shot length, distribution, motion activity and face detection features. Authors found that anchorperson shots tend to be clustered together due to their high similarity and make use of a SVM for its classification. In [12] it is proposed to make use of compressed-domain extracted features (motion activity and DC-images) for the detection of the anchorperson based on color comparison in high motion areas of the image. In this case, the anchorperson audio is kept and a summary is generated by its combination with a summary of the following news report segment, constrained to a length equal to the kept audio length. In [13] a system for the selection of news highlights based on the analysis of closed-captions and its alignment with news bulletin audio is depicted. [38] deals with the presentation aspects of video search in the news domain proposing, in this case, video collages as the tool for fast browsing. Work described in [175] proposes the division of news bulletins in anchorperson and news shots. Anchorperson shots are identified by calculating the difference between consecutive frames and comparing those with small differences (anchorperson shots are almost static) with a quite simple anchorperson model which defines certain areas, like head or body, where motion should be found. Another *off-line* clustering-based approach can be found in [176], where face detection is performed including the consideration of cloth color under the head. Shots with faces are clustered based on this information and the largest cluster is assumed to correspond to the anchorperson (it should be noted that this approach could fail in cases where, like in the content set we used, there is more than one anchorperson during the news bulletin). Weather report shots are detected as well by making use of color histograms (blue and green predominance can usually be found) and motion vector information. In cases where an anchorperson appears among two reports corresponding to the same story, a merging process is carried out based on textual information analysis and visual comparison allowing the fusion of segmented stories sharing the same topic. The usage of anchorperson cloth color can also be found in [177] where faces are detected based on flesh-color analysis

in images. Other systems consider a high number of possible shot categories: in [178] a decision tree, based on low level (color histogram, motion activity, shot duration, etc.) and high level (face detection and text captions) features, is applied for differentiating between 13 possible shot categories. A further HMM analysis is then applied to locate scene boundaries. A completely different approach can be found in [179] where stories segmentation heavily relies on closed-captions, speech alignment and commercial detection (the latter based on shot change rate and black frames detection).

In summary, the studied systems for video abstraction and their specific application for news content include a high variety of extracted features and applied techniques, many of them focused on the detection of anchorperson shots. For this purpose, face detection, color and shape analysis algorithms are commonly applied. Nevertheless, although several of the existing techniques are quite efficient, none of the existing approaches seem to work as an *on-line* system, providing instant abstract availability in any moment during the broadcast or abstraction process. Most techniques assume the complete availability of the original content and unlimited time for the generation of the news bulletin abstracts. Even those systems which provide real-time browsing capabilities or a high efficiency system rely on content analysis carried out with the availability of the complete original content and should be, therefore, considered *off-line* approaches. When studying existing generic video abstraction systems not necessarily focused on news content, it is possible to find several progressive generation systems. Nonetheless the complexity of the existing techniques and the type of generated abstract are limited. In this chapter we describe a complete system able to carry out content feature analysis, classification, video skimming based on visual redundancy elimination and, finally, output abstract composition and coding. The solution is able to operate *on-line*, that is, sequentially processing and outputting content. For this purpose, a set of novel techniques have been developed, and existing ones have been adapted, focusing on their computational efficiency.

## 8.3 Overview of the News Abstraction System

In this section an overview of the news abstraction system architecture and functionalities is presented. The system is in charge of generating *on-line* multimedia abstracts of news bulletins by combining efficient and progressive techniques for shot classification, news stories segmentation, video skimming and video layout composition. The main challenge of the system is to build an abstraction system running *on-line*, that is, while the content is being broadcasted (e.g., for making the content available in an Internet portal simultaneously to the program creation), and finishing the abstraction process with a negligible delay after the original video broadcast finishes. The application of *on-line* algorithms to solve those problems is not a common approach and most studied works do not aim to develop efficient progressive generation solutions. The development of such algorithms raise a number of technical challenges due to the high efficiency required for the different processes carried out and because only partial information (the already received/broadcasted original content) is available at any given instant during the abstraction process.

The developed techniques could be applied as well for fast video abstract generation from already stored multimedia content (e.g., avoiding to store pregenerated video abstracts and requiring minimal resources for their on-demand generation) or for personalization purposes (e.g., allowing to create abstracts with different characteristics for each user, thus avoiding the need of storing thousands of summary versions per video).

In order to enable the *on-line* abstract generation and reduce the complexity of frame/shot analysis and comparison algorithms, the input video is divided in short segments, never longer than 30

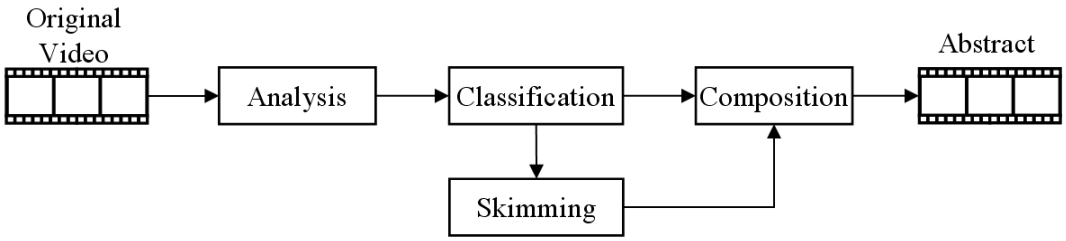


Figure 8.1: News Abstraction System Modules

frames (slightly over the commonly accepted minimal perceptible size of 25 frames [64] in order to increase the output smoothness in the video skimming stage), which are processed sequentially, being analyzed, classified, selected or discarded separately in the different stages of the abstraction process. The length of the obtained video segments can be smaller than 30 frames in cases where the segment contains a shot change. Such small granularity in the video processing enables to output partial results from the abstraction process when the original video has not been completely received. As depicted in chapter 3 taxonomy, most common approaches deal with visual classification, skimming and composition problems without taking into account computational constraints such as those needed for the *on-line* and *real-time* operation modalities.

As depicted in Figure 8.1, the system is divided in 4 modules:

- Analysis: The analysis stage is in charge of the extraction of low-level features from the original video stream for their use in the following classification and video skimming stages. The original video stream is divided in small segments and, for each, features such as the MPEG-7 Color Layout, frame differences, color analysis and face detection are extracted (extracted features are detailed in appendix B).

- Classification: In this stage each received video segment, annotated in the analysis stage, is classified based on the information provided by a set of independently trained SVMs for the different possible shot categories (see section 8.4). This stage works also at subshot level, with small video segments composed by a maximum of 30 frames so, once each segment is classified it can almost immediately be discarded, selected for the composition stage or sent to the skimming stage. The actions associated to each different video segment may vary depending on the configuration of the system as will be described in the following sections.

- Skimming: The skimming module is in charge of generating video skims from the combination of segments received after their classification. In this case, the binary trees algorithm described in chapter 5, section 5.5 is applied for the *on-line* generation of the video skims. The result of the skimming process is sent to the composition stage for its combination with other selected video segments.

- Composition: In the composition stage the final abstract presentation is generated. The final layout is a combination of resized video segments rendered in the foreground of the image together with full sized segments in the background plane. By default, the system generates video abstracts with a foreground window including the anchorperson's complete story introduction (which is tipically the most visually redundant part of the news story) while, in the background,

Figure 8.2: Layout for the News Video Abstract

a condensed video skim of the news report is presented (see Figure 8.2). The configuration of the abstract presentation layout can vary depending on the desired abstract characteristics as will be explained in section 8.5. The usage of the anchorperson's audio introduction is similar to the approach proposed in [12] although, in that case, the visual composition of the anchorperson is not carried out. Moreover, the algorithm presented in [12] does not work *on-line* (the whole video is needed to begin the process), it is applied for individual news stories that must always begin with an anchorperson shot, and it does not provide flexibility in terms of summary length, number of stories in the news bulletin and arbitrary number of anchorperson appearances.

The system is prepared for *on-line* processing of arbitrary length news bulletins. This approach, apart from the previously enumerated functionalities in terms of instant abstract availability, efficiency and personalization potential, would be easily adaptable to continuous running of the abstraction process for 24-hour news broadcasting and to any other kind of broadcasting or recording systems (e.g. video surveillance systems).

## 8.4 News Content Classification

The first stage in the abstraction process consists in the classification of the incoming video segments in the different possible categories included in a news bulletin. The available corpus for development and testing is constituted by 54 complete news bulletins, about 28 minutes long each, provided by Deutsche Welle to the IST-FP6-027685 Mesh project [1], totaling about 25 hours of news content. The basic structure of the news bulletins is quite similar to those identified in previous works [174, 175, 12, 177] and consists of a number of concatenated news stories, each introduced by an anchorperson section and followed by a visual report with the details of the story. The assumption of such a basic structure has been successfully applied for the segmentation of news stories but a further refinement would be useful for a meaningful abstraction process: inside a news bulletin many other types of shots, such as reporters, interviews, commercials, etc., can be found. By observing the available content, the following types of shots have been identified:

[1] http://www.mesh-ip.eu

- *Anchorperson*: In most of the observed cases the starting section of each news story consists in the anchorperson reading an introduction about the incoming report. In several cases, it is possible to find interleaved maps and graphics showing additional information about the narrated events. This kind of shots are mainly low activity shots with frontal face appearance in certain fixed locations in the image and static background.

- *Animation*: Synthetically generated animations which are used as transitions between sections in the news bulletin or at the beginning or end of the TV program. As in the maps category, the color palette is limited but, in this case, the shot activity is higher.

- *Black*: In some of the news bulletins, some completely black shots have been found in section transitions. In [179], those kind of shots are used as a clue for the detection of commercial sections.

- *Commercial*: Commercials are included in some of the news bulletins. The characterization of this kind of content based only in visual features is very difficult, as it can include shots with many different characteristics.

- *Communication*: The anchorperson maintains a conversation with a reporter or a relevant character. The reporter and, in some cases, the anchorperson are shown in a split-screen layout with several synthetically generated areas (maps, text information).

- *Interview*: As the reporter shots, one or more persons appear speaking outside the studio. It is even more likely to contain non-frontal faces and camera movements than the reporter shots.

- *Map*: Static shots showing synthetic images with the localization of the narrated news, in the DW news bulletin case, rendered with a limited color palette.

- *Report*: Once the anchorperson introduction has finished, the extended news report begins. It is narrated by a different voice to the anchorperson one and it is mainly composed by natural non-static shots. Nevertheless, the report can include interleaved shots with reporters, interviews, maps, etc.

- *Reporter*: Shots with a reporter providing further on-site explanations about the story. This kind of shot includes the reporter frontal face but, in many cases, are recorded in outdoor localizations, shots are not static, face positions are not fixed and illumination conditions are more variable than in anchorperson shots.

- *Studio*: Those are transition shots included in the DW news bulletins showing the TV set from different perspectives at the beginning or end of the news bulletin.

- *Synthetic*: As the maps category, synthetically generated static images associated to the news story which provides additional information (e.g. stock market information, sport classifications). The map category could be considered as a subset of this category.

- *Weather*: In the DW news bulletins the weather reports are mainly synthetic generated animations with predominant blue and green colors.


Figure 8.3: News Shot Categories

Figure 8.3 shows a typical example for each of the defined categories (except *Black* and *Commercial*). The most usual ones are the *Anchorperson* and *Report* categories, which can be found in almost every news story. The *Weather*, *Animation* and *Studio* categories are not associated to news stories and usually appear at the beginning or end of a news bulletin or as transitions between different parts of it. The *Map*, *Communication*, *Report* and *Synthetic* categories are usually found interleaved with other shot categories as part of a news story but may not appear at all.

The proposed system must process the original content and generate the output abstract progressively as the content is broadcasted, the whole abstraction process must fulfill certain requirements related to the efficiency and progressive operation. Small video segments will be the basic processing unit in all the stages of the abstraction process allowing to provide the needed granularity for *on-line* generation while reducing the complexity of shot analysis processes and comparisons dependent on the video segment length. In addition, the small video segment approach reduces the dependency on accurate shot boundary detection systems and enables the possibility of eliminating intra-shot redundancies (other systems in which the basic unit is the shot do not allow to discard only short portions for the reduction of visually steady segments length). This approach has been successfully applied in our previous *on-line* video abstraction works [96, 57, 56].

The fragment classification process relies in the fast extraction of low-level features from the original content which are feed to a SVM classifiers structure in charge of labeling each incoming video fragment in one of the existing categories. The feature extraction and classification processes must perform in a very efficient way due to the existing time constraints. A detailed description of the extracted features, training process and classification performance results can be found in appendix B.

## 8.5 Abstraction Process

In this section the news story abstract creation process is detailed. It is assumed that the anchorperson provides the essential audio information to allow the users to get an idea of what each news story is about and that, in most cases, it is followed by a report section in which the introductory information is extended. From this starting point three abstraction strategies are combined:

- Video Composition: The simultaneous display of different video segments allows to reduce the video abstract length, condensing the information presented. The anchorperson segments contain compact and high-interest audio information but not relevant visual information while the report sections include extended audio information together with relevant visual content. It is possible to take advantage of this particularity of the news bulletins by presenting the anchorperson segments, that provide a natural audio abstract of the news story, in a reduced window with a full-size background composed of the most relevant visual information of each news story.

- Video Skimming: The more relevant video segments from a visual point of view, those corresponding to the news story report, are selected to be displayed in the full-size background of the abstract layout. Any kind of video content usually contains redundant visual information so the news reports length can be reduced with a video skimming process and for this purpose the algorithm depicted in chapter 5, section 5.5 is applied.

- Segment Filtering: Segments which are included as part of the news story reports section but do not provide relevant visual information can be directly eliminated. For example, if it is considered that reporter or interview segments being part of the news report do not provide additional visual information about the news story, they can be discarded.

The selection of which video segment categories, from those defined in section 8.4, should be displayed in the small foreground window, which should be skimmed and presented in the background, and which should be directly eliminated is easily configurable. In the same way, the presentation layout could have different configurations, as shown in Figure 8.4 where different combinations of foreground-window/background are presented, depending on which shot category is to be emphasized.

In the implemented system, the layout depicted in Figure 8.4 (A) has been applied: the initial news introduction is completely kept (audio and images) and displayed in a reduced size window in the top-left corner of the image together with other information such as maps or synthetic content. This size reduction allows the full-size display of the more informative images from a visual point of view, in this case the abstraction of the report section of each news story.

The *on-line*, and hence progressive, operation mode implies to solve the correct alignment of anchorperson and corresponding report sections of each news story as the incoming video segments are received. Such sections can be of very different sizes and it is possible that, in some cases, one of them may not exist (for example special reports where no anchorperson introductory section exists). The *on-line* abstract generation process has been implemented by the definition of a 3-state machine and individual buffers for the temporal storage of foreground and background content. Figure 8.5 shows the 3-state machine diagram, temporal buffers and state change conditions (detailed in Table 8.1). The overall abstraction process begins in the *Start* state where the incoming, already classified video segments, are received. Any kind of incoming content is discarded with the exception of *Anchorperson* or *Report* segments which are accumulated until the conditions for changing to *News Intro* or *Report* states are fulfilled (sufficient amount of *Anchorperson* or *Report* content accumulated -see table 8.1-). In order to avoid undesired effects caused by incorrect segment classification all the state change conditions require the accumulation of a minimum number of segments from a given category, that is, a stable category classification. The *News Intro* state is reached when the incoming video corresponds to an anchorperson section and, in this case, all *Anchorperson*, *Map* or *Synthetic*

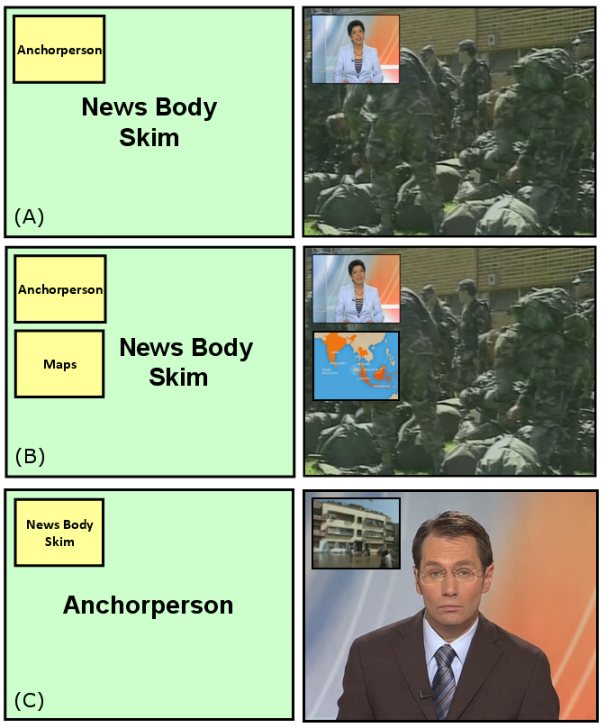


Figure 8.4: Abstract Composition Layouts

| Condition | Description |
|---|---|
| Anchorperson Detected | 5 seconds of accumulated consecutive *Anchorperson* segments. |
| Report Detected | 5 seconds of accumulated *Report* video segments. |
| Intro End | 5 seconds of accumulated no *Anchorperson, Map, Synthetic* or *Report* content. |
| Report End | 5 seconds of accumulated no *Anchorperson, Interview, Map* or *Synthetic* content . |

Table 8.1: State Change Conditions

incoming video is stored in the *Overlay Buffer* which will be later displayed in the reduced-size interface window. The *Report* state is reached from the *Start* or *News Intro* states when a number of *Report* segments have been received. In this state the *Interview, Reporter* or *Animation* incoming segments are discarded while the *Report* video segments are stored for a further video skimming. Both *News Intro* and *Report* states return to the *Start* state if a few seconds of unexpected content categories are received.

In a typical news bulletin structure the state machine will mainly switch between the *News Intro* and *Report* states. Each *Report* segment received in the *Report* state is skimmed by applying the binary-tree based summarization approach, targeting 1/3 of the original size (which corresponds to the average proportion between anchorperson and other kind of content in the news bulletins). The result of the video skimming process is progressively presented in the output abstract background. If the *Overlay Buffer* contains previously stored content it is presented simultaneously in the foreground reduced-size window obtaining the anchorperson-report synchronization, otherwise, if no foreground content is available, the overlay window is not displayed.

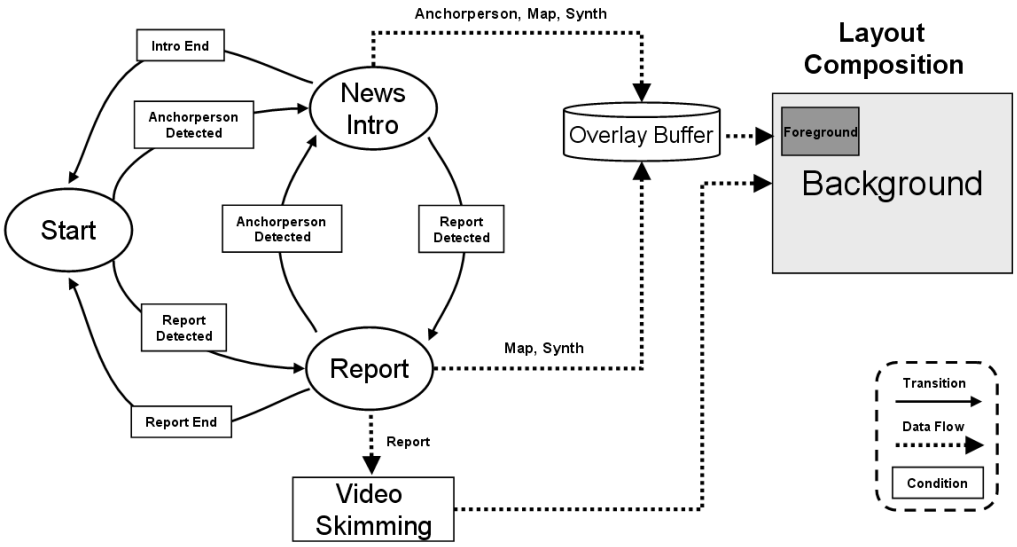


Figure 8.5: State Machine for Abstract Generation

Each time the *News Intro* state is reached the *Overlay Buffer* is flushed to the abstract output so, if the report video skim of a news story is shorter than the anchorperson introduction, the abstract corresponding to that story will finish with a full-screen anchorperson. Additionally, at the beginning of the foreground/background composition, when the *Report* state is reached, the first seconds are composed only by a full-screen anchorperson, taking out part of the *Overlay Buffer* content, before making the foreground and background composition. This mechanism, besides providing a pleasant edition effect, helps to avoid incorrect anchorperson-report alignment in situations when, after a news report, the anchorperson makes a short comment about the preceding news story before starting with the following one. For dealing with those cases in which the anchorperson section is too long, a length limit has been defined for the *Overlay Buffer*. If such limit is exceeded the buffer begins to be progressively displayed in full-screen size automatically, avoiding excessive delay in the abstract outputting and memory consumption for the storage of too long video segments.

Figure 8.6 shows a simple example of the abstraction process for two consecutive news stories. Both abstracts begin with the full-screen anchorperson followed by a simultaneous display of the anchorperson and report skim sections in a composed layout. Finally, the first story ends with a full-screen report skim while the second one, where the video skim is considerably shorter, finishes with the news story in a full-sized layout.

The proposed model enables the sequential processing of the incoming video, and therefore the *on-line* abstract generation with progressive output: each received video segment is immediately analyzed, classified and processed with one of the defined actions according to the state in which the abstract generation is and each news story abstract is finished with negligible delay once finished.

Table 8.2 summarizes the average time[2] required by each abstraction stage and for the complete abstraction process if considering a 28 minutes original news bulletin and a 30% length generated abstract. The average amount of time required for a video abstract generation is below 1/3 of the original news bulletin length.

[2]Hardware platform: Intel Core 2 Duo @2.53GHz with 4GB of RAM.

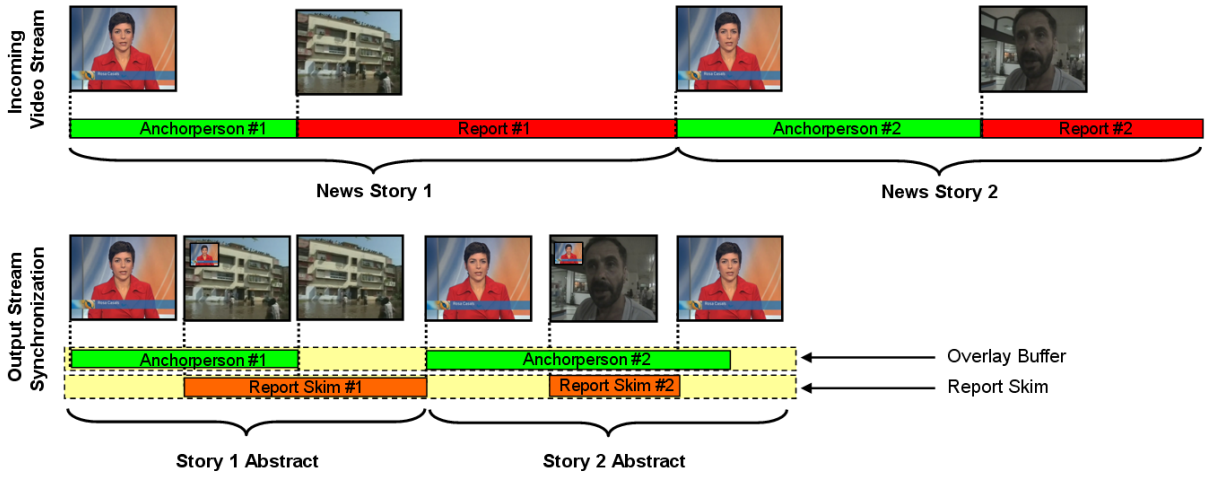


Figure 8.6: Abstraction Example

| Step | Average Time (per second) | 28 min. Bulletin |
|---|---|---|
| Decoding | 120.3 ms | 202.10 s |
| Feature Extraction | 98.84 ms | 166.05 s |
| Classification | 2.6 ms | 4.37 s |
| Skimming | 3.12 ms | 5.24 s |
| Composition | 40.51 ms | 68.05 s |
| Coding | 42.168 ms | 70.84 s. |
| Total | 326.81 ms | 516.65 s (~ 8'36".) |

Table 8.2: Average Abstraction Time (30% length abstract)

## 8.6 Conclusions

In this chapter, a system for the *on-line* generation of complete multimedia news bulletin abstracts has been described. The *on-line* operation mode requires the sequential processing of the incoming video as well as progressive output generation and implies to work with only partial original content information (the already broadcasted content at any given instant). Considering the *on-line* and efficiency requirements, the individual way in which the different techniques for content classification, video skimming and abstract composition have been applied and how such techniques have been combined represents a novel way to deal with news abstract generation.

A validation of the system has been carried out with a set of user tests in which the quality and representativeness of the proposed approach have been evaluated by the visualization of several of the generated abstracts by different users. The user evaluation includes examples of incorrectly composed news stories for the study of their impact in the users perception. Details about the validation tests and obtained results can be checked in appendix C and demonstrate a very high acceptance by the users with respect to the summaries representativeness and quality. The quality of the video skimming approach, previously discussed in chapter 7 within the scope of the TRECVid 2008 BBC rushes summarization task [133], is implicitly confirmed with the user tests carried out.

The generalization of the abstraction algorithm has been validated with its application to different news content providers and the obtained results demonstrate that the developed system provides a complete solution for instant news abstract availability during or at the end of the broadcast. The progressive abstract generation scheme allows the continuous abstract generation for 24-hour channels, and provides new application possibilities such as its extension to other fields where continuous abstraction could be applicable (for example surveillance recordings). The proposed system approach can be extended to *real-time* visualization systems where abstracts are generated on viewing time and could allow many personalization and interactivity possibilities as described in the next chapter.

# Chapter 9

# Real-Time Interactive Video Summaries Player

## 9.1 Introduction

In this chapter, a novel application for the *real-time* generation and visualization of pre-stored videos, named 'Real-time Interactive video Summaries Player' -RISPlayer-, is presented. The *real-time* abstraction, as defined in chapter 4 section 4.2, consists in an abstraction operation modality in which the video summaries are generated *on-line* and fast enough so that the results can be watched in *real-time*, that is, while they are generated, without pauses and with a small delay. Furthermore, the RISPlayer provides an interactive video abstract generation process, allowing to vary and control the generation parameters of the algorithm on the fly and immediately watch the results of the parameters modifications.

For the implementation of the enumerated features, the binary tree based *on-line* summarization algorithm defined in chapter 5 section 5.5, including the fragment filtering functionalities defined in chapter 5 section 5.6.2, has been integrated with a visualization and parameter control interface. The application takes advantage of the functionalities provided by the algorithm in terms of type of generated summary and balance between the generation computational performance and quality control.

The rest of the chapter is organized as follows: section 9.2 overviews the developed application, describing their basic functionalities and components. In section 9.3, the mechanisms for enabling the *real-time* visualization of the video summaries, with adaptive process complexity control and preventing pauses in the visualization, are discussed. Section 9.4 presents the application interface and the different configurable parameters and mechanisms for user interaction during the summary generation process. Finally, conclusions are presented in section 9.5.

## 9.2 RISPlayer Application Overview

Figure 9.1 shows an overview of the principal elements of the RISPlayer. The original video is decoded and the extracted BUs (generated splitting the video at regular intervals) are inserted as selection or discard leafs in a summarization tree -figure 9.1 (A)-. Once certain number of BUs have been decoded and inserted in the summarization tree and such tree reaches a predefined depth, the root node ad-

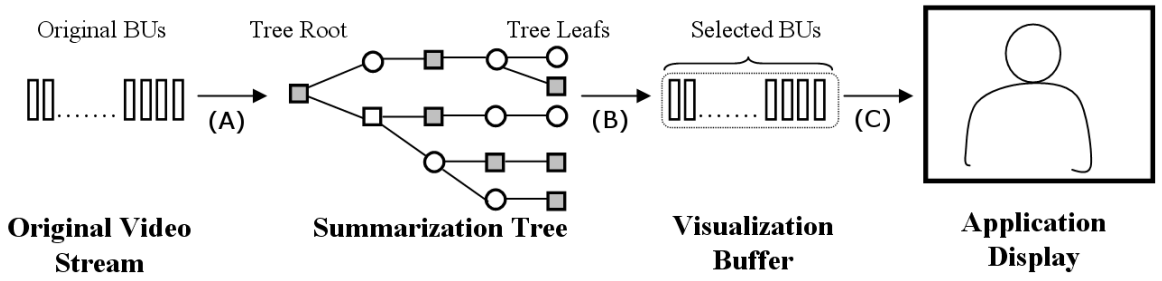


Figure 9.1: RISPlayer Components

vances progressively determining which of the incoming BUs are selected to be displayed and which of them are discarded (see details of binary tree based *on-line* summarization in chapter 5 section 5.5). Instead of writing the selected BUs in a video file, the BUs are inserted in a visualization Buffer -figure 9.1 (B)-. The application display is constantly extracting BUs from the visualization buffer -figure 9.1 (C)- and displaying them at normal play rate.

It must taking into consideration that, when the visualization buffer is empty, the application display will not have any content to display and, therefore, the summary visualization will stop. In order to avoid an empty visualization buffer, the summarization algorithm must decode the incoming BUs fast enough, not just reaching the *on-line* processing of the original video but outputting the selected BUs at a fast enough rate for enabling its *real-time* visualization. Such conditions implies that the incoming video must be processed fast enough so the selected portion of the original video could be displayed at normal play speed without interruptions. For example, in the case of generating a 1/10 length summary of an original video displayed at 25 frames per second, it must be processed, at least, at a speed of 250 frames per second so the 1/10 length summary could be displayed at normal speed (see chapter 4 section 4.3 for more details). Of course, the desired summary length ratio will have a direct influence in the required summary generation speed. Moreover, depending on the location of the selected BUs, even reaching the required processing rates, it is possible that, if no BUs are selected for a long time interval, the visualization buffer could empty out. Several mechanisms, described in section 9.3, have been implemented in the developed RISPlayer application to deal with the mentioned issues and assure a continuous visualization of the generated summary.

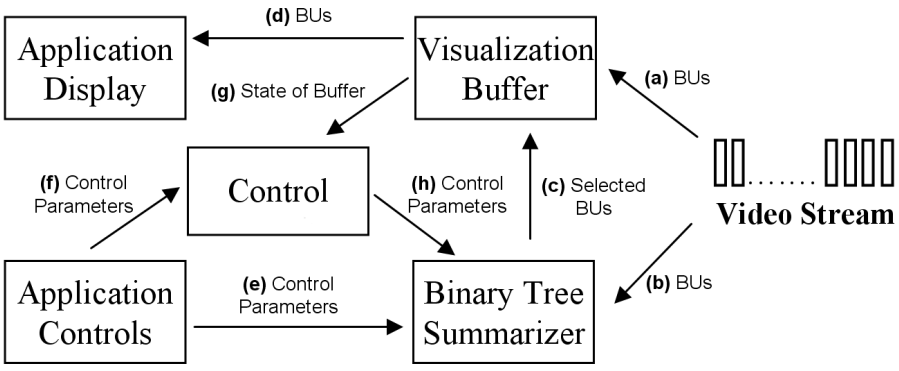


Figure 9.2: RISPlayer Information Flow

Figure 9.2 shows the different functional modules integrated in the system and the data and control flows between them. Depending on the mode in which the application is operating, there will be a different data flow between the components:

- *Normal play*: The RISPlayer behaves as a normal video player, displaying the complete original video at normal speed. In this case, the BUs travel directly from the original video to the visualization buffer -figure 9.2 (a)- and from there to the application display -9.2 (b)-. In this mode the summarization tree is not generated.

- *User-controlled summarization*: In this mode, the RISPlayer generates and displays a video summary by applying the summarization tree algorithm. The original video BUs are inserted in the summarization tree -figure 9.2 (b)- and, from there, the selected BUs are inserted in the visualization buffer -figure 9.2 (c)-. The user can control the summarization tree parameters (tree scoring, depth or number of branches) making use of the application interface controls -figure 9.2 (e)-. This mode grants total control of the summarization parameters to the user so, depending on their choices, the visualization buffer can empty out if the chosen summary generation parameters are too computationally demanding (e.g. an excessive amount of nodes in the summarization tree).

- *Assisted summarization:* In case of requiring a *real-time* visualization of the video summary (that is, viewing the generated summary without pauses), it is possible to set this operation mode. In this case, the application takes control over the summary generation parameters. The system, based on the parameters established by the user -figure 9.2 (f)- and on the information about the state of the visualization buffer -figure 9.2 (g)-, controls the summarization tree creation -figure 9.2 (h)- aiming to keep a constant flow of selected BUs for its visualization. Details about the implemented mechanisms are provided in section 9.3.

The developed RISPlayer provides different functionalities for the fulfillment of different user requirements, being able to operate like a common video player or a summary viewer in which the user can control the generation parameters (or let the system control them for *real-time* visualization of the summaries). In the following sections, the mechanisms for enabling the *real-time* summaries generation will be described, as well as the interface and functionalities of the application.

## 9.3 Visualization Buffer Control Strategies

As mentioned in the previous section, one of the main functionalities of the RISPlayer is to offer *real-time* video summarization. In the *user-controlled summarization* mode, summaries are generated applying the user-defined generation parameters (see section 9.4 for details about the configuration mechanisms). This may produce that a summary, even being *on-line* processed, could not be watchable in *real-time* due to the selection of parameters which may prevent the system from a processing fast enough to avoid pauses in the summary visualization. In the case of the *assisted summarization* mode, the RISPlayer provides automatic control of the summary generation parameters. Such automatic control focus in two fundamental aspects of the summary generation process: the speed of the summarization tree and the visualization buffer filling ratio, which are described in the following subsections.

### 9.3.1 Summarization Tree Speed Control

The main mechanism for controlling the speed of the process is the variation of the summary tree generation parameters. For such purpose, the tree depth and number of nodes (see chapter 5 section 5.6.2 for a description of both concepts) can be modified achieving different computational performances. In the evaluation of the binary tree summarization approach (see chapter 7 section 7.3.3), it was confirmed how the reduction in the tree depth and number of branches produced an increment in the computational performance of the algorithm with the drawback of a reduction in the summaries quality. In the RISPlayer automatic control mode, the size of the summarization tree is set according to the level of occupation of the visualization buffer. We will define as $bufferOccupation \in [0,1]$ the ratio of occupation of the visualization buffer (assuming that there is a maximum defined occupation) and *maxDepth*, *minDepth*, *maxLeafs* and *minLeafs* as the maximum and minimum depth and possible leafs (i.e. possible branches) in the tree. On every iteration of the process, that is, every time a new BU is received, the depth and number of branches for the summarization tree are recalculated as:

$$treeNodes = minNodes + (maxNodes - minNodes) \cdot bufferOccupation \tag{9.1}$$

$$treeDepth = minDepth + (maxDepth - minDepth) \cdot bufferOccupation^2 \tag{9.2}$$

In this way, the parameters of the tree vary according to the number of frames in the visualization buffer: in case of a high occupation ratio there is time enough for a more time consuming summarization process and, therefore, the depth and number of leafs of the summarization tree are increased aiming for a higher quality summary. In the opposite case, when the occupation ratio is low, the size and branch population of the tree are reduced for a faster summary generation and, therefore, visualization buffer filling. It must be pointed out that, as can be observed in equations 9.1 and 9.2, the tree depth and nodes are not calculated in the same way: in the case of the *treeDepth* calculation, the *bufferOccupation* value is squared, so the depth of the tree rapidly drops as the *bufferOccupation* does. The reason for the different treatment is to favor higher *treeNodes*/*treeDepth* ratios, which, as seen in chapter 7 section 7.3.3, produce better summarization results.

### 9.3.2 Visualization Buffer Filling Control

The described summarization tree speed control mechanism permits to avoid, in many cases, the empty out of the visualization buffer (with the consequent pause in the summary visualization). However, it may not be possible in all cases: a too reduced output summary length or a non-uniform distribution of the selected BUs may produce such empty out. Assuming that, in the case of *real-time* summarization, it is preferred to avoid pauses in the visualization than producing a exact length summary, a mechanism for preventing the visualization buffer from empty out has been implemented. The most straightforward solution would consist in just forwarding not selected BUs to the visualization buffer, but the summarization trees algorithm can provide more sophisticated solutions allowing to control the impact in the characteristics of the generated summary. In the RISPlayer implementation two procedures have been adopted for controlling the occupation ratio of the visualization buffer: selection of tree branches with a higher short-term probability of adding BUs to the visualization buffer; and the conversion of 'discard' nodes in 'inclusion' nodes.

#### Tree Branch Selection

The first approach consists in calculation of the distance from the first selected nodes contained in each tree branch to the root node. If the *bufferOcupation* runs below a predefined value, the tree summarization process is forced to keep the branch which contains the inclusion node closer to the root, instead of keeping the branch containing the highest score nodes. In this way, by accepting a negative effect in the generated summary quality, the probability of a fast addition of selected BUs to the visualization buffer increases.

#### Node Type Conversion

Although the proposed mechanism allows to produce a more convenient distribution of the selection of BUs, the pauses in the summary visualization are still possible due to the summary length constraints. For this reason, the second mechanism is applied when the visualization buffer empties out. In such case, the root node of the summarization tree is marked as an 'inclusion' node regardless of its previous state. In this way, the corresponding BU will be automatically included in the visualization buffer. Such state change will, of course, distort the calculated characteristics and scores of the child tree branches appended to the node and, for this reason, all the nodes pending from the modified one are reevaluated so the modification can be taken into account. With this mechanism the scores associated to all the calculated possible summaries maintain the coherence with the generated summary modification that is introduced by the inclusion of the root node in the summary.

## 9.4 RISPlayer Application Interface

All the components and internal mechanisms described in previous sections have been integrated in an application which serves as mock-up for the validation of the *real-time* summarization concept as well as for the exploration of the binary tree based summarization possibilities. Figure 9.3 depicts the interface elements of the RISPlayer application.

The main component of the application is the 'Display' element, where the original video and summarized versions are displayed. The 'Display' contains the 'Play Information' area where the video time and play mode ('Play', 'Pause' or 'Summary') are displayed. The 'Summary Information Bar' and 'Timeline' components are located below the main 'Display' component. Figure 9.4 shows a more detailed view of both elements. The 'Timeline' displays the current position of the video with a slider which can be positioned in any desired position. The 'Summary Information Bar' contains several details about the generated summary: The blue filled bar and red line correspond to the current display position in the original video; the yellow marks correspond to the positions of fragments already selected for the video summary; the orange line locates the position of the root node of the summarization tree and, therefore, the video fragments between the current display position (red line) and root node (orange line), have been already selected or discarded to be part of the video summary; finally, the green line represents the current position of the summarization tree leafs, that is, the position of the last BUs decoded from the original video and appended to the summarization tree. The distance between the orange and green line represents the summarization tree depth.

The 'Play', 'Pause' and 'Summary' buttons included in the interface (see figure 9.3) allow to change between the different display modalities. When the 'Summary' button is clicked, a summarization tree is generated starting from the current play position. In that mode, the RISPlayer displays only the selected positions (corresponding to yellow marks in figure 9.4) skipping the rest of the original video

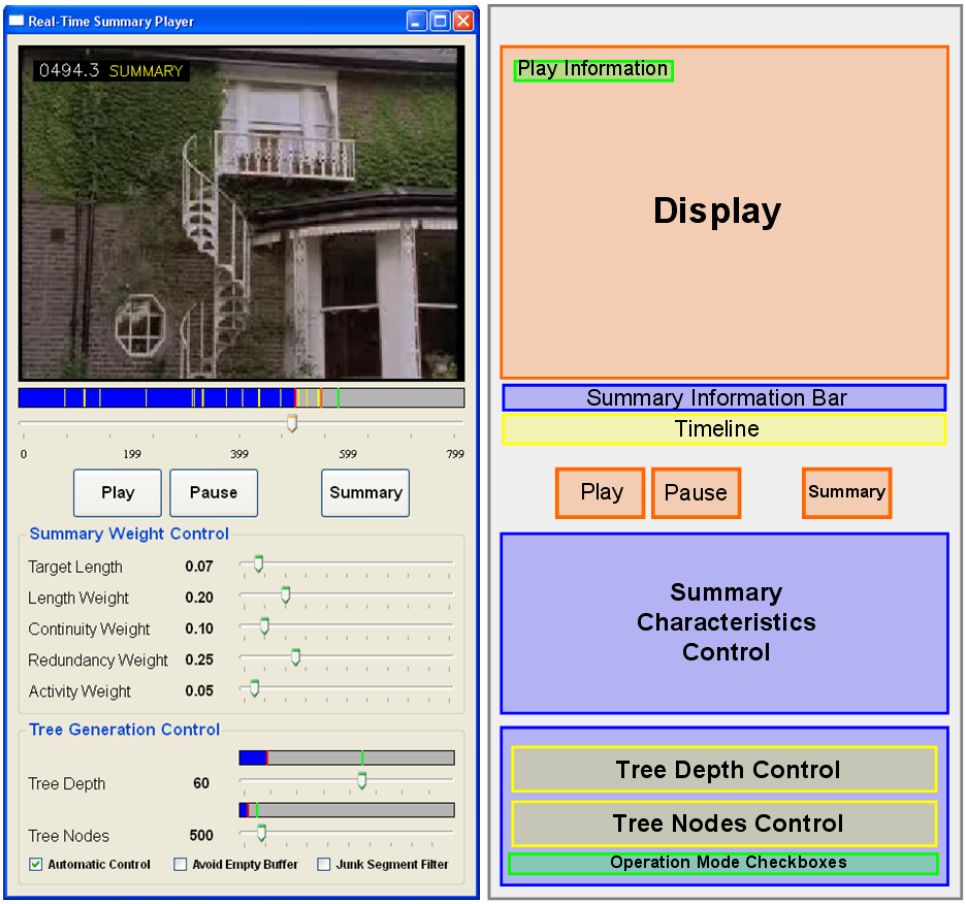


Figure 9.3: RISPlayer Interface Elements

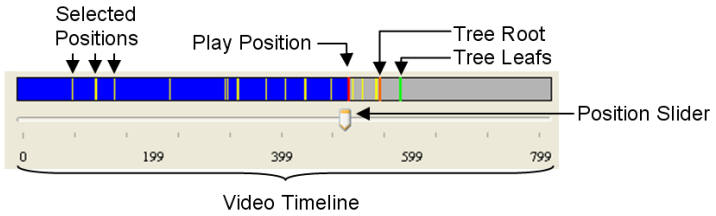


Figure 9.4: RISPlayer Summary Information Bar and Timeline

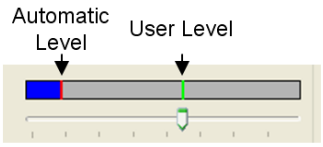


Figure 9.5: RISPlayer Summary Information Bar and Timeline

positions. If there are not available selected positions, the video play is stopped until the summarization algorithm marks more positions as included in the summary.

The 'Summary Characteristics Control', shown in figure 9.3, allows to control the characteristics of the generated summary making use of an implemented scoring mechanism which allows to assign different weights to the length, redundancy, continuity and activity of the generated summary (see details in chapter 5 section 5.5.3). The user may change the summary generation weights and watch the results on the fly. Each time a weight parameter is changed, the summarization tree is rebuilt starting in the last displayed video position. In the same way, when the time slider position is changed, a summarization tree is rebuilt from the new position.

The 'Tree Nodes Control' and 'Tree Depth Control' components allow to define the applied tree depth and number of leafs. In the case of automatic control of the tree depth and nodes (see previous section), the value set in both sliders determine the maximum values for both characteristics and the blue filled bar above the sliders show the automatically chosen values (see figure 9.5).

Finally, the 'Operation Mode Checkboxes' shown in 9.3 allows to activate the *assisted summarization*, with the application of the automatic visualization buffer control depicted in section 9.3. The 'Automatic Control' activates the automatic control of the summarization tree depth and number of leafs. The 'Avoid Empty Buffer' checkbox determines whether the mechanisms implemented for avoiding the empty out of the visualization buffer are applied or not. Finally, the 'Junk Segment Filter' enables the application of BU filtering implemented for the TRECVid campaigns, designed to avoid junk content such as blank frames, color bars or clapboards (see 5 chapter section 5.6.2).

## 9.5 Conclusions

In this chapter we have described RISPlayer, a video player designed for the experimentation with the interactive generation and visualization of *on-line* and *real-time* generated video summaries. The application takes advantage of the configurable properties of the binary tree based summarization algorithm for generating customized video summaries in *real-time*. The user is able to interactively control the different weights which guide the summarization process generating different types of video summaries (see analysis in chapter 7) as well as for controlling the quality of the generated summaries by varying the summarization tree depth and number of branches. Moreover, the RISPlayer includes the possibility of automatically controlling the summarization process for assuring *real-time* visualization of the summaries. Such automatic control mechanisms allow as well the self-adaptation of the RISPlayer, which could automatically decrease its computational requirements for slow machines or generate more computational demanding and precise summaries in fast machines.

The implemented application demonstrates the flexibility of the proposed summarization algorithm and outlines future possibilities for user-oriented *real-time* interactive summarization and video retrieval applications.

# Part VI

# Conclusions

# Chapter 10

# Conclusions

## 10.1 Main Contributions

In this thesis, a complete study of novel techniques for *on-line* and *real-time* video abstraction has been carried out. The overall work analyzes the different possible aspects of the problem: study of existing works; establishment of an analysis framework; definition and previous analysis of the targeted problem; proposal of specific solutions in terms of novel algorithms; analysis and evaluation of the proposed solutions; and, finally, the application of the proposed solutions in real applications.

The main contributions of the present work are the following:

- An abstraction systems taxonomy and video abstraction architecture (chapter 3).
- An analysis and definition of the *on-line* and *real-time* abstraction modes as well as the implications and constraints related to the development of such systems (chapter 4).
- Proposal of novel algorithms for *on-line* and *real-time* video skimming (algorithm description in chapter 5 and algorithm evaluation in chapter 7).
- Proposal of a novel system for the automatic evaluation of video abstraction approaches (chapter 6).
- Development of an application for the generation of broadcasted news abstracts integrating *on-line* techniques for segment classification video skimming and abstract presentation composition (chapter 8).
- Development of a novel application, RISPlayer, for the *real-time* interactive generation and visualization of video skims (chapter 9).

This work starts by providing an overview of existing abstraction approaches and classifications proposed in the literature (chapter 2). Based on the study of such approaches, an unified framework (chapter 3) constituted by a novel abstraction systems taxonomy and a generic architecture able to model most existing abstraction techniques, has been proposed. The taxonomy classifies the video abstraction approaches according to their external and internal characteristics and represents a novel point of view convenient for the analysis of the operational characteristics of video abstraction systems. On the other hand, the proposed architecture describes how to characterize almost any abstraction approach by dividing the process in three basic conceptual stages (namely 'analysis', 'scor-

ing' and 'selection') and considering the abstraction process as a flow of BUs (fragments extracted from the original video) trough the different stages.

The defined framework defines concepts and terms which have served for the analysis of existing approaches and for the proposal of new ones. The *on-line* and *real-time* abstraction modes have been defined and the computational constraints associated to each mode have been established (chapter 4). Moreover, an analysis of potential implications of *on-line* and *real-time* operation modes in the underlying abstraction mechanisms has been performed.

Two novel *on-line* video skimming algorithms have been proposed taking into consideration the previously defined constraints and limitations (chapter 5). Both algorithms were designed targeting to their application for generic content abstraction. For this reason both of them rely in visual redundancy removal mechanisms, although other aspects about the generated summaries (e.g. continuity, pleasantness) have been taken into account. The two algorithms differ in their complexity level with a first approach, based on a 'sufficient content change' mechanism, providing less functionalities and a second proposal, the binary tree based approach, with many customization possibilities. The binary tree based approach provides a generic *on-line* abstraction framework with integrated mechanisms for customized scoring, content filtering and scalability, in terms of computational performance, generation delay and summary quality. Two rushes abstraction systems, based on the proposed algorithms, were submitted to the TRECVid 2007 and 2008 BBC Rushes Summarization Tasks obtaining results comparable to the rest of participants (mainly submitting *off-line* approaches).

However, for a complete validation of the proposed algorithms, a more exhaustive testing was required. For such reason, a novel system for automatic summaries evaluation was proposed (chapter 6). The automatic evaluation system was developed making use of the submissions and corresponding evaluations of all participants in the TRECVid 2008 BBC Rushes Summarization Task, training individual predictors for several of the evaluation measures extracted in such campaign. The proposed system allows approximating human assessments based on computable features calculated from the video summaries. By making use of such system, the described *on-line* video skimming approaches were in-depth evaluated, analyzing their functionalities and capabilities compared with *off-line* approaches and demonstrating the possibility of the *on-line* generation of video summaries with quality levels comparable to *off-line* techniques (chapter 7).

Two applications, integrating the binary tree based *on-line* abstraction algorithm, were developed demonstrating the applicability of the proposed abstraction approaches. The first one is devoted to the generation of *on-line* abstract of multimedia news bulletins (chapter 8). It combines several techniques for the analysis and categorization of video segments (based on feature extraction and classifiers training depicted in appendix B), video skimming (based on the binary tree algorithm proposed in chapter 5) and layout composition. The complete system is able to process the original video in an *on-line* manner, allowing the application of the system for news abstraction on broadcasting time. The quality of the generated summaries was evaluated with both objective and subjective tests, carrying out a user validation campaign which obtained very good results (see details in appendix C).

The second application consists in a *real-time* interactive video summaries player -RISPlayer- (chapter 9). It has been developed integrating the binary tree abstraction approach and allows the *real-time* generation and visualization of video abstracts. The application provides the functionality of a normal video player being also able of generating video skims (in such case, only the selected portions of the original video are played). The application includes automatic mechanisms for controlling the abstraction computational performance, based in the scalability properties of the binary tree approach, and allows the user to interactively control the different weights for scoring and fil-

tering the video summary. The RISPlayer demonstrates the potential of the *real-time* summarization sketching the possibilities for future work on personalized and interactive video abstraction systems.

## 10.2 Future Work

Beyond the results achieved so far, there are several directions for the continuation of the work carried out in this thesis, mainly focused in the following aspects of the presented work:

- Identification of new types of content and potential applications of *on-line* video abstraction:
  - The generic *on-line* video skimming algorithms presented in this work are based in the application of a redundancy removal approach, which has been proven to be applicable for certain types of content (commercial movies, BBC rushes). The analysis of new types of content (e.g. sports, television series) for the validation of the proposed approach could be an interesting future direction.
  - One of the presented abstraction algorithms is devoted to the abstraction of broadcasted multimedia news, by combining specific algorithms for news content classification and composition, together with the generic binary tree-based video skimming algorithm. The identification of possible application scenarios for *on-line* systems and the development of specialized techniques for dealing with such kind of content is one of the promising future directions. Broadcasted content, continuous recording systems (e.g video surveillance), Internet video providers, low computational power terminals for video recording and storage (e.g. mobile terminal) are examples of potential application scenarios for *on-line* video abstraction techniques.

- *On-line* abstraction algorithms improvement:
  - The binary tree video skimming approach presented in this work provides a powerful framework for future *on-line* abstraction systems development. However, there is room for improvement, for example with the optimization of the computational efficiency of the algorithm, development of 'intelligent' strategies for improved branch selection and pruning or the experimentation with new scoring mechanisms and different types of video content.

- *Real-time* interactive abstract generation:
  - One of the most interesting aspects of the presented work is the possibility of *real-time* interactive abstract generation. Such kind of approaches will provide functionalities for allowing the user to interact with the video abstract generation process, watching the results of his actions on-the-fly and enabling the possibility of interactive navigation through the content. The application presented in this work focuses in the validation and experimentation with the possibilities of *real-time* and interactive abstract generation concepts. However, further improvements for enabling the usage of such kind of applications by non-expert users, the study of the possible interaction mechanisms with the application and the development of more personalization aspects are fields of interest for future research.

- Automatic summary evaluation system:
  - The development of automatic video summary evaluation mechanism has many potential applications and possibilities in the development and improvement of future video abstraction approaches. The work carried out demonstrates the feasibility of such automation, at least under the conditions and type of content applied in the TRECVid 2008 BBC rushes summarization task. Future work in this are will be focused on the improvement of the extracted features and prediction techniques as well as the validation of the developed techniques with new types content and evaluation measures. A further validation of the proposed techniques could enable a better understanding about what makes a video summary good and to improve existing video abstraction approaches.

# Part VII

# Appendixes

# Appendix A

# Example of Application of the Abstraction Systems Framework

## A.1 Introduction

Operative functionalities are a very relevant factor given the practical inconveniences for the implementation of complex abstraction systems in real environments. The huge amount and growth of content in video repositories forces to consider the abstraction systems operative characteristics and not only their output quality. It is not possible to find complex abstraction approaches in commercial systems where efficient but limited functionalities (keyframes, subsampling...) abstraction approaches are applied. There is a need to balance the abstract quality and the operational functionalities of the abstraction system itself. The first step is a clear definition of a set of common concepts and abstraction systems capabilities for comparing approaches in terms of provided operational functionalities. The taxonomy proposed in chapter 3 aims to define a standardized classification scheme for operative characteristics of abstraction systems allowing a clear specification of a system capabilities with independence of the underlying abstraction mechanisms, selection criteria or generated abstracts quality. Given the number of existing approaches, similar in many cases, the operational functionalities provide a different criterion for measuring the system's quality and its possible application scenarios.

The external abstract generation characteristics (defined in chapter 3, section 3.2.1) may have a relevant impact in many aspects of an abstraction approach: the output abstract size -*bounded*, *unbounded*- will necessarily affect a possible abstract presentation interface, the time needed to watch the abstract, etc. The system performance -*linear*, *non-linear*- can determine if an abstraction approach is applicable in real time, its integrability in a streaming video portal with hours of video being constantly uploaded or if it is most appropriate for an off-line processing scenario. A *progressive* system, able to generate a summary while the content is being generated or received, will allow new scenarios such as on-line abstraction of broadcasted content (for example making video abstracts of several simultaneous sport events for presenting one-minute summaries in a carousel way), instant surveillance recording reviewing and so on.

In the same way the internal video abstract characteristics (chapter 3, section 3.2.2) are relevant due to their influence in the described external characterization of the system. The size and type of BU will affect the quality and computational performance of the analysis, scoring and selection stages. Intra-BU analysis, rating and selection would ease the construction of progressive abstraction

systems as the scoring and selection of each BU do not depend on other previous or incoming BUs. When dealing with inter-BU analysis, scoring or selection, the performance of the system (linear, non-linear) will be influenced by the number of inter-BU comparisons, size of the BUs, etc.

The number of influences and dependencies between the different operations carried out within an abstraction process are manifold and hence the proposed architectural decomposition in different stages (see section chapter 3, 3.3.2) provides a mechanism for analyzing the behavior, functionalities and performance (in terms of computational effort and output quality) of a given system in a modular way. The research can be focused on the improvement of independent stages working on separate abstraction aspects. The proposed modularity will enable as well the modification of abstraction algorithms by exchanging those stages which make the system be classified in a specific category.

The following subsections provide practical examples of abstraction system analysis, classification and modeling according to the proposed taxonomy and architecture. Section A.2 depicts several examples of abstraction system decomposition while section A.3 provides a complete system classification, modeling and the formulation of different possibilities for providing it with alternative functionalities.

## A.2 Abstraction System Decomposition

This section is devoted to show different abstraction systems decomposition according to the architectural models depicted in chapter 3, section 3.3.2, specifying the tasks carried out on every different abstraction stage.

The most simple defined abstraction model is the Non-Iterative architecture including a single 'selection' stage. One of the video skimming systems depicted in [64], which simply subsamples the original video frames at a given rate, is suitable to be modeled with such architecture. In this case the system's BUs are single frames and only a 'selection' stage with the subsampling mechanism is included. Figure A.1 (a) shows the abstraction system internal mechanisms mapped to the defined architecture. The 'User Preferences' are, in this case, the choice of the selected output length ratio. The same architecture is valid for similar approaches, for example considering small video fragments as BUs instead of individual frames or systems where the video abstract is generated just selecting a fixed number of BUs from the beginning of the video (Open Video Project[1]). No 'analysis' nor 'scoring' stages are present in the architecture and there are not dependencies between BUs easing the implementation of those systems as on-line approaches.

In [39] a keyframe extraction system which can be modeled as a 'Non-Iterative' system with 'analysis', 'scoring' and 'selection' stages is presented. The approach consists on the extraction of one representative keyframe per incoming video shot. In this case, the frame which accumulated motion activity is half the value of the entire shot is selected. Figure A.1 (b) depicts the distribution of the abstraction system elements in the three stages. In the 'analysis' stage a 64 levels RGB color histogram and motion vectors, which standard deviation is calculated as motion activity measure, are extracted from the incoming video stream. The 'scoring' stage includes a shot change division which is probably based in the extracted color histogram (authors do not provide details about the employed mechanism) and the accumulated motion activity per frame within a given shot is calculated as well. Finally, in the 'selection' stage, the shot frame with accumulated motion activity value equal to half of the shot total accumulated activity value is selected as keyframe. As authors point, this system is suitable for

[1] www.open-video.org

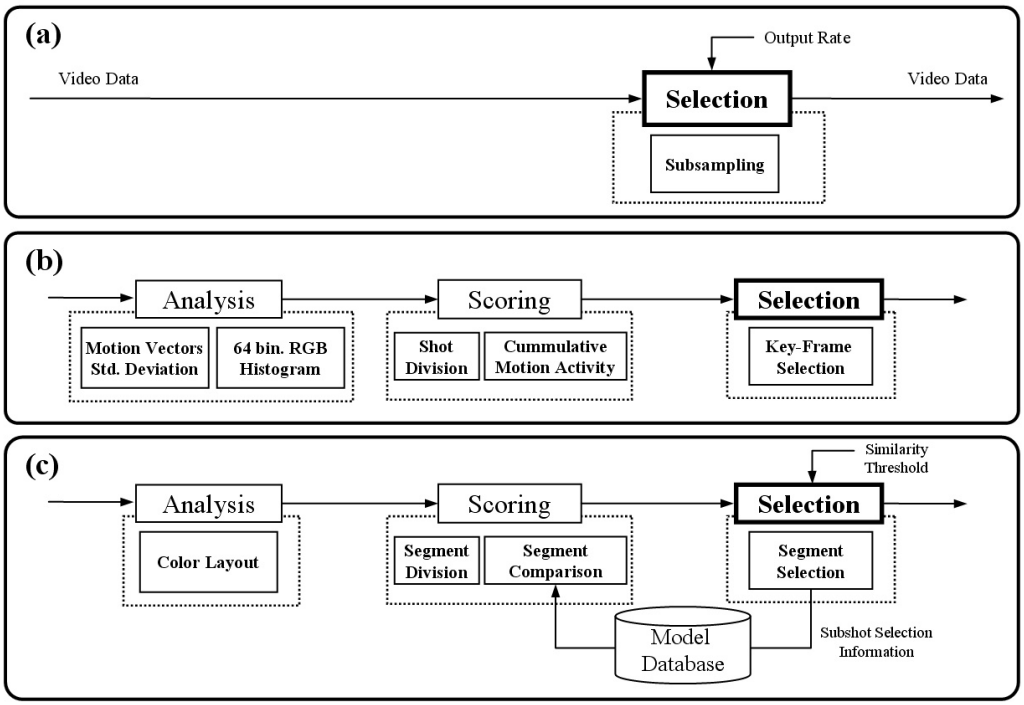


Figure A.1: System Decomposition Examples

*on-line* operation mode although the size of the output abstract can not be controlled. Nevertheless, a straightforward improvement could be the selection of a variable number of keyframes from each shot depending for example on each shot length.

A 'Non-Iterative' system with 'analysis','scoring' and 'selection' stages as well as a Metadata Feedback mechanism is depicted in [56]. Such system is an on-line 'sufficient content change' video skimming approach, that is, incoming video segments are included in the output summary only if they have a high enough visual difference with respect to other included segments. In this case the system also provides a filtering mechanism so segments too similar to the ones previously defined as 'junk' are discarded. Figure A.1 (c) depicts the system architectural distribution. In the 'analysis' stage the MPEG-7 Color Layout [144] is extracted in a per-frame basis. In the 'scoring' stage, incoming frames are grouped in fixed size segments with a variable step sliding window and the visual similarity (based on the Color Layout descriptor) with previously included segments, which information is stored in the 'Model Database', is calculated. The visual similarity with pre-stored content is calculated as well providing a filtering mechanism for undesired content (in this case junk content such as blank frames). The 'selection' stage is in charge of selecting those video segments which similarity to previous segments or the models in the database are over a user defined threshold. The Color Layout of the selected fragments is included in the model database for further comparisons. The output abstract size will be determined by the original video characteristics as well as the defined thresholds and is not possible to control. The main reason is the system *on-line* operation mode, avoiding off-line approaches, e.g. clustering, which would permit a precise control of the output abstract length.

## A.3 Abstraction System Classification and Modeling

For illustrating the complete application of the proposed classification taxonomy and modeling framework to a real example we will consider a recently published abstraction approach [180] presented in the last TRECVid BBC Rushes Summarization Task [133]. We will focus in the operative characteristics of the algorithm under analysis without discussing the participant's summarization evaluation results.

The algorithm works as follows: in a first stage the input is segmented into shots by the application of a combination of transition detectors (fade in, fade out, fast dissolve, dissolve, etc.) relying on a number of extracted features. Such features are extracted in both *intra-BU* (color histogram, edge and related statistical features) and *inter-BU* (motion intensity, histogram changes) analysis (if we consider the frame as the BU in the first stage). After the shot boundary detection, the system uses the shot (variable length) as BU. In the next step, junk frames (in the case of the BBC rushes content they are clapboards, black frames, etc.) are eliminated considering the extracted visual as well as audio features (in this case no details are provided in the paper to infer the kind of BU and modality -intra or inter BU- of the audio analysis), and the shot boundaries are modified according to the results. Next, the shots are subdivided in a fixed number of subshots depending on the original shot length. A clustering process is then carried out computing each pair of shot distances as a combination of color histogram difference and motion characteristics. The clustering process allows to identify three more characteristics of the summarization process: it relays on an inter-BU based scoring mechanism (the subshots are compared with other subshots), the computational performance of the system is *non-linear* (as the clustering process complexity is *non-linear* and limits the overall system computational performance), and the classification of the system, attending to the generation delay category, is *off-line* because the clustering algorithm needs the whole video information to proceed. Once clustered, the subshots importance is determined according to an average frame saliency value and visual difference with the previous frame (the difference is calculated based in the same features as in the clustering process). The final summary is created by distributing the output summary specified length among the different created clusters according to their accumulated importance. Within each cluster, the available time is distributed again between subshots depending on their importance value. As the output summary size is defined, it is possible to state that the approach is classified as a bounded-size abstraction system.

Figure A.2 (a) depicts a possible distribution of the summary generation algorithm according to the proposed abstraction architecture (see chapter 3, section 3.3.3). In the first step, all the analysis algorithms applied for the extraction of low level features from the original video are included, using a frame as BU: intra-BU features -color histogram, edge features, frame saliency-, inter-BU features -motion intensity, histogram changes- and audio features -probably applied with an inter-BU approach-. The feature extraction processes are independent from the rest of the summarization process and they can be included in an autonomous stage. All the enumerated features can be extracted in a progressive way (there is no need to have the complete video available) so this analysis stage does not constrain system generation delay. The second stage, scoring, includes all those processes devoted video content ranking. There are two stages ('Shot Division' and 'Subshot Sampling') for internal managing the BUs, transforming them from frames to shots and then subshots without adding additional information suitable for the selection process. The 'Junk Detection' stage can be considered as a tagging process where undesired BUs are marked. The 'Clustering' module, aimed for redundancy elimination, classifies each BU in different clusters (once more this can be understood as

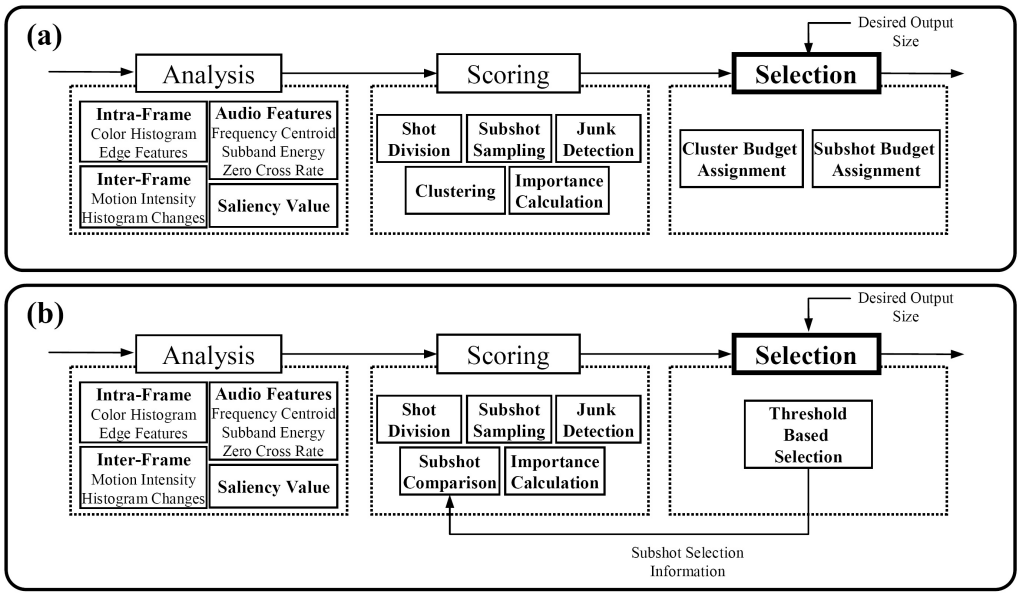


Figure A.2: Abstraction System Modeling

a tagging process). Finally, the 'Importance Calculation' ranks the BUs according to the measure depicted in the algorithm description. At this point it is possible to finalize the process in the 'Selection' stage by picking as many BUs as the desired output size. The previous extracted information (mainly BU division in subshots, cluster tags and subshot importance measure) is taken into account distributing the available output length between clusters/subshots according to the previously depicted criteria.

Given the proposed algorithm separation in stages, it would be quite straightforward to explore different possibilities for the abstraction process in a structured way. For example, the extracted low-level features could be substituted by different ones, keeping the rating and selection stages of the approach and allowing the study of this change effects in the output. In the same way the scoring and selection algorithms could be substituted and benchmarked by their comparison with other equivalent module implementations. If the modules and algorithms were developed following standardized interfaces it could be even possible to build new summarization systems taking modules from different algorithms.

One of the advantages of a clear modularization of an abstraction system and the consideration of the process as a BU flow between different stages is to identify the system elements which limit the system operational features. The studied system is able to perform in a *non-linear*, *off-line* manner. Given the algorithm stage distribution depicted in Figure A.2 (a) it is possible to determine that the limiting module is the clustering process which introduces a non-linear computational complexity element in the process. The selection mechanism requires the complete content annotation, introducing another *off-line* constrained stage in the system. The rest of the processes in the abstraction algorithm, given the available information, can be considered to work in an *on-line* way.

Figure A.2 (b) shows an alternative solution for the studied system in which the clustering process is substituted by an *on-line* approach which compares each incoming BU with previously selected BUs (an information flow from the selection to the scoring stage has been included in order to

share such information). The selection stage is then able to select the BUs to be part of the output abstract by considering the importance value (calculated as in the previous system) and the redundancy scores obtained by the subshot comparison module. A simple thresholding mechanism would be enough. The computational performance of the system will mainly depend on the subshot comparison mechanism. [57] presents a system in which a similar comparison and selection process is carried out in an *on-line, linear* complexity way. The study of the effects in the output summary quality of this kind of modifications in the algorithm operational aspects is out of the scope of this work but the proposed system classifications, architecture decomposition and modeling allow approaching the study of this kind of issues in a systematic and structured way.

# Appendix B

# News Content On-Line Classification

## B.1 Introduction

The first stage in the news abstraction process described in chapter 8 consists in the classification of the incoming video segments in the different possible categories included in a news bulletin. A news bulletin is basically composed by a number of concatenated news stories, each introduced by an anchorperson section and followed by a visual report with the details of the story. However, such basic structure can be refined and a news bulletin may contain many other types of shots, such as reporters, interviews, commercials, etc., can be found. By observing the available content (see chapter 8 for details), the following types of shots have been identified: *Anchorperson, Animation, Black(Frame), Commercial, Communication, Interview, Map, Report, Reporter, Studio, Synthetic* and *Weather*. The fragment classification process, described in the following sections, relies in the fast extraction of low-level features from the original content which are feed to a SVM classifiers structure in charge of labeling each incoming video fragment in one of the existing categories.

## B.2 Feature Extraction

The feature extraction process, part of the *on-line* news abstraction system segment classification described in chapter 8, starts with the calculation of the MPEG-7 Color Layout descriptor [144] for each decoded video frame. This descriptor is particularly suitable for the system purposes as it has been designed as a fast solution for high-speed image retrieval. After its calculation, frames are grouped in blocks of a maximum of 30 consecutive frames, depending on a simple threshold-based shot change detection mechanism. Such mechanism has been previously experimented in [56] and it is implemented by calculating the color layout distance between consecutive frames and splitting the video segments when the difference exceeds an experimentally set threshold. This mechanism provides only a slight improvement with respect to the fixed block separation, avoiding the mix of different shots in a single video segment. Nevertheless, the performance of the overall abstraction algorithm does not have a high dependency on this mechanism because the small video segment size minimizes the possible impact of shot change location errors.

For the classification of each video segment in one of the defined categories, additional features must be extracted segment by segment. Such extraction must be efficient enough so the classification process, together with the execution of the rest of the abstract generation modules, can be completed *on-line*. For the reduction of the required computational complexity, the features are not extracted in

Figure B.1: Anchorperson Face Position Examples

a frame by frame basis but subsampled and averaged for each video segment. It is assumed that, given the small length of the video segments and the subsampling rate, only small variations may occur in the reduced time intervals between the feature extraction instants. The set of extracted features has been selected trying to maximize their meaningfulness (given the different shot categories a news bulletin can contain) while keeping low extraction complexity. The obtained classification results, shown later, demonstrate the feasibility of applying the following 'light' descriptors for the category classification:

- Face detection: The OpenCV library [1] provides a very fast method for arbitrary object detection based on Haar features [173, 157]. In our case a frontal face detection model, particularly suitable for the anchorperson detection, always staring at the camera and with the face located in particular positions (see Figure B.1), has been applied. The average number, size and coordinates of detected faces as well as the variance of such features are calculated for each independent video segment. These features are aimed to allow the differentiation between *Anchorperson*, *Reporter*, *Interview* and the rest of possible categories (see chapter 8 section 8.4 for categories definition).

- Color Variety: The color distribution varies between natural and synthetically generated images and represents a good feature for their differentiation. To measure the number of representative colors in an image, the Y, U and V channels histograms are calculated. For each of them, a color representativeness threshold is experimentally defined as 1/3 of the maximum histogram value. For each video segment we obtain a single color variety value by averaging the number of colors in the histograms with a value over the defined threshold. Figure B.2 shows an example of the calculation of the histograms and representative colors (colors over threshold) for both a synthetic and a natural image.

- Frame Differences: As part of the Color Layout Descriptor extraction, an 8x8 thumbnail image is generated for each decoded frame. For an estimation of the video activity, the average variation for each video segment is calculated by subtracting consecutive frames thumbnails. In order to differentiate between different activity types, for example local or global motion patterns, five different activity areas, shown in Figure B.3 (A), have been defined.

- Shot Variation: In order to obtain an average segment variation measure, the Color Layout difference is calculated every three frames within the video segment and averaged. This provides a different activity measure to that obtained with thumbnail subtraction.

[1] http://sourceforge.net/projects/opencvlibrary/

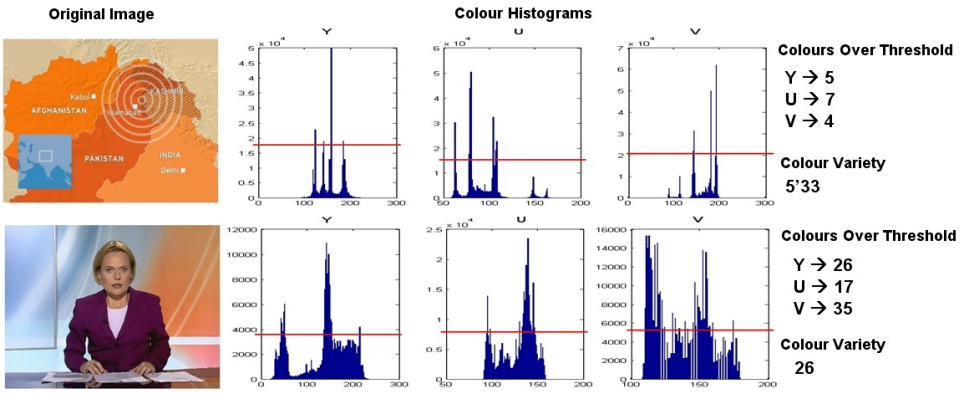


Figure B.2: Representative Color Calculation

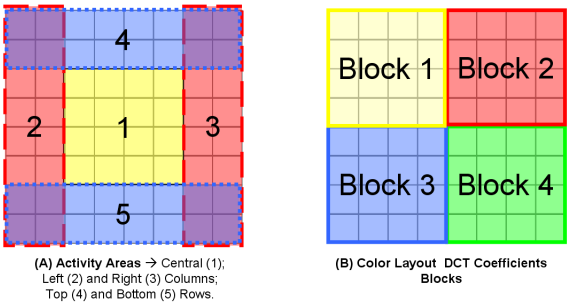


Figure B.3: (A) Frame Block Variation Areas; (B) DCT Coefficients Blocks

- DCT Coefficients Energy: The Color Layout descriptor consists in the DCT coefficients of each color plane 8x8 thumbnail. Making use of those pre-calculated coefficients, it is possible to characterize images with smooth or abrupt changes. Images with different variation characteristics contain different energy distribution within the DCT coefficients. For example, high variation images contain more energy in the DCT coefficients located in the lower right corner of the Color Layout descriptor. In this case the descriptor coefficients have been divided in four areas (see FigureB.3 (B)), which are added up and averaged within each video segment to obtain four frequency measures.

- Image Intensity: Shots recorded in a TV set usually have constant and controlled illumination conditions. In this case the mean and variance of the intensity of each frame are calculated and averaged for each video segment.

The set of extracted features has been selected trying to keep both simplicity and discrimination capacity. Several of the extracted features make use of the Color Layout descriptor associated information, which is later used for shot comparison in the video skimming process, avoiding the need of extracting new features which could slow down the process.

All the extracted features are based on visual information only. Several experiments where carried out with the inclusion of simple audio features (audio energy, zero-cross rate, etc.) which did not produce significant improvement in the classification process. This might be produced because

| Feature | Avg. Extraction Time per Second. | Extraction Frec. (every x frames) |
|---|---|---|
| Frame Decoding | 120.3 ms. | 1 |
| Color Layout | 12.4 ms | 1 |
| Face Detection | 75.6 ms | 7 |
| Color Variety | 1.5 ms | 4 |
| Frame Diffs. | 8.7 ms | 4 |
| Shot Variation | 0.6 ms | 4 |
| DCT Coeffs. Blocks | 0.0045 ms | 4 |
| Image Intensity | 0.018 ms | 4 |
| Total | 219.1 ms | — |

Table B.1: Feature Extraction Average Time per Second of Video

in most of the news content the audio track contains only narrations; and ambient sound or music in a minority of the shots which are already characterized by visual descriptors only (for example in the case of the news bulletin introductory animations with music). On the other hand, the most relevant categories such as anchorperson or reports are separable taking into consideration visual-only features. Previous works devoted to audiovisual scene change detection [181] found out that news is one of the genres in which the audio features are least effective. Moreover, in this case, the performance constraints and lack of available information related to the *on-line* operation complicates the inclusion of more sophisticated (e.g. speech recognition, prosodic analysis, speaker change) and potentially effective audio analysis techniques.

Table B.1 summarizes the extraction times[2] for each of the described features, together with the decoding time. Values are averaged so the reported time represents the average feature calculation time for every second of incoming video. The extraction frequency is included in the table as well. It must be noted that, for the achievement of *on-line* performance, the average decoding, feature extraction, classification, selection and coding steps required for each incoming video segment must be smaller than its playing time. In this case, the average feature extraction time per second, including the frame decoding, is 219.1 milliseconds, providing 780.9 milliseconds per second of video still available for the rest of the abstraction processes.

## B.3 Video Segment Classification

Once all the features have been extracted, each incoming block of frames must be classified in one of the categories defined in chapter 8 section 8.4 . The chosen classifier is the broadly used SVMs -Support Vector Machines- [182] which has proven to provide a good performance in different classification problems [183]. The *libSVM* library [184], integrated in the system, provides a fast and easy to use SVM implementation.

For the training process, 10 complete Deutsche Welle (DW) news bulletins have been manually annotated classifying each shot according to the defined categories. In order to feed the classifier training process with a set of features extracted in the same way as in the abstraction process, the annotated videos are split and features are extracted following the process described in section B.2.

[2] Hardware platform: Intel Core 2 Duo @2.53GHz with 4GB of RAM.

| Category | # Segments | $log(C)$ | $log(gamma)$ | Correct Classification (%) |
|---|---|---|---|---|
| *Anchorperson* | 3124 | 3 | -1.025 | 98.6 |
| *Animation* | 236 | 10.25 | -3.87 | 99.6 |
| *Black* | 131 | -1.37 | 5.75 | 100 |
| *Commercial* | 293 | 24.92 | -18.55 | 84.6 |
| *Communication* | 128 | 6.55 | -7.87 | 99.5 |
| *Interview* | 1466 | -0.125 | -0.9 | 82.8 |
| *Map* | 286 | 22.5 | -17.75 | 98.8 |
| *Report* | 6212 | 4.5 | -3 | 94.5 |
| *Reporter* | 821 | 21.3 | -20.98 | 88.3 |
| *Studio* | 387 | 11.25 | -4.35 | 97 |
| *Synthetic* | 317 | -1 | 0.2 | 100 |
| *Weather* | 454 | 0.925 | -0.2 | 100 |

Table B.2: DW Single Category Classification Results

This process results in a total of 13855 annotated segments available for the training and validation process. Table B.2 summarizes the category distribution of those segments.

An independent binary SVM with RBF -Radial Basis Function- kernel classifier has been trained for each category with a grid search of C and gamma parameters of the SVM. The numbers of positive and negative samples have been equalized for each training process. For each possible C and gamma parameter combination a 5-fold cross validation is carried out with 90% of the training set. The obtained classifier is used for the classification of the 10% remaining test samples for validation purposes. Table B.2 summarizes the obtained C and gamma parameters and correct classification rate for each each category classifier for the 10% validation samples.

It can be observed how the synthetically generated categories, *Black*, *Weather*, *Synthetic, Animation* and *Communication* are the ones with better classification rates due to the low variability in the specific characteristics of this kind of content. The *Map* category classifier performs slightly under the other synthetic categories, probably because the variability in the maps is higher and can eventually contain animations. The *Anchorperson* classifier has a very high classification performance as well, given the well defined characteristics (face presence and location, illumination conditions) of this kind of shots in the DW content. The *Reporter* and *Interview* are two of the categories with lower classification performance because, in many cases, the classifiers are not able to differentiate between them or, under specific circumstances, can consider a reporter or interviewed as an anchorperson. The *Commercial* presents low performance as well, an expectable result because commercials contain very different kind of shots easily mistakable with any other categories. For the proposed abstraction process the good results obtained with the *Anchorperson* classification are very important: the correct identification of the anchorperson shots is of the highest relevance for the correct news segmentation, extraction of relevant news stories introduction and correct overlapping with news reports.

The individual SVM classifiers provide a very good starting point for the classification of the different kind of shots in the news bulletin. Nevertheless, the final decision about which category a shot belongs to is not straightforward: the classification of a shot with the complete set of trained binary classifiers produces, in many cases, a multiple positive situation, that is, the shot is simultaneously

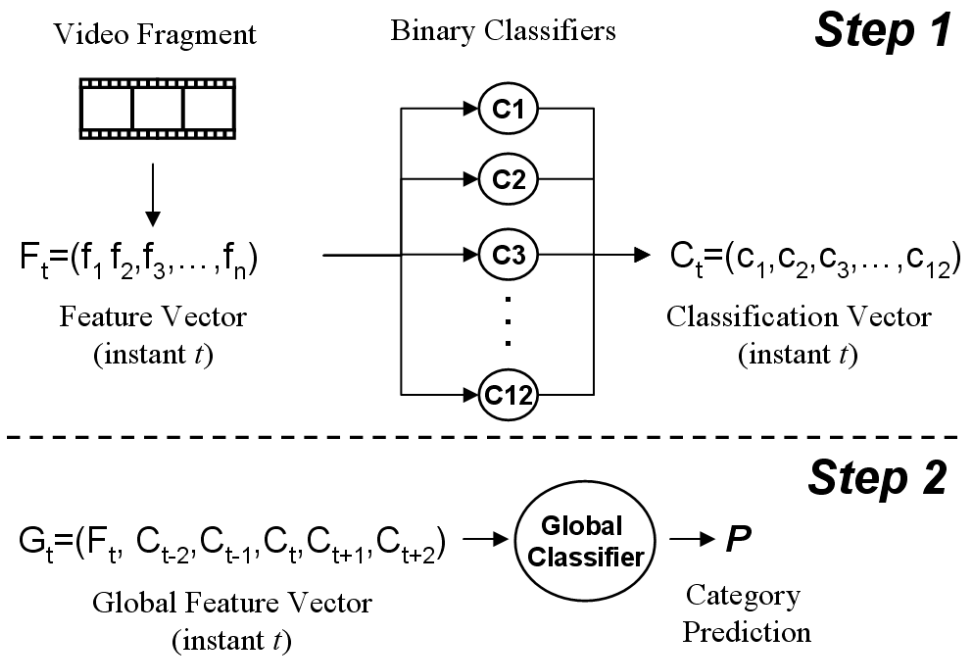


Figure B.4: Global Classification Steps

considered to belong to more than one category.

Another point to take into consideration is the consistency in the category of consecutive video segments. The proposed approach works at sub-shot level and, therefore, it is very likely to find consecutive video segments belonging to the same category.

Both situations have been solved by the training of an additional 'global' SVM which is fed with the individual classifiers predictions in a five segments window (including a temporal dimension in the classification data) and outputs a video segment category prediction in one of the 12 possible categories. Figure B.4 depicts the two steps in the classification process: in the first step a *Classification Vector* for a given time instant is composed by the 12 individual classifications of the video segment by the 12 binary SVMs. In the second step a *Global Feature Vector* for a given time instant is composed by its corresponding extracted features (enumerated in section B.2) and the *Classification Vector* of the two previous and two subsequent segments as well as its own *Classification Vector*. The original set of extracted features are included again in the *Global Feature Vector* because they can provide useful information, not taken into account in the binary classifiers, for the discrimination between two or more specific categories. For example the *anchorperson* single category classifier is trained for the discrimination between *anchorperson* and any other type of video shots (*reporter, animation, maps,* etc.) and relies in those features which provide the best overall classification performance. Considering, for example, the situation of having the *anchorperson* and *reporter* binary classifiers activated, the inclusion of the low level features in the *Global Feature Vector* would allow the global classifier to 'reconsider', for that particular case, the features which better discriminate between *anchorperson* and *reporter,* and that could be different if other binary classifiers are activated. The experiments during the classifiers training showed that the overall classification rate improved if the original set of extracted features were included in the *Global Feature Vector.*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ANCHORPERSON 1 | 99.75 | 0 | 0 | 0 | 0 | 0.25 | 0 | 0 | 0 | 0 | 0 | 0 |
| ANIMATION 2 | 0 | 91.5 | 0 | 0 | 0 | 0 | 0 | 8.5 | 0 | 0 | 0 | 0 |
| BLACK 3 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| COMMERCIAL 4 | 0 | 33.25 | 0 | 58 | 0 | 0 | 0 | 8.75 | 0 | 0 | 0 | 0 |
| COMMUNICATION 5 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| INTERVIEW 6 | 0.75 | 0 | 0 | 0 | 0 | 98.75 | 0 | 0.5 | 0 | 0 | 0 | 0 |
| MAP 7 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| REPORT 8 | 0 | 0 | 0 | 7.25 | 0 | 1.25 | 1.25 | 89.25 | 0 | 1 | 0 | 0 |
| REPORTER 9 | 1 | 0 | 0 | 0 | 0 | 22.75 | 0 | 0 | 76.25 | 0 | 0 | 0 |
| STUDIO 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| SYNTHETIC 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| WEATHER 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

Table B.3: DW Global Classifier Confusion Matrix

| Classification Step | Classification Time |
|---|---|
| Binary Classifiers | 1.111 ms |
| Global Classifier | 1.114 ms |
| Total | 2.226 ms |

Table B.4: Average Classification Time per Second of Video

The global classifier has been trained with the same corpus an methodology as the individual classifiers, with optimal $log(C) = 2.45$ and $log(gamma) = -58.9$ parameters obtaining an overall 92.79% correct classification rate. Table B.3 shows the confusion matrix values for the global classifier when applied to the 10% validation samples. The behavior of the classifier is similar to the individual classifiers performance: the best results are obtained for the *Anchorperson*, *Map, Studio*, *Synthetic* and *Black* categories. The higher incorrect classification rates have been obtained between the *Reporter* and *Interview* categories and between the the *Commercial* and *Report* ones for the previously exposed reasons.

Finally, in order to reduce possible classification mistakes, a temporal filtering of outliers is carried out considering that video segment categories can not present many consecutive changes. For this reason a video segment classified in a category *a* surrounded by two segments of a different category *b* is assigned to belong to category *b*. For the same reason, a segment classified in a category *a*, preceded by a segment belonging to category *b* and followed by segment with a different category *c* is classified as *b* or *c* depending on which of the adjacent segments is more similar from a visual point of view (the comparison is carried out making use of the Color Layout descriptor, with the same mechanism as the one applied for the video skimming process described in chapter 5, section 5.5).

Table B.4 summarizes the average time[3] per second of classified video consumed by the binary and global classification stages. The total average classification time is about 2 ms which is negligible and shows the high efficiency of the SVMs once trained.

[3]Hardware platform: Intel Core 2 Duo @2.53GHz with 4GB of RAM.

| Category | # Segments | $log(C)$ | $log(gamma)$ | Correct Classification (%) |
|---|---|---|---|---|
| *Anchorperson* | 1138 | 0.85 | -0.8 | 99.2 |
| *Animation* | 74 | -0.65 | -3.35 | 99.9 |
| *Commercial* | 115 | 19.2 | -11.45 | 98.1 |
| *Communication* | 122 | -1.7 | -1.5 | 99.9 |
| *Interview* | 625 | 4.55 | -9.8 | 89.5 |
| *Report* | 2990 | 7.25 | -3.25 | 93.0 |
| *Reporter* | 78 | 12.2 | -7 | 99.6 |
| *Studio* | 139 | 3.3 | -0.8 | 99.9 |
| *Synthetic* | 28 | -8.85 | 0.4 | 99.9 |

Table B.5: CCTV Single Category Classification Results

## B.4 Alternative Content Classification

For a further validation of the proposed descriptors and segment classification mechanism, the training and testing processes were repeated with a smaller set of alternative news bulletins. In this case the steps followed in section B.3 were repeated making use of news broadcasts from the Chinese channel CCTV available in the TRECVid 2005 content set. In this case ten news bulletins of about 10 minutes each were manually annotated for the training process. The CCTV news bulletins have a similar structure to the DW ones but no *maps*, *weather* or *black* categories were found in the training set. Moreover the content presents a smaller resolution and worse quality than the DW content. On the other hand, the *anchorperson* shots are very stable, without changes of anchorperson nor background during a single news bulletin and this fact should ease the anchorperson classification.

A total of 5309 individual segments resulted from the annotation process and were used for the feature extraction and training processes. Table B.5 summarizes the number of each segment category used, optimal C and gamma parameters and obtained classification rates for 10% validation samples. The obtained individual classification results are, in principle, better that those obtained for the DW news bulletins. Nevertheless the DW results are more reliable for different reasons: the amount of content is about three times higher than the applied CCTV, there is a higher number of different anchorperson, camera and background combinations as well as different kinds of animations and maps in the DW content. Finally, it was found that the commercials segments in the CCTV content are the same in all the bulletins and this fact explains the unusually good results obtained in the CCTV *commercial* category classification. The worst results were obtained in the *interview* category, behavior found in the DW content as well. It was observed that in several of the interviews, people were recorded in a profile position, what tends to produce fails in the frontal face detection. The *reporter* category obtained a surprisingly good result but the small amount of available *reporter* annotated fragments makes this result unreliable. In the same way, the small amount of *synthetic* content does not allow us to assure a so high classification precision.

The training process of the global classifier, following the same steps depicted in the previous section, resulted in $log(C) = -11.5$ and $log(gamma) = -26.75$ optimal parameters. Table B.6 shows the confusion matrix obtained for the global classifier. In this case the categories with higher classification error are the *interview* and *report* ones. Most of the erroneous *interview* segments are misclassified as *anchorperson* which is expectable given both categories common characteristics. The missclassification of *interview* segments as *report* or *commercial* may be produced by failures in the

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| ANCHORPERSON | 1 | **100** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ANIMATION | 2 | 0 | **100** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| COMMERCIAL | 3 | 0 | 0 | **100** | 0 | 0 | 0 | 0 | 0 | 0 |
| COMMUNICATION | 4 | 0 | 0 | 0 | **100** | 0 | 0 | 0 | 0 | 0 |
| INTERVIEW | 5 | **10.67** | 0 | **10** | 0 | **75.33** | **4.0** | 0 | 0 | 0 |
| REPORT | 6 | 0 | 0 | **1.33** | 0 | **4.0** | **92** | **1.33** | **1.33** | 0 |
| REPORTER | 7 | 0 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 |
| STUDIO | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** | 0 |
| SYNTHETIC | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** |

Table B.6: CCTV Global Classifier Confusion Matrix

frontal face detection process. In the opposite case we can find *report* fragments incorrectly classified as *interview* probably because in such segments, although not being interviews, frontal faces appear. It should be expected that the obtained classification results could be improved with a bigger training content set. Nonetheless, apart from those specific issues, the overall results are coherent with the results obtained with the DW bulletins and demonstrates the possible application of the extracted descriptors and classification scheme for their application with different content to the one used during the development of the system. In section C.2.2 additional evaluations of the CCTV content abstraction results are reported.

# Appendix C

# On-Line News Summarization Evaluation

## C.1 Introduction

This appendix presents the results obtained in the evaluation of the *on-line* news abstraction system described in chapter 8. The evaluation has been carried from both objective and subjective points of view.

The objective evaluation (section C.2) consists on the analysis of the correct identification of anchorperson and report sections of the news stories as well as their correct composition. The individual video segment classification performance of the system has been previously evaluated and described in appendix B. Such individual category classification results have a great influence in the overall system performance because the correct news story identification and anchorperson-report alignment depend on them (see chapter 8 section 8.4 for categories definition). The results obtained in the *Anchorperson* classification are very important for the overall performance of the system while the incorrect classification of *Report, Interviews* or *Commercial* segments have a very reduced impact in the output abstract quality.

The subjective evaluation (section C.3) is based on the validation of the system by a set of user tests in which the quality and representativeness of the proposed approach have been measured by the visualization of several of the generated abstracts by different users. The user evaluation includes examples of incorrectly composed news stories for the study of their impact in the users perception.

## C.2 Objective Evaluation

For the evaluation of the system, the inclusion of each news story in the output abstract and the correct synchronization of its anchorperson (in case s/he exists) with the report video skim has been taken into account. For reports without introductory anchorperson, the inclusion of the associated video skim is considered as a correct news story inclusion and it is not computed for the correct anchorperson-report alignment statistics. The inclusion of no relevant content has not been penalized but, in order to have a reference to measure the length reduction performance (targeting to a 1/3 length ratio), the average relation between the anchorperson and the rest of the content in complete news bulletins has been considered as an optimal result (it is the maximum possible reduction if all anchorperson sections are kept).

| Video | Original Length (min.) | Abstract Length (min.) | # Stories (#compound) | # Inclusion | #Alignment | Process Time (min.) |
|---|---|---|---|---|---|---|
| 321070_4_journal_spa | 28 | 9.63 (34%) | 17 (14) | 17/17 (100%) | 12/14 (85.7%) | 7.9 |
| 327904_3_journal_spa | 28 | 8.68 (31%) | 21 (19) | 17/21 (81%) | 13/19 (68.4%) | 7.4 |
| 326156_3_journal_eng_16 | 28 | 7.34 (26%) | 12 (10) | 12/12 (100%) | 9/10 (90.0%) | 7.6 |
| 326156_3_journal_eng_18 | 28 | 7.15 (25%) | 14 (10) | 14/14 (100%) | 8/10 (80%) | 7.9 |
| 326181_3_journal_eng_08 | 28 | 7.93 (28%) | 16 (16) | 14/16 (87.5%) | 13/16 (81%) | 8.0 |
| 327916_3_journal_eng_18 | 28 | 10.43 (37%) | 19 (15) | 18/19 (94.7%) | 13/15 (86%) | 8.2 |
| *Total* | 168 | 51.16 (30.4%) | 99 (84) | 92/99 (92.9%) | 68/84 (80.9%) | 47.0 |

Table C.1: DW Abstraction Results

### C.2.1 DW Content

Table C.1 summarizes the results obtained for 6 complete DW news bulletins (different to those annotated and used for segment classification training). The table presents the original video and obtained abstract lengths, number of news stories in each original video, and how many of them are compound ones, that is, composed of anchorperson and a report and therefore subject to a possible foreground anchorperson/background video skim alignment. The obtained results are presented as the number of stories included in the abstract (when the corresponding anchorperson introduction is included) and the number of correct alignments between anchorperson and report skims when dealing with compound stories.

The obtained results demonstrate the feasibility of obtaining good size reduction (close to the 'optimal' target of 1/3 of the original length) with news video content while retaining most part of the news stories in the news bulletins (92.9%) and with a correct overlapping between the anchorperson and the report sections in 80.9% of the cases. Most of the incorrect news story inclusions are due to the incorrect classification of the anchorperson sections as interview or reporter ones. In the cases where incorrect alignment occurs, it is usually because *Interview* fragments are misclassified as *Anchorperson,* being overlapped over *Report* fragments, or when a *Report* fragment classification error produces an incorrect state change (for example *Report* fragments classified as *Commercial* may produce a premature news story composition finalization). It is expectable to correct those situations and improve the results with the development of more precise classification mechanisms. It should be pointed out that all the introductory and end animations for the news bulletins, as well as the studio shots, where correctly eliminated in all bulletins. A reduced part of the commercial sections included in the news bulletin (those not correctly classified as *Commercial* content) are skimmed and included in the output abstract, but heavily reducing the length of these sections.

The overall process is carried out in an *on-line* processing way and therefore, given the operative constraints, the classification, inclusion and alignment results can be considered as very good. The total processing time[1] is considerable below the original video duration and slightly under the output abstract length, thus demonstrating the feasibility of the proposed approach for continuous broadcasting processing and for *real-time* abstraction (displaying the output as it is being generated) of already stored content.

### C.2.2 CCTV Content

In section B.4 of appendix B, the feature extraction and classification processes were validated with a content set different to the one used during the development of the system (the DW news bulletins). In this section we carry out a simplified validation test of the complete abstraction process trying to determine the whole abstraction system applicability to a different content set just by a retraining of

[1] Hardware platform: Intel Core 2 Duo @2.53GHz with 4GB of RAM.


Figure C.1: CCTV News Abstract Composition Example

| Video | Original Length (min.) | Abstract Length (min.) | #Anchorperson Inclusion | #Reports Inclusion | #Alignment |
|---|---|---|---|---|---|
| 20041101_110000_CCTV4_NEWS3_CHN.mpg | 10:00 | 3:14 | 9/10 | 8/8 | 8/8 |
| 20041108_110000_CCTV4_NEWS3_CHN.mpg | 9:40 | 4:56 | 6/7 | 5/5 | 4/5 |
| 20041109_110100_CCTV4_NEWS3_CHN.mpg | 9:00 | 3:02 | 6/7 | 4/4 | 4/4 |
| *Total* | 28:40 | 11:12 (39%) | 21/24 (87.5%) | 17/17 (100%) | 16/17 (94.12%) |

Table C.2: CCTV Anchorperson-Report Inclusion Results

the segment classifiers.

The anchorperson overlapping window proportions were slightly modified for a better visualization given the smaller resolution of the CCTV bulletins (see figure C.1). With respect to the overall bulletin structure, the news bulletins are shorter (about ten minutes) than the DW ones (28 minutes) but the anchorperson-report news stories structure is kept along the news bulletin. A commercial section is included within the news bulletin but, as commented in previous sections, it is the same in all the news bulletins used and, therefore, it is always correctly detected and eliminated. In this case, exact alignment between the anchorperson speech and the background video skim was not evaluated[2] and therefore only the correct inclusion of all the anchorperson sections and the following reports (if applicable) were evaluated. Table C.2 summarizes the obtained results for 3 CCTV news videos depicting the original and abstract lengths, the number of existing and included anchorperson, and the number of correct alignments (anchorperson overlapped over the following news report).

The anchorperson inclusion errors are produced in all cases because the news bulletins include short anchorperson fragments under the established 5 seconds minimal length (see chapter 8 table 8.1). In two cases, this situation was produced at the finalization of the news bulletin and, therefore, no relevant information was missed. Only one of the cases was produced in the middle of a news story. All the reports were appropriately skimmed and included in the abstracts with a correct alignment in 16 out of 17 cases (the only exception is the no detection of the anchorperson within a news story). The obtained abstract length is close to the 'optimal' 1/3 value (it is substantially higher in one of the bulletins due to a long anchorperson appearance at the end of the video without being followed by a news report).

The obtained results are quite good in general terms, even considering the relatively small amount of data utilized in the training of the classifiers (see appendix B), and demonstrate the applicability of the proposed approach for different news broadcast content.

[2] Due to the lack of knowledge about the chinese language.

| Segment | Test | Original Video | #Stories | Original Length (sec.) | Abstract Length (sec.) | Output Length Ratio | Language | Correct Alignment |
|---|---|---|---|---|---|---|---|---|
| S1 | 1 | 327904_3_journal_spa | 2 | 94 | 37 | 0.39 | SPA | No |
| S2 | 1 | 327904_3_journal_spa | 2 | 123 | 92 | 0.75 | SPA | Yes |
| S3 | 1 | 327904_3_journal_spa | 2 | 113 | 50 | 0.44 | SPA | No |
| S4 | 1 | 327904_3_journal_spa | 1 | 83 | 27 | 0.33 | SPA | Yes |
| S5 | 2 | 327904_3_journal_spa | 1 | 93 | 49 | 0.53 | SPA | Yes |
| S6 | 2 | 327904_3_journal_spa | 2 | 111 | 37 | 0.33 | SPA | Yes |
| S7 | 2 | 327904_3_journal_spa | 2 | 159 | 49 | 0.31 | SPA | Yes |
| S8 | 2 | 327904_3_journal_spa | 2 | 129 | 64 | 0.5 | SPA | No |
| S9 | 3 | 327916_3_journal_eng_18 | 2 | 121 | 43 | 0.36 | ENG | Yes |
| S10 | 3 | 327916_3_journal_eng_18 | 1 | 38 | 13 | 0.34 | ENG | Yes |
| S11 | 3 | 327916_3_journal_eng_18 | 2 | 144 | 45 | 0.31 | ENG | Yes |
| S12 | 3 | 327916_3_journal_eng_18 | 1 | 174 | 81 | 0.47 | ENG | Yes |

Table C.3: News Segments for User Evaluation

## C.3 Subjective Evaluation

For the validation of the proposed abstraction approach from an subjective point of view, an user test campaign was carried out. There were three principal aspects in which the tests were focused: the representativeness of the proposed approach, the generated abstracts pleasantness, and the usefulness of the abstracts. The tests were carried out with a total of 27 users. Three different tests were implemented, combining different news bulletin fragments from the DW content set, and each user was asked to visualize and evaluate one of the tests (yielding a total of nine users per test). Each test was composed of four different news bulletin summarized fragments and their corresponding original video. Instead of evaluating complete bulletins, small fragments of one or two news stories were presented to the users. This design decision was taken, because a complete 28 minutes bulletin would have been too long to keep the users attention level and to allow the user remembering the details about all the individual news stories. Nevertheless, some of the evaluated videos are composed by two consecutive stories so that the user can check the individual story abstract concatenation. Table C.3 summarizes the different news story fragments used in each of the three different tests, including the number of stories that each segment contained, the original video duration, the summary length, the news story language and the correct alignment (specifying if the anchorperson introduction was correctly overlapped with the news report or news, or if there were overlapping errors). Most of the segments are in Spanish because most of the evaluators were Spanish native speakers and the correct understanding of the news stories has high relevance.

After the visualization of each pair of abstract/original video, the users were asked to rate their level of agreement with several assertions (Q1-Q4) about each video abstract, and, at the end of the test, they were asked to rate a final set of three general assertions (FQ1-FQ3). Table C.4 shows the different assertions. For each question the user was able to choose between 5 different levels of agreement except for question Q4 (about the length of the abstract), in which the user had to indicate his/her opinion about the length of the abstract. Finally, at the end of the questionnaire, the users were able to make any desired comment about the abstraction method or the questions.

Table C.5 shows the evaluation results (average and standard deviation) per video for questions Q1-Q4 (asked after the visualization of each news bulletin fragment). The obtained results are, in general terms, quite positive for the validation of the proposed approach. Q1 results ('*The summary adequately represents the original bulletin*') obtained very good results with the values for most part

| VIDEO QUESTIONS | |
|---|---|
| Question ID | **Assertion** |
| Q1 | The summary adequately represents the original bulletin... |
| Q2 | The summary rhythm and composition are pleasant... |
| Q3 | There is relevant/fundamental information missing in the summary... |
| Q4 | The summary length is... |
| **GENERAL QUESTIONS** | |
| Question ID | **Assertion** |
| GQ1 | The proposed summarization technique is useful for news video content... |
| GQ2 | The displayed summaries are pleasant to see... |
| GQ3 | The summaries provides a proper understanding about the original news bulletin video... |
| **POSSIBLE ANSWERS** | |
| Applied for | **Choices** |
| Q1-Q3; GQ1-GQ3 | 1 - Strongly Disagree, 2- Disagree, 3 - No Opinion, 4 - Agree, 5 - Strongly Agree |
| Q4 | 1 - Too Short, 2 - Short, 3 -Adequate, 4 - Long, 5 - Too Long |

Table C.4: Test Questions

of the videos close or above the 4 ('Agree') and an average value of 4.22. Q2 results ('*The summary rhythm and composition are pleasant*') present more variations. The average value, 3.76, is close to the agreement value and therefore users tend to think that the summaries are pleasant. Nevertheless some of the abstract obtained values closer to a neutral opinion (S3 and S7, with scores 3.22 and 3 respectively) or even to the disagreement score (S6, score 2.44). In the case of S3 there is a composition mistake in the video abstract which clearly affects the user perception. In the news segment S7 the main problem may be related to the anchorperson introductory narration which does not finish before the visual report starts and, therefore, it is incomplete in the output abstract. The S6 abstract presents the case of short visual reports fragments after the anchorperson finishes which may produce an unpleasant rhythm. Several of the users commented that the news report video skim, in the cases in which it was longer than the anchorperson introduction, presented audio cuts which had a negative influence in the abstracts pleasantness. Such issue is a typical problem in many video skimming approaches and should be addresses in the future to enhance the abstracts pleasantness (for example, selecting a continuous audio fragment from the report instead of the audio of each fragment). Nonetheless, the average results are good and the representativeness of the abstracts is still high even in abstracts with lower pleasantness score. The third question ('*There is relevant/fundamental information missing in the summary*') was aimed to determine if there was really important information missed in the abstract and complements question Q1. The average obtained score was 2.42, between the 'No Opinion' and 'Disagree' values, showing a slight tendency by the users to consider that there is not really fundamental information lost in the abstracts. Of course, the complete news stories provide more information about the story than just the single anchorperson introduction, but, taking

| SEGMENT | Q1 \| Average : Deviation | Q2 \| Average : Deviation | Q3 \| Average : Deviation | Q4 \| Average : Deviation |
|---|---|---|---|---|
| **S1** | 4.22 : 1.09 | 4.44 : 0.53 | 2.44 : 1.01 | 3.00 : 0.71 |
| **S2** | 4.33 : 0.50 | 3.77 : 1.20 | 1.78 : 0.97 | 4.00 : 1.00 |
| **S3** | 3.88 : 0.93 | 3.22 : 1.30 | 2.11 : 1.17 | 3.11 : 0.78 |
| **S4** | 4.66 : 0.50 | 4.66 : 0.50 | 1.89 : 0.93 | 3.22 : 0.44 |
| **S5** | 4.44 : 0.53 | 4.22 : 0.44 | 2.67 : 0.87 | 3.56 : 0.53 |
| **S6** | 3.55 : 1.01 | 2.44 : 1.13 | 3.00 : 1.00 | 2.89 : 0.33 |
| **S7** | 4.44 : 0.52 | 3.00 : 1.22 | 2.67 : 1.12 | 3.11 : 0.33 |
| **S8** | 4.11 : 1.17 | 3.67 : 1.12 | 2.67 : 0.87 | 3.33 : 0.50 |
| **S9** | 4.44 : 0.53 | 4.00 : 1.00 | 2.11 : 0.33 | 3.11 : 0.33 |
| **S10** | 4.11 : 0.78 | 4.67 : 0.50 | 3.11 : 1.45 | 3.00 : 0.50 |
| **S11** | 3.89 : 0.93 | 3.44 : 1.01 | 2.67 : 0.87 | 2.78 : 0.44 |
| **S12** | 4.55 : 0.53 | 3.66 : 1.12 | 1.89 : 1.05 | 3.33 : 0.70 |
| **Average** | 4.22 : 0.75 | 3.76 : 0.92 | 2.42 : 0.97 | 3.20 : 0.55 |

Table C.5: Evaluation Results per Video Segment

into consideration the combination of Q1 and Q3, it can be stated that the abstracts adequately represent the original video information. Several users pointed out that there was a high dependency on how well the anchorperson introduction described the rest of the news with this possible lack of information. For example, segment S10 obtained one of the worst results for Q3, 3.11, (which is, however. a neutral result) while the scores for Q1,Q2 and Q4 (later analyzed) were quite good. These results can only be explained by the specific content of such news story and the information they contain.

The last question presented to the user, Q4 ('*The summary length is...*'), was aimed to determine if there was any users' preference about the original and abstract lengths ratio. The average score obtained was 3.22, which is very close to the '*Adequate*' length choice in the test. Therefore, in general terms, the length of the generated abstracts seem to be correct. The abstracts lengths ratios are depicted in table C.3 and a high correlation between such values and the obtained Q4 score can be observed. Segments S2, S5, S8 and S11, with abstract length ratios of 0.75, 0.53, 0.50 and 0.47, obtained Q4 scores of, respectively, 4.00, 3.55, 3.33 and 3.33, showing a long abstract perception by the users. The abstracts with best Q4 scores, presenting an adequate length for the users, are those with a length ratio of about 1/3.

Figure C.2 depicts the answer frequencies for questions Q1-Q4 for the whole set of segments included in the 3 different tests carried out. Summarizing the results, in 87% of the cases, the users agreed or strongly agreed with '*The summary adequately represents the original bulletin*'. In 65.7% of the cases, the users disagreed or strongly disagreed with '*There is relevant/fundamental information missing in the summary*' against a 23.14% of the cases where the users considered relevant information was missed (agreed or strongly agreed with Q3). '*The summary rhythm and composition are pleasant*' for the users in a 71.29% of the cases while in the 20.3% of them, users disagreed or strongly disagreed with such assertion. Finally, users considered the summary lengths as adequate in 66.6% of the cases, somehow short or long in 29'6% of them, and too long or short in only 3.7% of the displayed abstracts.

After watching and rating each individual abstract, the users were asked to specify their level of agreement with three general questions (GQ1-GQ3, see table C.4). In this case, the users had to consider the whole set of displayed abstracts and original videos in order to provide an overall impression about the proposed abstraction approach. Figure C.3 presents the results obtained for the 27 users which carried out the tests: 96.3% of the users agreed or strongly agreed in FQ1 "*The proposed summarization technique is useful for news video content*", 81.6% of the users considered that (FQ2) "*The displayed summaries are pleasant to see*" and, finally, for question FQ3, "*The summaries provide*

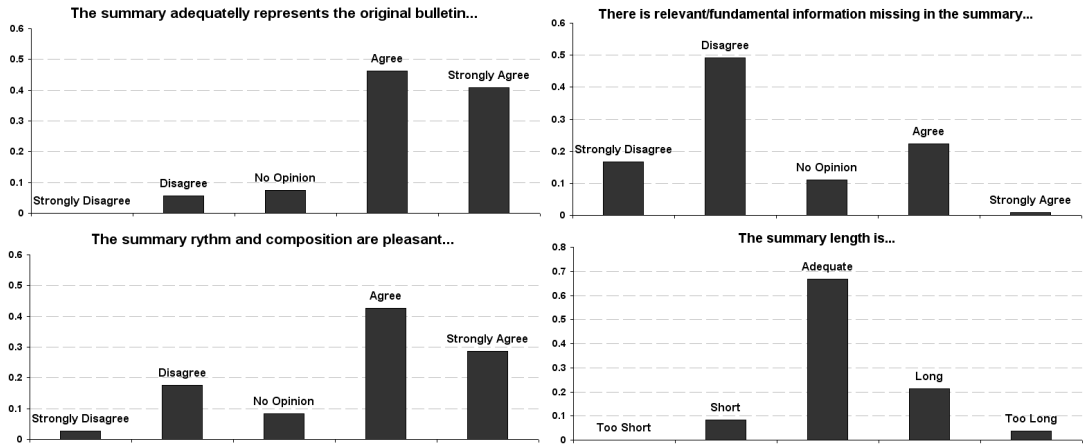


Figure C.2: Q1-Q4 Answer Frequencies

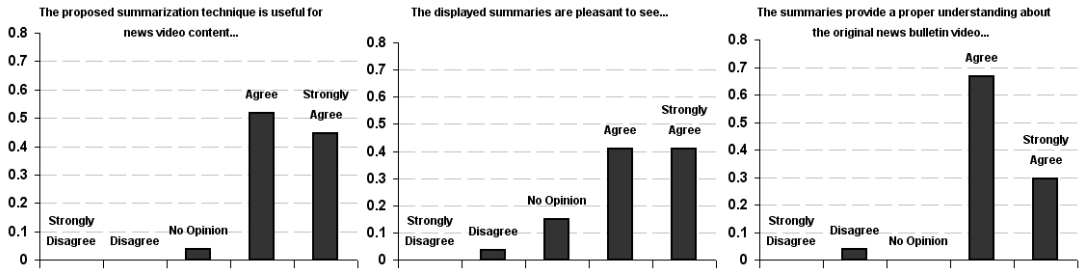


Figure C.3: FQ1-FQ3 Answer Frequencies

*a proper understanding about the original news bulletin video*", 96% of the users agreed or strongly agreed.

The obtained results validate the proposed news video abstraction approach in both the individual evaluation of the abstracts and the general questions about the usefulness and quality of the abstraction approach. The main purpose of the abstracts, to provide a representative short version of the original news content, is, according to the obtained results, successfully achieved. It seems that the pleasantness of the abstracts visualization, although validated by the users opinion, could be improved if, as several users commented, the cuts in the audio track could be correct in the cases where the report video skim is longer than the anchorperson introduction. In general terms, the obtained user evaluation results are very good, specially considering the speed constraints of the processing and progressive output generation (the latter limiting the amount of available information for the abstract generation).

# Appendix D

# Publications

List of published papers grouped by related topic and associated thesis chapter:

- Abstraction systems taxonomy and video abstraction architecture (chapter 3).
  - V. Valdés, J. M. Martínez, "A framework for video abstraction systems analysis and modelling from an operational point of view", Multimedia Tool and Applications (Online Published, 10 October 2009, DOI: 10.1007/s11042-009-0392-7 , ISSN: 1573-7721).
  - V. Valdés, J. M. Martínez, "On Video Abstraction Systems Architectures and Modelling", Proceedings of the 3rd International Conference on Semantics and Digital Media Technologies, Lecture Notes in Computer Science, Vol. 5392, Springer Verlag, 2008, pp. 164-177.
- Novel algorithms for *on-line* and *real-time* video skimming (chapter 5).
  - V. Valdés, J. M. Martínez, "Post-Processing Techniques for On-line Adaptive Video Summarization Based on Relevance Curves", Proceedings of the 2nd International Conference on Semantics and Digital Media Technologies, Lecture Notes in Computer Science, Vol. 4816, Springer Verlag, 2007, pp.144-157.
  - V. Valdés, J. M. Martínez, "On-line Video Skimming Based on Histogram Similarity", Proceedings of the ACM Multimedia 2007 Workshop on TRECVID Video Summarization, Augsburg, Germany, 24-29 September 2007, pp. 94-98.
  - V. Valdés, J. M. Martínez, "On-line video summarization based on signature-based junk and redundancy filtering", Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'2008, Klagenfurt, Austria, 7-9 May 2008. pp. 88-91.
  - V. Valdés, J. M. Martínez "Binary Tree Based On-Line Video Summarization", Proceedings of the ACM Multimedia 2008 (TRECVID Video Summarization Workshop), Vancouver, Canada, 27 October – 1 November 2008, pp. 134-138.
- Application for the generation of broadcasted news abstracts integrating *on-line* techniques for segment classification video skimming and abstract presentation composition (chapter 8).

– Á. García, J. Molina, F. López, V. Valdés, F. Tiburzi, J. M. Martínez, J. Bescós, “Instant Customized Summaries Streaming: a service for immediate awareness of new video content”, 7th Workshop on Adaptive Multimedia Retrieval, Lecture Notes on Computer Science, Springer Verlang, Madrid, 2009.

# Appendix E

# Conclusiones

## E.1 Contribuciones Principales

En esta tesis se ha llevado a cabo un estudio completo sobre técnicas de generación de resúmenes *on-line* (en vivo) y *real-time* (en tiempo real). El trabajo, en su conjunto, analiza los diferentes aspectos del problema: estudio de las técnicas existentes y antecedentes; definición de un marco de análisis; definición y estudio previos del problema abordado; propuesta de soluciones en forma de algoritmos novedosos; análisis y evaluación de las soluciones propuestas; y, finalmente, desarrollo de aplicaciones reales basadas en las soluciones y algoritmos propuestos. Las principales contribuciones de este trabajo son las siguientes:

- Una taxonomía y arquitectura genérica para sistemas de generación de resúmenes (capítulo 3).
- Definición y análisis de los conceptos de generación de resúmenes de vídeo *on-line* y *real-time* así como de las implicaciones y restricciones asociadas al desarrollo de dicho tipo de sistemas (capítulo 4).
- Propuesta de nuevos algoritmos para generación *on-line* y *real-time* de resúmenes (descripción de los algoritmos en el capítulo 5, evaluación de los mismos en el capítulo 7).
- Desarrollo de un sistema novedoso para la evaluación automática de sistemas de generación de resúmenes de vídeo (capítulo 6).
- Desarrollo de una aplicación para la generación de resúmenes de telediarios en tiempo de emisión, integrando técnicas *on-line* para clasificación de segmentos, generación de resúmenes y vídeo-composición de la presentación de los mismos (capítulo 8).
- Desarrollo de una aplicación novedosa, RISPlayer, para la generación y visualización en tiempo real de resúmenes de vídeo interactivos (capítulo 9).

Este trabajo comienza presentando una visión general sobre técnicas de generación de resumenes y clasificaciones propuestas en la literatura (capítulo 2). Basándonos en el estudio de las soluciones existentes, se propone un marco de estudio unificado (capítulo 3), constituido por una taxonomía de métodos de generación de resumenes y por una arquitectura genérica que permite el modelado de la gran mayoría de aproximaciones existentes. La taxonomía propuesta clasifica los métodos de generación de resúmenes de acuerdo con sus características externas e internas y representa un punto

de vista novedoso muy apropiado para el análisis de las características operativas de los sistemas de generación de resúmenes. Por otra parte, la arquitectura propuesta muestra como caracterizar prácticamente cualquier sistema de generación de resúmenes mediante la división del proceso en tres etapas básicas ('análisis', 'puntuación' y 'selección') y considerándolo como un flujo de 'unidades básicas' (fragmentos extraidos del vídeo original) a través de las diferentes etapas.

El marco establecido define conceptos y términos aplicados para el análisis de las aproximaciones existentes actualmente y la propuesta de nuevas soluciones. Se han definido las modalidades de generación de resúmenes *on-line* y *real-time*, estableciendo las limitaciones computacionales asociadas a cada una de ellas (capítulo 4). Además, se ha llevado a cabo un análisis de las potenciales implicaciones de ambas modalidades en los algoritmos subyacentes a las técnicas de generación de resúmenes.

Considerando las limitaciones e implicaciones estudiadas anteriormente, se han propuesto dos nuevos algoritmos para la generación de resúmenes de vídeo *on-line* (capítulo 5). Ambos algoritmos se han desarrollado con el objetivo de ser aplicados sobre contenido genérico y, por tanto, los dos están basados en técnicas de eliminación de redundancia visual, aunque otros aspectos con respecto a los resúmenes generados han sido tenidos en cuenta (ritmo, continuidad y calidad visual en los resumenes). Los dos algoritmos difieren en su complejidad, con una primera aproximación basada en un mecanismo 'variación suficiente del contenido' ('sufficient content change'), que proporciona una menor cantidad de posibilidades de configuración, y una segunda propuesta, basada en árboles binarios, que es muy flexible en términos de posibles configuraciones para la generación de resúmenes. Esta última aproximación proporciona un sistema genérico *on-line* con mecanismos integrados para personalizar el tipo de resúmens generados, filtrado de contenido y escalabilidad en términos de complejidad computacional, retardo en la generación de los resúmenes y calidad de los mismos. Dos sistemas para la generación de resúmenes de 'rushes' (grabaciones sin editar), basados en los algoritmos propuestos, se enviaron a las campañas de evaluación TRECVid BBC Rushes Summarization Task de los años 2007 y 2008, obteniendo resultados comparables con los del resto de participantes (compuestos basicamente por aproximaciones *off-line*).

No obstante, para una validación completa de los algoritmos propuestos, ha sido necesario llevar a cabo una batería de pruebas más exhaustiva. Por esta razón, haciendo uso de los resúmenes y resultados de todos los participantes en la campaña de evaluación TRECVid BBC Rushes Summarization Task de 2008, se ha desarrollado de un sistema de evaluación automático basado en el entrenamiento de predictores individuales para varias de las características evaluadas en dicha campaña. El sistema desarrollado permite aproximar resultados de evaluaciones llevadas a cabo por personas basándose en métricas extraidas de los resúmenes. Haciendo uso de dicho sistema, los algoritmos descritos para generación de resúmenes *on-line* han sido evaluados en profundidad, analizando sus funcionalidades y capacidades comparándolas con las de técnicas *off-line* (capítulo 7).

Se ha llevado a cabo el desarrollo de dos aplicaciones integrando el algoritmo para generación de resúmenes *on-line* basado en árboles binarios y demostrando la aplicabilidad de los algoritmos y conceptos propuestos. La primera aplicación consiste en un sistema para la generación de resúmenes *on-line* de telediarios (capítulo 8) y combina diferentes técnicas para el análisis y clasificación en categorías de los segmentos de vídeo (basada en un proceso de extracción de características y entrenamiento de clasificadores explicado en el apéndice B), generación de resúmenes de dichos segmentos (basada en el algoritmo de árboles binarios descrito en el capítulo 5) y composición de la presentación del resúmen. El sistema, en su conjunto, permite el procesado del vídeo original de forma *on-line*, haciendo posible la aplicación del sistema para la generación de resúmenes de noticias en

tiempo de emisión. Se ha llevado a cabo la evaluación de la calidad de los resúmenes generados desde puntos de vista tanto objetivos como subjetivos, con una validación con usuarios reales en la que se obtuvieron muy buenos resultados (consultar apéndice C para más detalles).

La segunda aplicación consiste en un reproductor de vídeo -RISPlayer- capaz de generar resúmenes de vídeo en tiempo real y de forma interactiva (capítulo 9). Dicha aplicación ha sido desarrollada integrando el algoritmo de generación de resúmenes basado en árboles binarios y proporciona las funcionalidades de un reproductor de vídeo habitual así como las de generación de resúmenes (en cuyo caso solamente los fragmentos del vídeo original seleccionados son reproducidos). El RISPlayer incluye mecanismos para el control automático de rendimiento, en términos de complejidad computacional, basados en las propiedades de escalabilidad del algoritmo de árboles binarios, y permite la modificación por parte del usuario de los diferentes pesos que influyen en las características de los resúmenes generados. La aplicación demuestra el potencial de las técnicas de generación de resúmenes en tiempo real y esboza posibilidades futuras en cuanto a generación de resúmenes personalizados e interactivos.

## E.2 Trabajo Futuro

Mas allá de los objetivos obtenidos hasta el momento, existen varias direcciones posibles para la continuación del trabajo desarrollado en esta tésis, principalmente relacionados con los siguientes aspectos:

- Identificación de nuevos tipos de contenido y aplicaciones potenciales de los algoritmos de generación de resúmenes *on-line*:
  - Los algoritmos genéricos de generación de resúmenes *on-line* presentados en este trabajo están basados en la aplicación de mecanismos de eliminación de redundancia, una aproximación cuya aplicabilidad ha sido demostrada para ciertos tipos de contenido (películas y 'rushes'). El análisis de nuevos tipos de contenido (como deportes o series de televisión) para la validación de los métodos propuestos constituiría una interesante línea de investigación futura.
  - Uno de los algoritmos de generación de resúmenes presentado en este trabajo se centra en el procesamiento de telediarios, combinando algoritmos específicos para la clasificación y composición de contenido de noticias, junto con el algoritmo de generación de resúmenes genéricos basado en árboles binarios. La identificación de nuevos escenarios de aplicación de sistemas *on-line* junto con el desarrollo de técnicas especializadas para el tratamiento de tipos de contenido específicos es una de las posibles lineas de investigación futuras. Algunos ejemplos de posibles escenarios para la aplicación de las técnicas de generación de resúmenes *on-line* serían la generación de resúmenes de distintos tipos de contenido en tiempo de emisión, sistemas de grabación continua (como sistemas de videovigilancia), vídeo en Internet o aplicaciones para terminales de baja capacidad computacional (por ejemplo terminales móviles).

- Mejora de los algoritmos de generación de resúmenes *on-line*:
  - El algoritmo basado en árboles binarios propuesto proporciona un marco de desarrollo con muchas posibilidades para futuras implementaciones de sistemas de generación de

resúmenes *on-line*. No obstante existen aspectos de dicho algoritmo potencialmente mejorables, tales como su eficiencia computacional, el desarrollo de estrategias 'inteligentes' para la selección de caminos en el árbol o la experimentación con nuevas fórmulas de 'puntuación' de los resúmenes generados y nuevos tipos de contenido.

- Generación de resúmenes en tiempo real:

  - Uno de los aspectos más interesantes del trabajo presentado es la posibilidad de generación de resúmenes en tiempo real. Este tipo de aproximaciones proporcionan las funcionalidades necesarias para permitir al usuario interactuar con el proceso de generación del resumen, visualizando los resultados de sus cambios sobre la marcha y permitiendo la posibilidad de navegación interactiva por el contenido. La aplicación presentada en este trabajo se centra en la validación y experimentación con las funcionalidades que ofrecen dichas modalidades de generación de resúmenes. Sin embargo, futuras mejoras orientadas a incrementar la usabilidad de dichas aplicaciones para usuarios no expertos así como el estudio de posibles mecanismos de interacción con el usuario son aspectos de interés para futuras investigaciones.

- Sistema de evaluación automática de resúmenes:

  - El desarrollo de mecanismos para la evaluación automática de resúmenes tiene una gran cantidad de potenciales aplicaciones en cuanto al desarrollo y mejora de los algoritmos de generación de resúmenes existentes. El trabajo presentado demuestra la posibilidad real de dicha automatización, al menos con el tipo de contenido y las condiciones de evaluación aplicadas en la campaña 'TRECVid 2008 BBC Rushes Summarization Task'. El trabajo futuro en este área se centrará en la mejora de las técnicas de extracción automática de métricas de los resúmenes y los mecanismos de predicción, así como en la validación de las técnicas desarrolladas con nuevos tipos de contenido y medidas de evaluación. La validación de las técnicas desarrolladas podría permitir, en el futuro, una mejor comprensión sobre los aspectos que caracterizan a los buenos resúmenes y en la aplicación de dicho conocimiento para el desarrollo de nuevas técnicas para su generación automática.

# Bibliography


[1] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 3, pp. 1–37, February 2007.

[2] J. Ren and J. Jiang, "Hierarchical modeling and adaptive clustering for real-time summarization of rush videos," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 906–917, 2009.

[3] Y. Gong and X. Liu, "Video summarization and retrieval using singular value decomposition," *Multimedia Systems*, vol. 9, no. 2, pp. 157–168, 2003.

[4] Y. Ma, X. Hua, L. Lu, and H. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 907–919, 2005.

[5] X. Shao, C. Xu, N. Maddage, Q. Tian, M. Kankanhalli, and J. Jin, "Automatic summarization of music videos," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 2, no. 2, p. 148, 2006.

[6] X. Shao, C. Xu, and M. Kankanhalli, "Automatically generating summaries for musical video," in *ICIP '03: Proceedings of the 2003 International Conference on Image Processing*, vol. 2, pp. 547–550, 2003.

[7] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Generation of personalized abstract of sports video," in *ICME '01: Proceedings of the 2001 IEEE International Conference on Multimedia and Expo*, pp. 800–803, 2001.

[8] A. Hanjalic, "Generic approach to highlights extraction from a sport video," in *ICIP '03: Proceedings of the 2003 International Conference on Image Processing*, vol. 1, pp. I–1–4 vol.1, Sept. 2003.

[9] A. Ekin, A. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image Processing*, vol. 12, pp. 796–807, July 2003.

[10] B. Li, H. Pan, and I. Sezan, "A general framework for sports video summarization with its application to soccer," in *ICASSP '03: Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. III–169–72 vol.3, April 2003.

[11] K. A. Peker, A. Divakaran, and T. Lanning, "Browsing news and talk video on a consumer electronics platform using face detection," in *Proceedings of SPIE Conference on Multimedia Systems and Applications VIII*, vol. 6015, pp. 430–435, SPIE, 2005.

[12] W.-N. Lie and C.-M. Lai, "News video summarization based on spatial and motion feature analysis," in *PCM '04: Proceedings of the 5th IEEE Pacific Rim Conference on Multimedia*, pp. 246–255, 2004.

[13] J. Kim, H. Chang, K. Kang, M. Kim, J. Kim, and H. Kim, "Summarization of news video and its description for content-based access," *International Journal of Imaging Systems and Technology*, vol. 13, no. 5, pp. 267–274, 2003.

[14] M. G. Christel, A. G. Hauptmann, H. D. Wactlar, and T. D. Ng, "Collages as dynamic summaries for news video," in *MULTIMEDIA '02: Proceedings of the 10th ACM international conference on Multimedia*, pp. 561–569, ACM, 2002.

[15] C. Kim and J. Hwang, "Object-based video abstraction for video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1128–1138, 2002.

[16] Y. Li, S.-H. Lee, C.-H. Yeh, and C.-C. Kuo, "Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques," *IEEE Signal Processing Magazine*, vol. 23, pp. 79–89, March 2006.

[17] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 345–353, 1996.

[18] Y. Takeuchi and M. Sugimoto, "User-adaptive home video summarization using personal photo libraries," in *CIVR '07: Proceedings of the 2007 International Conference on Image and Video Retrieval*, pp. 472–479, ACM, 2007.

[19] L. He, E. Sanocki, A. Gupta, and J. Grudin, "Auto-summarization of audio-video presentations," in *MULTIMEDIA '99: Proceedings of the 7th ACM international conference on Multimedia*, pp. 489–498, ACM, 1999.

[20] S. Ju, M. Black, S. Minneman, and D. Kimber, "Summarization of videotaped presentations: automatic analysis of motion and gesture," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 686–696, 1998.

[21] K. Miura, R. Hamada, I. Ide, S. Sakai, and H. Tanaka, "Motion based automatic abstraction of cooking videos," in *MIR '02: Proceedings of the 4th ACM international workshop on Multimedia information retrieval*, pp. 29–32, 2002.

[22] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," *HP Laboratories Palo Alto*, 2001.

[23] P. Aigrain, H. Zhang, and D. Petkovic, "Content-based representation and retrieval of visual media: A state-of-the-art review," *Multimedia Tools and Applications*, vol. 3, no. 3, pp. 179–202, 1996.

[24] A. G. Money and H. Agius, "Video summarisation: A conceptual framework and survey of the state of the art," *Journal of Visual Communication and Image Representation*, vol. 19, no. 2, pp. 121–143, 2008.

[25] J. Oh, Q. Wen, S. Hwang, and J. Lee, *Video abstraction*, ch. 14, pp. 321–346. IGI Global, 2004.

[26] H. J. Chien, S. W. Smoliar, and J. H. Wu, "Video parsing, retrieval and browsing: An integrated and content-based solution," in *MULTIMEDIA '95: Proceedings of the 3rd ACM international conference on Multimedia*, (San Francisco, California), Novembro 1995.

[27] B. Wildemuth, G. Marchionini, M. Yang, G. Geisler, T. Wilkens, A. Hughes, and R. Gruss, "How fast is too fast? evaluating fast forward surrogates for digital video," in *JCDL '03: Proceedings of the 2003 Joint Conference on Digital Libraries*, pp. 221–230, May 2003.

[28] J. Nam and A. Tewfik, "Video abstract of video," in *MMSP '99: Proceedings of the IEEE 3rd International Workshop on Multimedia Signal Processing*, pp. 117–122, 1999.

[29] K. Otsuji, Y. Tonomura, and Y. Ohba, "Video browsing using brightness data," in *VCIP '91: Proceedings of 1991 SPIE Conference on Visual Communications and Image Processing*, vol. 1606, pp. 980–989, 1991.

[30] Y. Taniguchi, A. Akutsu, Y. Tonomura, and H. Hamada, "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *MULTIMEDIA '95: Proceedings of the 3rd ACM international conference on Multimedia*, pp. 25–33, ACM, 1995.

[31] M. Mills, J. Cohen, and Y. Y. Wong, "A magnifier tool for video data," in *CHI '92: Proceedings of the 1992 SIGCHI conference on Human factors in computing systems*, pp. 93–98, ACM, 1992.

[32] S. Smoliar and H. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia Magazine*, vol. 1, no. 2, pp. 62–72, 1994.

[33] F. Arman, R. Depommier, A. Hsu, and M. Chiu, "Content-based browsing of video sequences," in *MULTIMEDIA '94: Proceedings of the 2nd ACM international conference on Multimedia*, pp. 97–103, ACM, 1994.

[34] H. Zhang, C. Low, and S. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools and applications*, vol. 1, no. 1, pp. 89–111, 1995.

[35] A. Ferman and A. Tekalp, "Multiscale content extraction and representation for video indexing," in *Proceedings of the 1997 SPIE Multimedia Storage and Archiving Systems II*, vol. 3229, pp. 23–31, 1997.

[36] R. Lagendijk, A. Hanjalic, M. Ceccarelli, M. Soletic, and E. Persoon, "Visual search in a smash system," in *ICIP '97: Proceedings of the 1997 International Conference on Image Processing*, pp. 671–674, 1997.

[37] B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proceedings of the 1995 SPIE Conference on Digital Video Compression: Algorithms and Technologies*, vol. 2419, pp. 2–13, 1995.

[38] H. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognition*, vol. 30, no. 4, pp. 643–658, 1997.

[39] A. Divakaran, R. Radhakrishnan, and K. Peker, "Motion activity-based extraction of key-frames from video shots," in *ICIP '02: Proceedings of the 2002 International Conference on Image Processing*, vol. 1, pp. I–932–I–935 vol.1, 2002.

[40] Y. Zhuang, Y. Rui, T. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *ICIP '99: Proceedings of the 1999 International Conference on Image Processing*, vol. 1, pp. 866–870, October 1998.

[41] C. Toklu and S.-P. Liou, "Automatic key-frame selection for content-based video indexing and access," in *Proceedings of the 1999 SPIE Conference on Storage and Retrieval for Media Databases*, vol. 3972, pp. 554–563, SPIE, 1999.

[42] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," *Multimedia Tools and Applications*, vol. 11, no. 3, pp. 347–358, 2000.

[43] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video manga: generating semantically meaningful video summaries," in *MULTIMEDIA '99: Proceedings of the 7th ACM international conference on Multimedia*, pp. 383–392, ACM, 1999.

[44] M. Yeung and B. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771–785, 1997.

[45] R. Hammoud and R. Mohr, "A probabilistic framework of selecting effective key frames for video browsing and indexing," in *RISA '00: Proceedings of the 2000 International workshop on Real-Time Image Sequence Analysis*, pp. 125–136, 2000.

[46] F. Dirfaux, "Key frame selection to represent a video," in *ICIP '00: Proceedings of the 2000 International Conference on Image Processing*, vol. 2, pp. 275–278, 2000.

[47] W. Wolf, "Key frame selection by motion analysis," in *ICASSP '96: Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1228–1231, 1996.

[48] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *MULTIMEDIA '98: Proceedings of the 6th ACM international conference on Multimedia*, pp. 211–218, ACM, 1998.

[49] L. Zhao, W. Qi, S. Li, S. Yang, and H. Zhang, "Key-frame extraction and shot retrieval using nearest feature line (nfl)," in *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pp. 217–220, ACM, 2000.

[50] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 1280–1289, Dec 1999.

[51] H. Lee and S. Kim, "Rate-driven key frame selection using temporal variation of visual content," *Electronics Letters*, vol. 38, pp. 217–218, 2002.

[52] J. Calic and E. Izquierdo, "Efficient key-frame extraction and video analysis," in *ITTC '02: Proceedings of the 2002 International Conference on Information Technology: Coding and Computing*, pp. 8–10, IEEE Computer Society, 2002.

[53] X.-D. Zhang, T.-Y. Liu, K.-T. Lo, and J. Feng, "Dynamic selection and effective compression of key frames for video abstraction," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1523 – 1532, 2003.

[54] W. Xiong, C. Lee, and R. Ma, "Automatic video data structuring through shot partitioning and key-frame computing," *Machine Vision and Applications*, vol. 10, no. 2, pp. 51–65, 1997.

[55] X.-D. Yu, L. Wang, Q. Tian, and P. Xue, "Multilevel video representation with application to keyframe extraction," in *MMM '04: Proceedings of the 10th International Multimedia Modelling Conference*, pp. 117–123, Jan. 2004.

[56] V. Valdés and J. M. Martínez, "On-line video summarization based on signature-based junk and redundancy filtering," in *WIAMIS '08: Proceedings of the Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 88–91, IEEE Computer Society, 2008.

[57] V. Valdés and J. M. Martínez, "On-line video skimming based on histogram similarity," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 94–98, ACM, 2007.

[58] B. Gunsel and A. Tekalp, "Content-based video abstraction," in *ICIP '98: Proceedings of the 1998 International Conference on Image Processing*, pp. 128–132 vol.3, Oct 1998.

[59] X. Sun and M. Kankanhalli, "Video summarization using r-sequences," *Real-time imaging*, vol. 6, no. 6, pp. 449–459, 2000.

[60] H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 1269–1279, Dec 1999.

[61] I. Yahiaoui, B. Merialdo, and B. Huet, "Automatic video summarization," *MMCBIR '01: Proceedings of the 2001 Conference on Multimedia Content-based Indexing and Retrieval*, 2001.

[62] Y. Gong and X. Liu, "Summarizing video by minimizing visual content redundancies," in *ICME '01: Proceedings of the 2001 IEEE International Conference on Multimedia and Expo*, pp. 607–610, Aug. 2001.

[63] Y. Gong and X. Liu, "Video summarization using singular value decomposition," in *CVPR '00: Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 174–180, 2000.

[64] A. G. Hauptmann, M. G. Christel, W.-H. Lin, B. Maher, J. Yang, R. V. Baron, and G. Xiang, "Clever clustering vs. simple speed-up for summarizing rushes," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 20–24, ACM, 2007.

[65] K. Ratakonda, M. I. Sezan, and R. J. Crinon, "Hierarchical video summarization," in *VCIP '99: Proceedings of 1999 SPIE Conference on Visual Communications and Image Processing*, vol. 3653, pp. 1531–1541, SPIE, 1999.

[66] N. Doulamis, A. Doulamis, Y. Avrithis, and S. Kollias, "Video content representation using optimal extraction of frames and scenes," in *ICIP '98: Proceedings of the 1998 International Conference on Image Processing*, vol. 1, pp. 875–879, IEEE Computer Society, 1998.

[67] T. Liu and J. R. Kender, "Optimization algorithms for the selection of key frame sequences of variable length," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, (London, UK), pp. 403–417, Springer-Verlag, 2002.

[68] T. Liu and J. Kender, "An efficient error-minimizing algorithm for variable-rate temporal video sampling," in *ICME '02: Proceedings of the 2002 IEEE International Conference on Multimedia and Expo*, pp. 413–416, 2002.

[69] T. Liu, X. Zhang, J. Feng, and K. Lo, "Shot reconstruction degree: a novel criterion for key frame selection," *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1451–1457, 2004.

[70] Z. Li, G. Schuster, A. Katsaggelos, and B. Gandhi, "Optimal video summarization with a bit budget constraint," in *ICIP '04: Proceedings of the 2004 International Conference on Image Processing*, pp. 617–620, 2004.

[71] H.-C. Lee and S.-D. Kim, "Iterative key frame selection in the rate-constraint environment," *Signal Processing: Image Communication*, vol. 18, no. 1, pp. 1 – 15, 2003.

[72] L. Latecki, D. DeMenthon, and A. Rosenfeld, "Extraction of key frames from videos by polygon simplification," in *ISSPA '01: Proceedings of the 6th International Symposium on Signal Processing and its Applications*, vol. 2, pp. 643–646, 2001.

[73] L. Latecki, D. de Wildt, and J. Hu, "Extraction of key frames from videos by optimal color composition matching and polygon simplification," in *MMSP '01: Proceedings of the 2001 IEEE Workshop on Multimedia Signal Processing*, pp. 245–250, 2001.

[74] Z. Xiong, R. Radhakrishnan, and A. Divakaran, "Generation of sports highlights using motion activity in combination with a common audio feature extraction framework," in *ICIP '03: Proceedings of the 2003 International Conference on Image Processing*, vol. 1, pp. I–5–8, Sept. 2003.

[75] T. Liu, H.-J. Zhang, and F. Qi, "A novel video key-frame-extraction algorithm based on perceived motion energy model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 1006–1013, Oct. 2003.

[76] S. Han and I. Kweon, "Scalable temporal interest points for abstraction and classification of video events," in *ICME '05: Proceedings of the 2005 IEEE International Conference on Multimedia and Expo*, pp. 670–673, 2005.

[77] J. Bescós, J. Martínez, L. Herranz, and F. Tiburzi, "Content-driven adaptation of on-line video," *Signal Processing: Image Communication*, vol. 22, no. 7-8, pp. 651–668, 2007.

[78] M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 905–921, 1998.

[79] M. Lee, W. Chen, C. Lin, C. Gu, T. Markoc, S. Zabinsky, and R. Szeliski, "A layered video object coding system using sprite and affine motion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 130–145, 1997.

[80] M. G. Christel, "Evaluation and user studies with respect to video summarization and browsing," in *MCAMR '06: Proceedings of the 2006 SPIE Conference on Multimedia Content Analysis, Management, and Retrieval*, vol. 6073, pp. 196–210, 2006.

[81] J. Calic, D. Gibson, and N. Campbell, "Efficient layout of comic-like video summaries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 931–936, 2007.

[82] D. Byrne, P. Kehoe, H. Lee, C. . Conaire, A. F. Smeaton, N. E. O'Connor, and G. J. Jones, "A user-centered approach to rushes summarisation via highlight-detected keyframes," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 35–39, ACM, 2007.

[83] Q. Huang, Z. Liu, A. Rosenberg, D. Gibbon, and B. Shahraray, "Automated generation of news content hierarchy by integrating audio, video, and text information," in *ICASSP '99: Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3025–3028, 1999.

[84] Z. Li, G. Schuster, A. Katsaggelos, and B. Gandhi, "Rate-distortion optimal video summary generation," *IEEE Transactions on Image Processing*, vol. 14, pp. 1550–1560, Oct. 2005.

[85] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *MULTIMEDIA '02: Proceedings of the 10th ACM international conference on Multimedia*, pp. 533–542, ACM, 2002.

[86] H. Sundaram and S.-F. Chang, "Condensing computable scenes using visual complexity and film syntax analysis," in *ICME '01: Proceedings of the 2001 IEEE International Conference on Multimedia and Expo*, pp. 389–392, 2001.

[87] C. M. Taskiran, A. Amir, D. B. Ponceleon, and E. J. D. III, "Automated video summarization using speech transcripts," vol. 4676, pp. 371–382, SPIE, 2001.

[88] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in *MULTIMEDIA '00: Proceedings of the 8th ACM international conference on Multimedia*, pp. 105–115, ACM, 2000.

[89] Y. Ariki, M. Kumano, and K. Tsukada, "Highlight scene extraction in real time from baseball live video," in *MIR '03: Proceedings of the 5th ACM international workshop on Multimedia information retrieval*, pp. 209–214, ACM, 2003.

[90] N. Peyrard and P. Bouthemy, "Motion-based selection of relevant video segments for video summarization," *Multimedia Tools and Applications*, vol. 26, no. 3, pp. 259–276, 2005.

[91] M. Cooper and J. Foote, "Summarizing video using non-negative similarity matrix factorization," in *MMSP '02: Proceedings of the IEEE 6th International Workshop on Multimedia Signal Processing*, pp. 25–28, Dec. 2002.

[92] B. Erol, D. Lee, and J. Hull, "Multimodal summarization of meeting recordings," in *ICME '03: Proceedings of the 2003 IEEE International Conference on Multimedia and Expo*, pp. 25–28, 2003.

[93] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Video abstracting," *Communications of the ACM*, vol. 40, no. 12, pp. 54–62, 1997.

[94] V. Beran, M. Hradis, P. Zemcik, A. Herout, and I. Reznicek, "Video summarization at brno university of technology," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 31–34, ACM, 2008.

[95] E. Dumont and B. Merialdo, "Split-screen dynamically accelerated video summaries," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 55–59, ACM, 2007.

[96] V. Valdés and J. M. Martínez, "Binary tree based on-line video summarization," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 134–138, ACM, 2008.

[97] J. Assfalg, M. Bertini, C. Colombo, A. D. Bimbo, and W. Nunziati, "Semantic annotation of soccer videos: automatic highlights identification," *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 285–305, 2003.

[98] M. Christel, A. Hauptmann, A. Warmack, and S. Crosby, "Adjustable filmstrips and skims as abstractions for a digital video library," in *ADL '99: Proceedings of the 1999 IEEE Advances in Digital Libraries Conference*, pp. 98–104, 1999.

[99] L. Shi, I. King, and M. Lyu, "Video summarization using greedy method in a constraint satisfaction framework," in *DMS '03: Proceedings of 9th International Conference on Distributed Multimedia Systems*, pp. 456–461, 2003.

[100] F. Chen, J. Adcock, and M. Cooper, "A simplified approach to rushes summarization," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 60–64, ACM, 2008.

[101] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden markov models," in *ICIP '02: Proceedings of the 2002 International Conference on Image Processing*, vol. 1, pp. 609–612, 2002.

[102] F. Coldefy and P. Bouthemy, "Unsupervised soccer video abstraction based on pitch, dominant color and camera motion analysis," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 268–271, ACM, 2004.

[103] B. Katz, J. Lin, C. Stauffer, and E. Grimson, "Answering questions about moving objects in surveillance videos," in *Proceedings of 2003 AAAI Spring Symposium on New Directions in Question Answering*, pp. 113–128, 2003.

[104] V. Valdés and J. M. Martínez, "Post-processing techniques for on-line adaptive video summarization based on relevance curves," in *SAMT '07: Proceedings of the 2nd International Conference on Semantic and Digital Media Technologies*, pp. 144–157, 2007.

[105] S. Lu, M. R. Lyu, and I. King, "Semantic video summarization using mutual reinforcement principle and shot arrangement patterns," in *MMM '05: Proceedings of the 11th International Multimedia Modelling Conference*, pp. 60–67, IEEE Computer Society, 2005.

[106] M. Detyniecki and C. Marsala, "Adaptive acceleration and shot stacking for video rushes summarization," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 109–113, ACM, 2008.

[107] H. Bredin, D. Byrne, H. Lee, N. E. O'Connor, and G. J. Jones, "Dublin city university at the trecvid 2008 bbc rushes summarisation task," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 45–49, ACM, 2008.

[108] W. Bailer and G. Thallinger, "Comparison of content selection methods for skimming rushes video," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 85–89, ACM, 2008.

[109] W. Bailer, F. Lee, and G. Thallinger, "Skimming rushes video using retake detection," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 60–64, ACM, 2007.

[110] F. Wang and C. Ngo, "Rushes video summarization by object and event understanding," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 25–29, ACM, 2007.

[111] B. Truong and S. Venkatesh, "Generating comprehensible summaries of rushes sequences based on robust feature matching," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 30–34, ACM, 2007.

[112] K. Yamasaki, K. Shinoda, and S. Furui, "Automatically estimating number of scenes for rushes summarization," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 129–133, ACM, 2008.

[113] B. Li and M. I. Sezan, "Event detection and summarization in american football broadcast video," in *Proceedings of the 2001 SPIE Conference on Storage and Retrieval for Media Databases*, vol. 4676, pp. 202–213, 2001.

[114] V. Chasanis, A. Likas, and N. Galatsanos, "Video rushes summarization using spectral clustering and sequence alignment," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 75–79, ACM, 2008.

[115] S. Ju, M. Black, S. Minneman, and D. Kimber, "Summarization of videotaped presentations: automatic analysis of motion and gesture," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 686–696, Sep 1998.

[116] M. Sugano, Y. Nakajima, and H. Yanagihara, "Automated mpeg audio-video summarization and description," in *ICIP '02: Proceedings of the 2002 International Conference on Image Processing*, vol. 1, pp. 956–959.

[117] S. Aoyagi, K. Sato, T. Takada, T. Sugawara, and R. Onai, "Implementation of flexible-playtime video skimming," in *MMCN '04: Proceedings of the 2004 SPIE Conference on Multimedia Computing and Networking*, vol. 5305, pp. 178–186, 2003.

[118] M. Furini and V. Ghini, "An audio-video summarization scheme based on audio and video analysis," in *CCNC '06: Proceedings of the IEEE 2006 Conference on Consumer Communications and Networking*, pp. 1209–1213, 2006.

[119] S. Lu, M. Lyu, and I. King, "Video summarization by spatial-temporal graph optimization," in *ISCAS '04: Proceedings of the 2004 International Symposium on Circuits and Systems.*, vol. 2, pp. II–197–200, May 2004.

[120] C. Liang, J. Kuo, W. Chu, and J. Wu, "Semantic units detection and summarization of baseball videos," in *Proceedings of the 2004 IEEE Midwest Symposium on Circuits and Systems*, vol. 1, pp. I–297–300, 2004.

[121] Y. Wu, Y. Lee, and C. Chang, "Vsum: summarizing from videos," in *Proceedings of the 2004 IEEE Sixth International Symposium on Multimedia Software Engineering*, pp. 302–309, IEEE Computer Society, 2004.

[122] M. Han, W. Hua, W. Xu, and Y. Gong, "An integrated baseball digest system using maximum entropy method," in *MULTIMEDIA '02: Proceedings of the 10th ACM international conference on Multimedia*, pp. 347–350, ACM, 2002.

[123] S. Dagtas and M. Abdel-Mottaleb, "Multimodal detection of highlights for multimedia content," *Multimedia Systems*, vol. 9, no. 6, pp. 586–593, 2004.

[124] M. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *CVPR'97: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 775–781, 1997.

[125] C. Gianluigi and S. Raimondo, "An innovative algorithm for key frame extraction in video summarization," *Journal of Real-Time Image Processing*, vol. 1, pp. 69–88, 10 2006.

[126] M. Fayzullin, V. S. Subrahmanian, A. Picariello, and M. L. Sapino, "The cpr model for summarizing video," in *MMDB '03: Proceedings of the 1st ACM international workshop on Multimedia databases*, pp. 2–9, ACM, 2003.

[127] F. Chen, M. Cooper, and J. Adcock, "Video summarization preserving dynamic content," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 40–44, ACM, 2007.

[128] G. Ciocca and R. Schettini, "Dynamic key-frame extraction for video summarization," in *Proceedings of the 2005 SPIE Internet Imaging VI*, vol. 5670, pp. 137–142, SPIE, 2005.

[129] M. Albanese, M. Fayzullin, A. Picariello, and V. Subrahmanian, "The priority curve algorithm for video summarization," *Information Systems*, vol. 31, no. 7, pp. 679 – 695, 2006. (1) SPIRE 2004.

[130] M. Koskela, M. Sjöberg, J. Laaksonen, V. Viitaniemi, and H. Muurinen, "Rushes summarization with self-organizing maps," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 45–49, ACM, 2007.

[131] X.-S. Hua, L. Lu, and H.-J. Zhang, "Optimization-based automated home video editing system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 572–583, May 2004.

[132] J. Kleban, A. Sarkar, E. Moxley, S. Mangiat, S. Joshi, T. Kuo, and B. S. Manjunath, "Feature fusion and redundancy pruning for rush video summarization," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 84–88, ACM, 2007.

[133] P. Over, A. F. Smeaton, and G. Awad, "The trecvid 2008 bbc rushes summarization evaluation," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 1–20, ACM, 2008.

[134] M. G. Christel, A. G. Hauptmann, W.-H. Lin, M.-Y. Chen, J. Yang, B. Maher, and R. V. Baron, "Exploring the utility of fast-forward surrogates for bbc rushes," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 35–39, ACM, 2008.

[135] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework," in *ICME '03: Proceedings of the 2003 IEEE International Conference on Multimedia and Expo*, vol. 3, pp. 401–404, 2003.

[136] K. Masumitsu and T. Echigo, "Video summarization using reinforcement learning in eigenspace," in *ICIP '00: Proceedings of the 2000 International Conference on Image Processing*, vol. 2, pp. 267–270, Sept. 2000.

[137] Y.-X. Xie, X.-D. Luan, S.-Y. Lao, L.-D. Wu, P. Xiao, and J. Wen, "Edu: A model of video summarization," in *CIVR '04: Proceedings of the 2004 International Conference on Image and Video Retrieval*, vol. 3115 of *Lecture Notes in Computer Science*, pp. 106–114, Springer, 2004.

[138] T. Mei, C.-Z. Zhu, H.-Q. Zhou, and X.-S. Hua, "Spatio-temporal quality assessment for home videos," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 439–442, ACM, 2005.

[139] Y. Li, S. Narayanan, and C. Kuo, *Movie Content Analysis, Indexing and Skimming Via Multimodal Information*, ch. 5. 2003.

[140] S. Shipman, A. Divakaran, and M. Flynn, "Highlight scene detection and video summarization for pvr-enabled high-definition television systems," in *ICCE '07: Proceedings of the 2007 International Conference on Consumer Electronics*, pp. 1–2, January 2007.

[141] K. A. Peker, I. Otsuka, and A. Divakaran, "Broadcast video program summarization using face tracks.," in *ICME '06: Proceedings of the 2006 IEEE International Conference on Multimedia and Expo*, pp. 1053–1056, IEEE, 2006.

[142] K. A. Peker and A. Divakaran, "Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure," in *ICME '04: Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, pp. 2055–2058, IEEE, 2004.

[143] H. Sundaram and S.-F. Chang, "Video skims: taxonomies and an optimal generation framework," in *ICIP '02: Proceedings of the 2002 International Conference on Image Processing*, vol. 2, pp. II–21–24, 2002.

[144] E. Kasutani and A. Yamada, "The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval," in *ICIP '01: Proceedings of the 2001 International Conference on Image Processing*, vol. 1, pp. 674–677, 2001.

[145] R. Chakravarti and X. Meng, "A study of color histogram based image retrieval," *Information Technology: New Generations, Third International Conference on*, pp. 1323–1328, 2009.

[146] J. Park and J. Nang, "Content based web image retrieval system using both mpeg-7 visual descriptors and textual information," in *MMM '07: Proceedings of the Advances in Multimedia Modeling: 16th International Multimedia Modeling Conference*, pp. 659–669, 2007.

[147] P. Over, A. F. Smeaton, and P. Kelly, "The trecvid 2007 bbc rushes summarization evaluation pilot," in *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pp. 1–15, ACM, 2007.

[148] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pp. 321–330, ACM, 2006.

[149] L. Kotoulas and I. Andreadis, "Colour histogram content-based image retrieval and hardware implementation," *IEE Proceedings on Circuits, Devices and Systems*, vol. 150, pp. 387–393, Oct. 2003.

[150] A. M. Ferman, A. M. Tekalp, and R. Mehrotra, "Robust color histogram descriptors for video segment retrieval and identification," *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 497 – 508, 2002.

[151] A. Mufit Ferman, S. Krishnamachari, A. Murat Tekalp, A.-M. A., and R. Mehrota, "Group-of-frame/picture color histogram descriptors for multimedia applications," in *ICIP '00: Proceedings of the 2000 International Conference on Image Processing*, pp. I–65–68, 2000.

[152] B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 522–529, 1995.

[153] A. Abdesselam and W. Y. Chung, "Spatial distribution of color clusters for image retrieval," in *TENCON '00: Proceedings of the 2000 IEEE Region 10 International Conference*, vol. 1, pp. 290–294, 2000.

[154] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems*, vol. 3, no. 3-4, pp. 231–262, 1994.

[155] H. Zhang, J. Wu, D. Zhong, and S. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognition*, no. 30, pp. 643–658, 1997.

[156] R. Dugad, K. Ratakonda, and N. Ahuja, "Robust video shot change detection," in *MMSP '98: Proceedings of the IEEE 2nd International Workshop on Multimedia Signal Processing*, pp. 376–381, Dec 1998.

[157] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *ICIP '02: Proceedings of the 2002 International Conference on Image Processing*, vol. 1, pp. 900–903, 2002.

[158] W. Ding, G. Marchionini, and T. Tse, "Previewing video data: Browsing key frames at high rates using a video slide show," in *ISDL '97: Proceedings of the 1997 International Symposium on Research, Development and Practice in Digital Libraries*, pp. 151–158, 1997.

[159] M. G. Christel, M. A. Smith, C. R. Taylor, and D. B. Winkler, "Evolving video skims into useful multimedia abstractions," in *CHI '98: Proceedings of the 1998 SIGCHI conference on Human factors in computing systems*, pp. 171–178, ACM, 1998.

[160] A. Ferman and A. Tekalp, "Two-stage hierarchical video summary extraction to match low-level user browsing preferences," *IEEE Transactions on Multimedia*, vol. 5, pp. 244–256, June 2003.

[161] C. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. Delp, "Automated video program summarization using speech transcripts," *IEEE Transactions on Multimedia*, vol. 8, pp. 775–791, Aug. 2006.

[162] L. Bai, S. Lao, A. F. Smeaton, and N. E. O'Connor, "Automatic summarization of rushes video using bipartite graphs," in *SAMT '08: Proceedings of the 3rd International Conference on Semantic and Digital Media Technologies*, pp. 3–14, Springer-Verlag, 2008.

[163] E. Dumont and B. Mérialdo, "Automatic evaluation method for rushes summarization: experimentation and analysis," in *CBMI '08: Proceedings of the 6th International Workshop on Content-Based Multimedia Indexing*, pp. 518–525, 06 2008.

[164] E. Dumont and B. Merialdo, "Rushes video summarization and evaluation," *Multimedia Tools and Applications*, 2010.

[165] T. Ren, Y. Liu, and G. Wu, "Full-reference quality assessment for video summary," in *Proceedings of the 2008 IEEE International Conference on Data Mining Workshops*, pp. 874–883, IEEE Computer Society Washington, DC, USA, 2008.

[166] E. Dumont and B. Mérialdo, "Sequence alignment for redundancy removal in video rushes summarization," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 55–59, ACM, 2008.

[167] H. Bay, T. Tuytelaars, V. Gool, and L., "Surf: Speeded up robust features," in *CVIU '08: Proceedings of the 2008 Conference in Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.

[168] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman and Hall, January 1984.

[169] T.-S. Chua, S.-F. Chang, L. Chaisorn, and W. Hsu, "Story boundary detection in large broadcast news video archives: techniques, experience and trends," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 656–659, ACM, 2004.

[170] L. Chaisorn, T. seng Chua, C. keat Koh, Y. Zhao, H. Xu, H. Feng, and Q. Tian, "A two-level multimodal approach for story segmentation of large news video corpus.," in *Proceedings of 2003 TRECVID Conference*, pp. 51–62, 2003.

[171] P. Browne, C. Czirjek, G. Gaughan, C. Gurrin, G. J. F. Jones, H. Lee, S. Marlow, K. M. Donald, Noel, N. Murphy, N. E. O'connor, A. F. Smeaton, and J. Ye, "Dublin city university video track experiments for trec 2003," in *Proceedings of 2003 TRECVID Conference*, 2003.

[172] Z. Liu and Y. Wang, "Major cast detection in video using both audio and visual information," in *ICASSP '01: Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Washington, DC, USA), pp. 1413–1416, IEEE Computer Society, 2001.

[173] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR '01: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, (Los Alamitos, CA, USA), p. 511, IEEE Computer Society, 2001.

[174] N. O'hare, A. Smeaton, C. Czirjek, N. O'Connor, and N. Murphy, "A generic news story segmentation system and its evaluation," in *ICASSP '04: Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. iii–1028–1031, May 2004.

[175] H.-J. Zhang, Y. Gong, S. Smoliar, and S. Y. Tan, "Automatic parsing of news video," in *ICMCS '94: Proceedings of the 1994 International Conference on Multimedia Computing and Systems*, pp. 45–54, May 1994.

[176] Y. Zhai, A. Yilmaz, and M. Shah, "Story segmentation in news videos using visual and text cues," in *CIVR '05: Proceedings of the 2005 International Conference on Image and Video Retrieval*, pp. 92–102, 2005.

[177] C. Yeh, M. Chang, K. Lu, and M. Shih, "Robust tv news story identification via visual characteristics of anchorperson scenes," in *PSIVT '06: Proceedings of the Pacific-Rim Symposium on Image and Video Technology, 2006*, pp. 621–630, 2006.

[178] L. Chaisorn, T.-S. Chua, and C.-H. Lee, "The segmentation of news video into story units," in *ICME '02: Proceedings of the 2002 IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 73–76, 2002.

[179] A. G. Hauptmann and M. J. Witbrock, "Story segmentation and detection of commercials in broadcast news video," in *ADL '98: Proceedings of the 1998 IEEE Advances in Digital Libraries Conference*, pp. 168–179, 1998.

[180] Z. Liu, E. Zavesky, B. Shahraray, D. Gibbon, and A. Basso, "Brief and high-interest video summary generation: evaluating the at&t labs rushes summarizations," in *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 21–25, ACM, 2008.

[181] K. W. Wilson and A. Divakaran, "Discriminative genre-independent audio-visual scene change detection," in *Proceedings of the 2009 SPIE Conference on Multimedia Content Access: Algorithms and Systems III*, vol. 7255, p. 725502, SPIE, 2009.

[182] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[183] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169–186, 2003.

[184] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001.