Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Video Analysis for Replay Detection in Sport Events

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: David Martínez Junyent

DIRECTOR: Francesc Tarrés Ruiz

SUPERVISOR: Eusebio Carasusan Pérez

DATE: June 15 th 2012

Títol: Anàlisis de Video per la detecció de repeticions en events esportius

Autor: David Martínez Junyent

Director: Francesc Tarrés Ruiz

Data: 15 de Juny de 2012

Resum

El cost de postproducció d'un vídeo d'un esdeveniment esportiu requereix la dedicació de molt recursos i temps en situar sobre el vídeo els moments destacats que s'utilitzaran, per exemple, en la creació del resum del l'esdeveniment. Aquest procés pot ser optimitzat i millorat en quant a eficiència.

Durant el transcurs d'aquest, els moments més destacats solen repetir-se per tal d'oferir l'escena varies vegades i des de diferents punts de vista.

Aquest treball té com a objectiu principal la detecció d'aquestes repeticions per tal d'identificar els moments destacats i senyalitzar-ho per tal d'agilitzar el procés de postproducció. Els resultats formaran part del projecte CENIT-E BUSCAMEDIA CEN20091026, desenvolupat als estudis de Televisió de Catalunya (TVC) i que tracta de generació automàtica mitjançant l'anàlisi de continguts.

S'ha desenvolupat un software capaç de detectar les repeticions que apareixen en diferents tipus d'esdeveniments esportius, principalment futbol. Aquest, implementa diferents modes d'operació que veurem explicats en detall al llarg de la memòria. Trobem des d'un mode mes aviat manual fins a un completament automàtic i es mostren els percentatge d'èxit obtinguts després de realitzar proves funcionals utilitzant vídeos de la basa de dades de TVC.

L'estructura del treball s'ha dividit en cinc grans apartats:
El primer capítol comença introduint-nos en el context on es situa el projecte, proposant els objectius que es volen assolir, així com també parla sobre les dades i eines utilitzades pel seu desenvolupament.

Posteriorment, s'exposarà l'estat de l'art amb un recull dels mètodes més emprats per la detecció de repeticions i que han estat els fonaments sobre els que hem desenvolupat la nostra metodologia.

El tercer capítol és el més llarg i complex. Conté tot el procés d'experimentació i millores plantejat des de l'inici fins arribar al sistema que s'ha implementat.

D'altra banda, el següent apartat ens fa cinc cèntims de la part tècnica i exposa en forma de diagrama de blocs l'algorisme implementat, explicant els mètodes possibles per utilitzar el sistema.

Finalment, l'últim capítol recull tot els resultats i conclusions extretes després d'aplicar l'algorisme en un conjunt de vídeos extrets de la base de dades de TVC, així com també l'aplicació del mateix en altres àmbits com vídeos de Formula1.

Title: Video Analysis for Replay Detection in Sport Events

Author: David Martínez Junyent

Director: Francesc Tarrés Ruiz

Date: June, 15th 2012

Overview

The postproduction cost of a sport event video requires lots of resources dedication and expenses of time trying to find the best highlights moments that will be used, for instance, in creating the summary of the event. This process can be optimized and improved in efficiency.

During the event, the most important moments are repeated to offer to the audience the outstanding scene several times and from different points of view.

The objective of the project is to automatically find the replays in live or pre-recorded transmission and accelerating the post-production process. The results will be part of the project CENIT-E BUSCAMEDIA CEN20091026, developed in the studios of Televisió de Catalunya (TVC) and which are focused on automated generation through content analysis.

A software has been developed to detect the replays for different kind of sport events, principally soccer. This, implements many operation modes detailed during this report. We find from a mode rather manual to a full automatic mode, and moreover the percentages of success are presented after testing then using some videos from the TVC database.

The structure of the work has been divided into five major sections:
The first chapter begins by introducing us to the context in which it places the project, proposing the objectives to be achieved, and also discusses the data and tools used for their development.

Subsequently, there is exposed the state of the art with a collection of methods used for the detection of repeats, which are the foundations on which we developed our methodology.

The third chapter is the longest and complex. This contains the entire process of experimentation and improvements planned from the inception until the system implemented.

In addition, the following section talks about the technical and exhibits the algorithm implemented in form of block diagram detailing all the operation modes.

Finally, the last chapter contains all the results and conclusions after applying the algorithm on a set of videos taken from the database o f TVC, as well as its application in other areas such as Formula1 videos.

Acknowledgments

Francesc Tarrés for his services, help and support at all times

Eusebi Carasusan for his willingness, support and contribution of ideas

Dani Giribet for his contributions in aid of knowledge

This work has been possible thanks to funding from the Centro para el Desarrollo Tecnológico Industrial as part of the Spanish National Research CENIT BUSCAMEDIA 2009 ref. CEN20091026.

Televisió de Catalunya to provide the resources and tools needed to develop this project

INDEX

INTRODUCTION

Nowadays, multimedia content has increased almost exponentially. So, there is an urgent need to establish effective and efficient methods for information management, classification and retrieval.

The broadcasting of sport events is one of the biggest attractions in television. The Broadcast networks spends large amount of resources in covering many events, like soccer matches or Formula 1 races, and personalizing content for many different types of audiences. In particular, soccer is one of the most popular sports around the world and their live broadcasting has huge audience. Often, following the end of the match these contents become material for sport or talk programs, news, etc. So, the video postproduction acquires a vital role.

During broadcasting all the media is ingested into the database of the system. This way, all content from an event could be accessible easy and directly in digital format from whatever PC with access to the network just finished the event. Furthermore, this data may attach textual information (metadata), summarizing the content both: as a data, for instance frames number, time codes, etc. and as content, describing the highlights.

The postproduction should manage manually all this amount of data to generate specific videos, for instance, the summary of a football match. So, each video ingested must be analysed to extract the highlighted moments to include in the summary. Moreover, usually the content includes lots of additional recorded data that is not relevant for the summary.

So, many hours are spent performing repetitive tasks, such as annotating and selecting video segments to create event reports. Furthermore, broadcast networks also have to create many specific reports and news shows for different regions that include local highlights amongst the general interest ones and sometimes they have not additional metadata and the unique reference is the visual content.

The R&D engineering department of the Televisió de Catalunya[1] (TVC) develops technological projects providing services to others departments and

[1] TVC (www.tv3.cat) is the public broadcaster of Catalonia (6 DTT channels, 4 Radio stations, Internet, mobile/smartphones, etc.). With more than 25 years of experience TVC currently employs around 2000 people. TVC was founded in 1983 with the mission of restoring the Catalan language and culture to their rightful place, and belongs to the Catalan Broadcasting Corporation (Corporació Catalana de Mitjans Audiovisuals – CCMA, www.ccma.cat), the public group that manages the radio and TV broadcasting services of the Catalan government. Its whole video and audio workflows are fully digitized. TVC has a large production capacity, generating or commissioning more than 15.000 hours of new content per year. At the end of 2011, more than a million digital assets (over 3 petabytes of A/V data) are available to be edited, annotated, aggregated, published and broadcast. Apart from traditional broadcasting services, TVC covers a wide range of Internet services, from on-demand video (3alacarta, www.tv3.cat/3alacarta) to mobile apps, extensive social network presence and traditional websites.

externals, at both, National and European levels. Currently, is working on the project CENIT-E BUSCAMEDIA 20091026 [1] which is developing mechanisms of automation for some of these tasks specifically in soccer, in order to bring down the cost, and freeing editors from mechanical tasks, allowing them to focus in the more creative ones

In order to address this problem GENA[2] [2] uses a CBR[3] approach (reasoning mechanisms of artificial intelligence). TVC kindly provided us with a collection of audio-visual narratives (soccer game summaries, Formula 1 reports, news shows) generated by professional reporters, together with the complete original assets[4] from where the narratives were generated.

Each of these narratives was captured as a case. For example, in the case of soccer summaries, a case consists of the complete original soccer game (complete audiovisual content plus metadata), a description of the type of summary desired for the game, and the actual summary.

Given a new request to generate a summary, GENA retrieves similar cases of summaries generated by professional reporters, and generates a new candidate summary ready to be directly broadcasted.

To apply CBR trough a completely automatic way the audiovisual content must be analyzed deeply. So, the different scenes must be identified and differentiated to find the highlights.

Our project is a very specific part inside this big topic. It is focused on one of the factors to analyse from the media: the replays.

Replays are one of the most reliable indicators of event highlights because with them another chance to watch the content is given to the audience. Therefore, as a first hypothesis, it seems possible to say: "...If the most outstanding moments of a match are replayed, surely most of them should be included in the summary. So, we can get a set of points of interest from the video if we know where replays are placed..."

This is only one indicator to take into account in the overall project, but it composes by itself an independent one. In fact, this is going to be the main goal to achieve, which consist in analyzing soccer videos offering solutions to detect its replays and help in the postproduction process of video summarizing.

The study, analysis, research of algorithms and development of final software has been divided in different sections that follow the structure explained below:

---

[2] GENA is a case-based reasoning system capable of generating audio-visual narratives by drawing from previously annotated content.

[3] CBR means **Case-Based Reasoning** and is the process of solving new problems based on the solutions of similar past problems.

[4] Asset is the name assigned to each video ingested on the database, independently of its type.

First chapter is going to introduce the big picture of the project. So, it talks about the goals to achieve, shows an example of the media data to analyse that is and used to test the proposed algorithm. This chapter concludes with a presentation if the processing tools that have been used for its development. The media data comes from TVC databases. All the audiovisual material used for testing is explained more extensively on the APPENDIX A. The programming language used has been C++, implemented using the Visual Studio 2010 and the most important external libraries included have been the FFMPEG, OpenCV and LibXML libraries.

Following this overview of the project, the second chapter is devoted to the study and analysis of the current state of the art in replay analysis. It summarizes the last progress in this area, explains the methodology being used by different strategies and summarizes some of the results which have been reported in literatures.

The third chapter contains the most important part of this project. It starts presenting a first proposal of an algorithm based on one of the best performance methods have been presented in the previous chapter. Then, it explains step by step some of the modifications that have been implemented with respect to the previous algorithms and explains in detail the improvements and achievements that have been obtained.

Next chapter shows the final block diagram of the implemented algorithm. It explains the two modes of operation which are activated using the command line: the batch mode and the interactive mode. Finally, it describes by means of using examples the output XML format which is used for exporting the results.

The last chapter is devoted to reporting the results of different tests soccer videos and analyses the results to extract conclusions about the proposed methodology. Moreover possible lines for future work are proposed and other kinds of videos as Formula 1 are tested.

Chapter 1 Replay Detection in Sport Events: The Big Picture

First chapter presents the big picture of the project and places it in its respective framework. Furthermore, there is a briefly description about the media and processing tools used for the development and test of the system.

1.1. Main Goal

Currently, many techniques of automatic detection and indexing are been developed. These studies tend to focus on the analysis of the audio or the video components in the live or recorded material, offering a wide range of possibilities. For instance, it is possible to found studies for whistle detection, field lines detection, predominant colour or replay detection, all of these methods may be used in automatic video generation.

This project is focused towards the development of a new technique for replay detection that will try to improve the results obtained by other strategies that may found in literature. The main idea is to detect those moments considered important in broadcasting and which have been replayed extracting their respective time codes in some exportable format.

The necessity of implement this kind of methods is because of the big amount of hours spend by the editors performing repetitive tasks, such as annotating and selecting video segments to create event reports. This fact supposes costs and resources that can be supplied by using automatic methodologies. So, this will help in the process of postproduction to the point of being able to make a video-summary using only these short repetition segments.

The figure 1.1 shows the typical workflow associated to the TV broadcasting process. It is useful to place the framework of this project in the overall system.

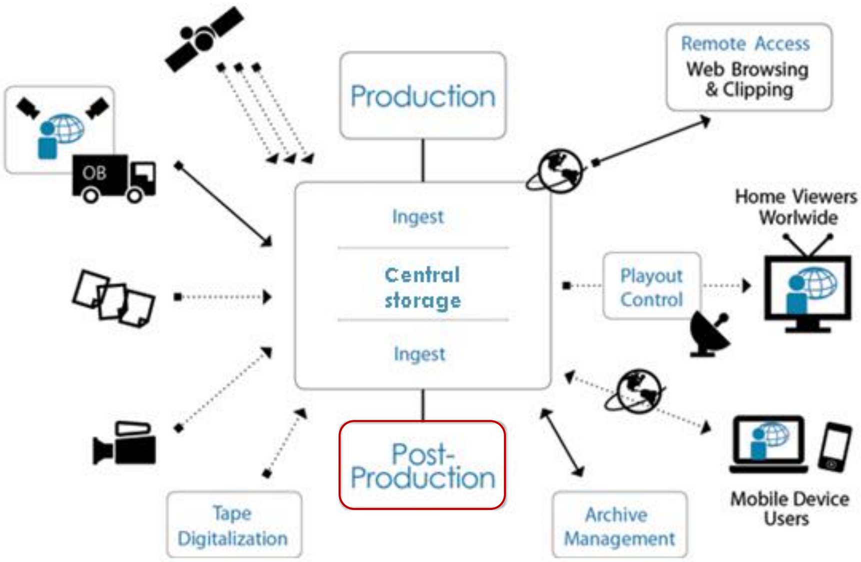


Fig. 1.1 TV broadcasting workflow

The application has to be executed using command lines, allowing launching batch processes. So, it has to be sought in process efficiency, not being necessary the design of a graphical interface.

The typical application cases are principally the first division Championship of Spain's football (Liga BBVA), as well as the Champions league. Normally second and third divisions have a worst realization, so, only the best produced cases will be considered from these divisions. Furthermore, the application should be applicable to other kind of events as F1.

Finally, the output of the algorithm has to be usable and exportable to other project applications developed in TVC. The typical format used to import and export data among different applications is XML. So, the output of the algorithm must be implemented using the same format.

The following sections will be devoted to explaining the media format that has been used to develop the algorithm and the output files.

1.2. Input data format

The resulting software has to be used by TVC, as part of the project CENIT-E BUSCAMEDIA 20091026 [1] about automatic video generation through the content analysis. So, the media to use is a conditioning factor to take into account in the designing process.

The data format used in TVC media has the following properties: The codec used is DVCPRO25, and the resolution 720 x 576. Its ratio of frames per second (FPS) is 25 and the Bit rate of 25Mbits/s. Its compression ratio is 5:1, and is compressed using discrete cosine transform (DCT) without using temporal redundancy of the video.

Video duration is between two and three hours and they are normally edited, containing playtime, replays, logos and so on. Furthermore all of them contain lots of additional recorded content, such as before or after the event conclusion, the rest and in some cases even the press conference after the game.

Many different videos from TVC database have been used in order to check the functionally of the algorithm. In addition, some videos have been created from pieces of other longer videos, especially for getting started. All of them are detailed in the APPENDIX A. The table 1.1 summarizes some of theses features.

Champions League match: F.C Barcelona vs F.C Copenhagen	
Frames number	208.151 frames
Duration	02h 18min 46s 1frame
Replays number	54 replays

Table 1.1 Video A.1

This example corresponds to a champions league match. The duration is around 2h and quarter, so, in addition to the 90 minutes of football content another 45 minutes are included and it presents 54 replays.

The complete case and other video cases could be found on the APPENDIX A

1.3. Processing tools

The algorithm uses external image and video libraries. These libraries have been chosen in the development of other modules of the project. Therefore, we have followed these recommendations in order to maintain the compatibility of the different software modules used in TVC.

The complete analysis can be found in the document: "Descriptors de moviment per l'analisi de continguts audiovisuals (25th July, 2011), by Arnau Raventós Mayoral" available in the UPC repositories. [3]

1.3.1. Image processing

All the programming and coding has been developed using Visual Studio 2010[5] environment.[4]

The programming language used has been C++, tacking advantage especially of the open source OpenCV library. [6]

The best characteristic of OpenCV libraries is their high performance in image analysis and processing algorithms. It is used in big companies such as IBM, Microsoft, Intel, SONY, Siemens, Google and Yahoo as well as in many research centres as Stanford or Cambridge. This library is focused towards images and video capture, management and processing, and contains many of the state of the art algorithm for image processing, object detection and pattern recognition. Moreover, the library is very well documented through manual, books and internet forums.

Another reason to choose C++ and OpenCV library for programming is that other parts of the overall project has been implemented using them, and for compatibility and coherence the same format has been chosen.

1.3.2. Video decoding

OpenCV libraries require a video conversion from TVC format DVCPRO25 to MP4. This has been implemented using the FFMPEG software. [6]

FFMPEG (Fast Forward MPEG) are a set of open source libraries able to process, codify and decode video and audio data. It is written in C language

[5] Visual Studio 2010 is a powerful IDE that ensures quality code throughout the entire application lifecycle, from design to deployment. Whether you are developing applications for SharePoint, the web, Windows, Windows Phone, and beyond, Visual Studio is your ultimate all-in-one solution.

under LGPL licences. Its principal library is named “libacvodec” and is the coding and decoding responsible. It offers a high performance and easy usability through the command line.

Moreover, due the high size of the data to manage, each video has been coded decreasing the resolution to 360x288 pixels. As TVC stores video and audio independently, the audio has not been considered.

The figure 1.2 shows the command line required by FFmpeg to transcode the information into the new file format:

“ffmpeg.exe -y -i D:\PathInput\Inputvideo.avi -deinterlace -vcodec mpeg4 -s 360x288 -threads 2 -g 1 D:\PathOutput\OutputVideo.avi"

Fig. 1.2 Logo transition “Champions”

1.3.3. XML library

Normally, the TVC projects export their final results in XML[6] format, allowing the possibility to use them in others applications just parsing the respective XML to access data.

The library chosen to create the XML has been Libxml2. It is an open library for parsing XML documents, and is written in the C programming language providing bindings to C++ among others.

The libXML code is highly portable, since it depends on standard ANSI C libraries only, and it is released under MIT license.

The next chapter is going to introduce the state of the art about replay detection. All information has been extracted from EEE Xplore. This is a scholarly research database that indexes, abstracts and provides full-text for articles and papers on computer science, electrical engineering and electronics.

The database mainly covers material from IEEE[7] and IET[8]. The IEEE Xplore database contains over two million records.

[6] XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications (all free open standards).

[7] The Institute of Electrical and Electronic Engineers (IEEE) is a non-profit professional association headquartered in New York City that is dedicated to advancing technological innovation and excellence. It has more than 400,000 members in more than 160 countries. It produces 30% of the world’s literature in the electrical and electronics engineering and computer science fields. IEEE is one of the leading standards-making organizations in the world.

[8] The Institution of Engineering and Technology (IET) is Europe’s biggest professional society in engineering and technology worldwide. It provides advice on all areas of engineering, regularly advising parliament and other agencies.

Chapter 2 State of the Art in Replay Detection Algorithms

In recent years, researchers have reported many approaches of replay detection in articles on computer science, electrical engineering and electronics literatures.

In this chapter, some of the more representative methodologies developed in recent years to detect replays in soccer games will be presented. In fact, as will be seen, recent techniques may be categorized in two classes.

First methodology (most practised) is to take into account logo transitions. In most cases, replays are preceded and closed by a logo transition. This transition has distinctive properties over the others frames. For instance, they present a luminance increase or a noticeable change on the image. The figure 2.1 shows an example of logo transition that appears just before a replay.

Fig. 2.1 Logo transition "LigaBBVA"

Furthermore, the transition length is on average around 20 frames, and the replay duration usually corresponds to less than two minutes. So, its transition pair is going to be placed near.

Taking advantage of all these characteristics, the strategy is to divide the process in two phases clearly different:

First criterion consists into find a logo template from the video and then to perform a matching research using the template along all video frames, pairing them respectively to obtain the replays. Unfortunately, there is not a unique criterion to design the logo and the transition. Every event could use a different graphics, changing duration, position, size, form and colors. So, the methodologies implemented must consider all these possibilities.

The second criterion tries to identify shot[9] transition patterns. Knowing the kind of shot broadcasted at all times and using a complex statistical and probabilistic algorithm the system proposes the instants where replays could be placed. Typically, replays are composed by much than one sequence from different

[9] In film, a shot is a continuous strip of motion picture film, created of a series of frames that runs for an uninterrupted period of time. Shots are generally filmed with a single camera and can be of any duration.

cameras and almost based on detail and short shots. So, through the study of a big amount of cases some patterns could be predefined and then searched on the video.

To check the correct functionality of the implementations done, two values are calculated: the recall and the precision.

The recall value is computed by dividing the correct transitions detected from the really existing in the video, and the precision value by dividing the correct transitions detected from the total detected. They are given using a percentage and are useful to get an idea of the overall performance of the detector.

In the following sections some of the replay detection strategies based on logo transitions and shot transitions patterns are explained.

2.1. Replay detection based on semi-automatic logo template

This paper [7] proposes a methodology based on logo template detection. However, the system is not able to identify the logo by itself and that makes the process semi-automatic.

This method divides the process in two phases: Identify logo template, and to find replays using it. The figure 2.2 shows its block diagram.

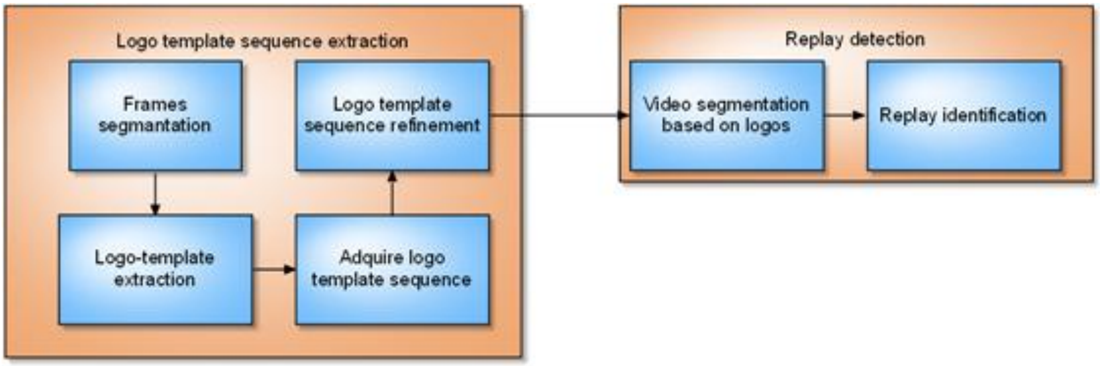


Fig. 2.2 Block diagram of the process

2.1.1. Logo template sequence extraction

To detect the logo template a sequence of frames where the logo can be clearly identified is required. Then, the best frame containing the logo must be selected. As the logo is a graphical image overlapped to the recorded content a segmentation process is performed to separate both, and extract only the logo from the whole frame. It has been done applying the JSEG algorithm.

The JSEG algorithm follows the process shown in the block diagram of the figure 2.3.

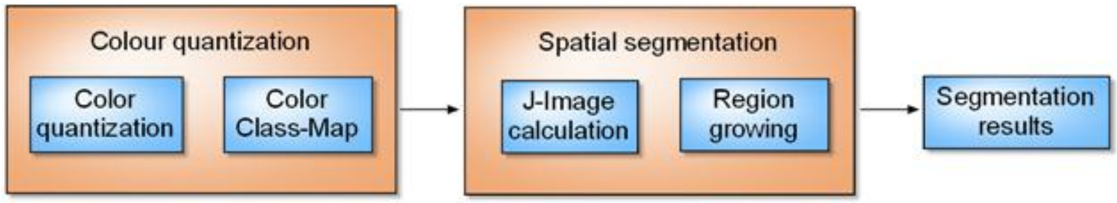


Fig. 2.3 Block diagram of the JSEG algorithm

The basic idea of the JSEG method is to separate the segmentation process into two stages: color quantization and spatial segmentation.

In the first stage, colors in the image are quantized to several representative classes that can be used to differentiate regions in the image. This quantization is performed in the color space without considering the spatial distributions of the colors. Then the image pixel values are replaced by their corresponding color class labels, thus forming a class-map of the image. The class-map can be viewed as a special kind of texture composition.

In the second stage, spatial segmentation is performed directly on this class-map without considering the corresponding pixel color similarity. The benefit of this two-stage separation is clear.

It is a difficult task to analyze the similarity of the colors and their distributions at the same time. The decoupling of color similarity from spatial distribution allows development of more tractable algorithms for each of the two processing stages.

Applying a criterion based in local windows, high and low values correspond to possible boundaries, and finally, a region growing method based on the multi-scale J-images segments the image. The figure 2.4 shows a theoretical result of applying the JSEG algorithm. For more information about JSEG algorithm view its reference [8].

Fig. 2.4 Example of segmentation result using the JSEG algorithm

After the image segmentation the frame is divided into region containing logo and not containing. Then, the same methodology is applied to the previous and following frames. Finally, comparing the overlap ratio and the colour distance between frames is possible to recover the common area between contiguous frames which has to correspond to the logo.

At the end, a refinement process is performed to reduce erroneous areas, and the logo template is obtained. The figure 2.5 shows the evolution from a starting frame to the last area used as template.

Fig. 2.5 Results from segmentation and refinement process

2.1.2. Replay detection

Once acquired the logo template, its colour histogram is computed and a matching process is performed establishing a threshold of coincidence to accomplish. Finally checking the duration between transitions they are paired.

2.1.3. Results

The performance of the proposed method was evaluated on several MPEG-1 format sports video. These sports videos were collected from TV and contain different kinds of logo transitions. The total size of the videos is about 10GB. In these videos, the replay segments were marked manually as ground truth to test the proposed replay detection method.

The paper only details the overall result of the experimentation, without showing partial results. The table 2.1 shows them.

Video cases	# total replays	# replays detected	# correct replays detected
Totals	336	325	321

Table 2.1 Test results

The results presented in this paper are really good. From 336 replays to find, the process detects 325, and 321 are correct. This numbers suppose a recall of 95.5% and Precision of 98.7%.

This high values make this system a very good candidate to detect replays, but has the problem that is semi-automatic because each video needs to introduce the initial sequence to extract the template, and therefore is not suited to the goals presented.

2.2. Replay detection in broadcast sports videos

This paper [9] is also based on logo template detection. The main advantage respect the previous methodology is that this time the process detects the initial logo transition completely automatic instead of introduce it manually. So, the process can be considered totally automatic.

The figure 2.6 shows its block diagram.

Fig. 2.6 Block diagram of the process

2.2.1. Logo template detection

The logo transitions suppose a big visual variation in the video content. So, the idea is to identify a frame sequence around 20 frames where something indicates a significant change.

For that purpose, the mean square difference between the intensity of contiguous frames is computed, and then a median filter is applied to the results normalizing the peaks. This produces that on the areas of the video where the frames differs too much between them the difference emphasizes them.

Then, a threshold is defined and if a sequence of frames is over this threshold and have the length estimated should be catalogued as a logo transition.

The figure 2.7 shows on the left an example of logo transition and on the right its resulting graphic obtained. The X and Y axis correspond respectively to the number of frame and to the value of its intensity mean square difference computed.

Fig. 2.7 Logo transition and Intensity mean square difference after applying the median filter

Once detected the transition, the best frame containing the logo has to be selected as a template. So, the most representative frame is chosen.

2.2.2. Replay detection

Finally, a matching process is performed using this template. The shape, the colours and a fixed threshold will determinate the frames coincident. Then, pairing them following length criterions the replays are focalized.

2.2.3. Results

Five games, in which four are soccer games in the FIFA World Cup 2002 and the other one is a table tennis game between J.O. Waldner and L.H. Kong in Olympic Game 2000, are used to test the performance of logo detection. The table 2.2 shows the success of results.

Video cases	# total logos	# logos detected	# correct logos detected	# total replays	# detect replays	# correct replays detected
Cameroon vs Germany	82	81	81	42	38	36
China vs Turkey	102	100	100	55	38	32
England vs Sweden	90	88	87	45	43	42
USA vs Portugal	132	132	131	67	58	54
Waldner vs Kong	56	55	55	-	-	-
Totals	462	456	454	209	177	164

Table 2.2 Test results

The recall and precision values computed to detect the logo template correspond to 98.26% for recall and 99.56% for precision. Then, only halves of the soccer matches was considered to check the replay detection success. The recall and precision values decreases respect the previous. Now, the recall was 78.46% and the precision 92.65%.

2.3. Scene transition structure analysis

In this case [9], the methodology consists in analysing the shots that compose the video and the relations between concatenated shots. Then, finding in the video the typical shot sequences used to replay for example a goal some correlations may be found. The figure 2.8 shows its block diagram.

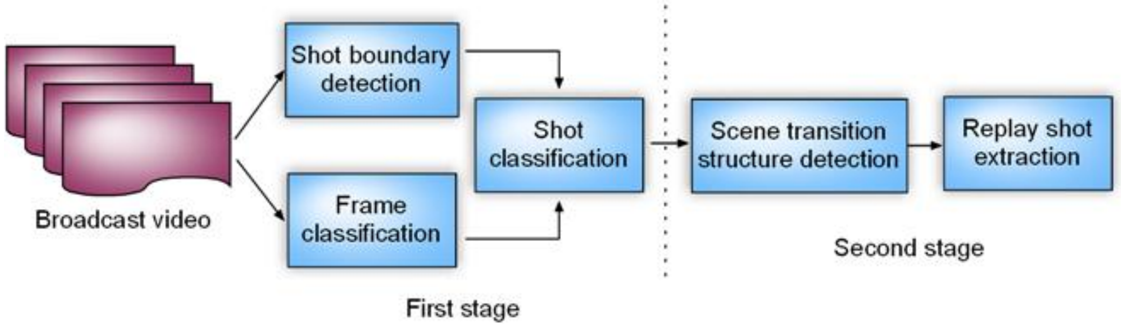


Fig. 2.8 Block diagram of the process

2.3.1. Shot boundary detection and classification

The first stage (Fig.2.8) is based on shot detection and classification. Three classes are possible: Long shot (LS), Mid Shot (MS) and close-up view (CV). The figure 2.9 shows and example of each type respectively.

Fig. 2.9 Shot view classification

(a) Long shot, (b) Mid shot, (c) Close-up view

For shot detection, the model uses the commercial software M2-Edit Pro. Then, for the classification, the algorithm realizes what is said “soccer field region extraction”. It consists into classify the frame depending on the green area detected corresponding to the field with respect whole frame.

Then, if the field size detected is big, processes of object segmentation from the field region are applied to distinguish between the three kinds of shots. Otherwise, the system uses edge detection to distinguish between close-up view and the mid shots.

The classification criterion of shot follows the block diagram shown on the figure 2.10.

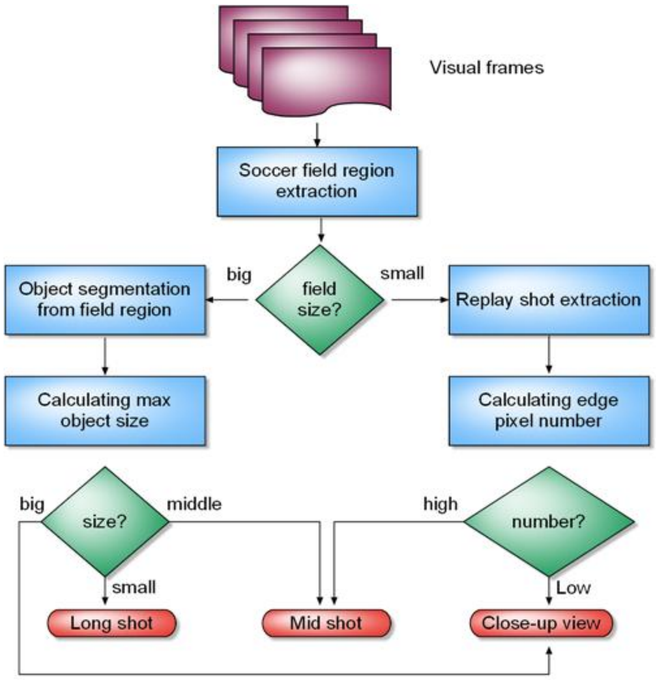


Fig. 2.10 Shot view classification

2.3.2. Scene transition structure detection

The second stage (Fig.2.8) is based on strategy. For instance, in soccer games the close-up views are often launched before and after each replay. So, it could be used to search for replay scenes.

Predefining scene transition structures, a dynamic programming algorithm (DPA) is applied to detect coincident shots sequences. Examining the total matching cost and then tracing back using spatial patterns the best correlations can be extracted.

2.3.3. Replay shot extraction

The model proposes next rules to predefine the replay shot sequence:

1. CV shots are not replays scenes.
2. LS and mid MS can be replay scenes, but any MS preceding another LS is considered as a replay scene.
3. Any LS or MS can be a replay scene if it is too short.

2.3.4. Results

Three broadcast game videos (totally 2.5 hours) from England Premier League (EPL) were used for testing. After the first step processing, a shot classification accuracy of 94% (89 incorrect in totally 1328 shots) is achieved.

Then, three pattern structures are defined to check correlations. These sequences are:

1. CV-MS-MS-CV
2. CV-CV-LS-MS-CV
3. CV-CV-CV-MS-CV

The table 2.3 shows the results achieved using this methodology.

Video cases	# Total replays	# replays detected	# correct replays detected
Bolton vs Liverpool (first half)	23	28	21
Everton vs Manchester United (first half)	23	20	18
Manchester City vs Birmingham (first half)	12	16	11
Totals	58	64	50

Table 2.3 Test results of logo detection

The recall and precision are 86.20% and 78.13% respectively. There are some factors that cause false detection and to miss replays. One factor could be shot classification errors on first stage, and another factor, the suitability of defined structures. Although, the method is computationally fast and the performance is promising.

2.4. Robust replay detection algorithm for soccer video

This paper [13] also uses the logo transition as the detection idea. This strategy will be presented in greater depth due to its higher performance and because it has been the inspiration for the detection algorithm proposed in this thesis.

This model proposes a completely automatic methodology for logo template detection and then uses matching and a set of rules to find replays. First part is based in two different stages: logo candidate set generation and logo template detection. The figure 2.11 shows its block diagram.

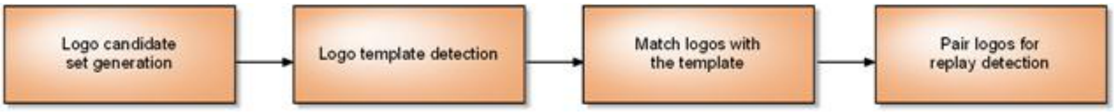


Fig. 2.11 Block diagram of the process

2.4.1. Logo candidate set generation

The main goal of this stage is to distinguish a set of frames candidates to be the logo template. This could be done taking advantage from the high luminance normally presented in logos. Logos are made by computer graphics designers, and therefore, normally have a greater range of distinctive colours than the images usually grabbed by cameras. The figure 2.12 shows this high luminance property of different logo examples.

Fig. 2.12 Logo examples

The model needs to distinguish which frames present more luminance than the others, so it computes the difference among adjacent histogram bins for each frame. Then, it calculates the intensity frame difference between adjacent frames (formula 2.1), and moreover, taking advantage of the typical length of 20 frames of the logo transitions, it accumulates the differences in windows of 20 positions (formula 2.2). It allows concentrating more values and therefore amplifying the differences.

$$\mathrm{Diff}(m) = \sum_{k=1}^{\mathrm{binsize}} \frac{(I(k) - I(k-1))^2}{\mathrm{Max}(I(k), I(k-1))} \tag{2.1}$$

$$\mathrm{AccDiff}(n) = \sum_{m=n-19}^{n} \mathrm{Diff(m)}. \tag{2.2}$$

Finally, the results are filtered by a threshold and a set of logo candidates are selected. This threshold is fixed experimentally. Next figure shows a graph from and empirical example. On the figure 2.13, the red colour represents accumulated intensity, and the blue the threshold fixed. All frames over the threshold will be candidates.

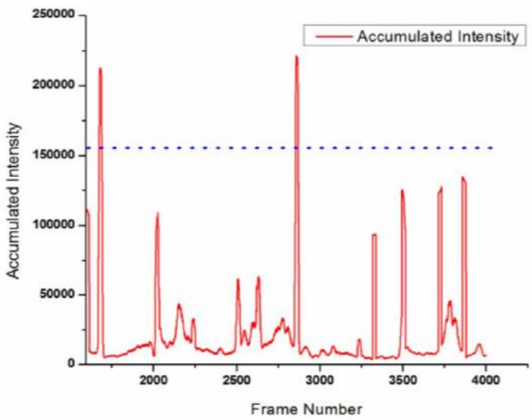


Fig. 2.13 Accumulated intensity from a video segment

2.4.2. Logo template detection

After the candidate set generation, the logo template detection process should be performed. The main goal is to choose one candidate as the most representative of the set. First, is needed to filter as erroneous candidates as possible, and therefore a k-mean algorithm is used to divide the candidate set of images in two groups. The value used to identify and differentiate each image is the mean luminance, and is computed using the formula 2.3.

$$\text{lum\_mean} = \frac{\sum_{i=0}^{\text{binsize}-1} I[i] * i}{\sum_{i=0}^{\text{binsize}-1} I[i]}$$

(2.3)

The k-mean algorithm [12] is an image processing tool able to cluster images from a list. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori.

The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different results. So, the better choice is to place them the most separated as possible.

The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done.

At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the new data set points and the nearest new centroid. So, a loop is generated. As a result of this loop we may notice that the k centroids will change their location step by step until they do not move any more. These finals centroids will be the resulting ones and all points will be associated to one of them. The figure 2.14 shows an illustrative example. The colours represent different clusters, and starting centroids are recalculated changing location.

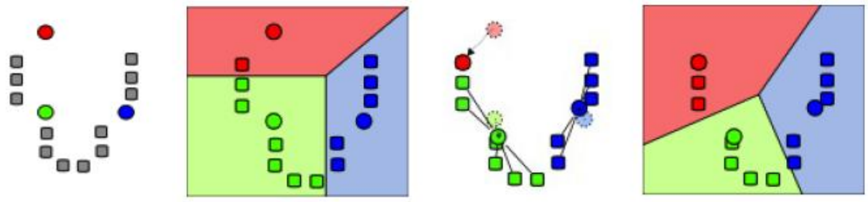


Fig. 2.14 K-mean algorithm example

Finally, after the classification, the frames included to the high centroid are selected as final candidates because of the high luminance characteristic of logos explained before.

At this point, a unique frame should be selected as logo template among all final candidates, so the difference matrix is performed. It consists in computing the difference among all the images in the set (representing each one as a “H_{number}” in the formula 2.4) and to arranging the results as a matrix (formula 2.5). At the end, adding all the differences of each row respectively, the row with the lowest value will correspond to the image more similar to the rest in the set, and will be selected as logo template.

$$d_{mn} = \sqrt{\sum_{i=0}^{\text{binsize}} \left[\frac{H_m(i)}{\sum_{j=0}^{\text{binsize}} H_m(j)} - \frac{H_n(i)}{\sum_{j=0}^{\text{binsize}} H_n(j)} \right]^2} \tag{2.4}$$

$$\begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{m1} & \cdots & d_{mn} \end{pmatrix} \tag{2.5}$$

2.4.3. Match logo with logo template

Once the logo template has been detected, a matching process starts to find the logo template coincidences along all the video. Again, the luminance values are used for comparing the logo template and each frame. If the value is under a new threshold fixed the frame is marked as logo transition.

The figure 2.15 shows the results after and before the matching process. In the left graph, the red colour represents the mean intensity of a set of video frames, and the blue colour to the logo intensity value. The right graph corresponds to the difference between these components. So, as could be seen the frames with a mean luminance value similar to the template should be near the zero after the subtraction.

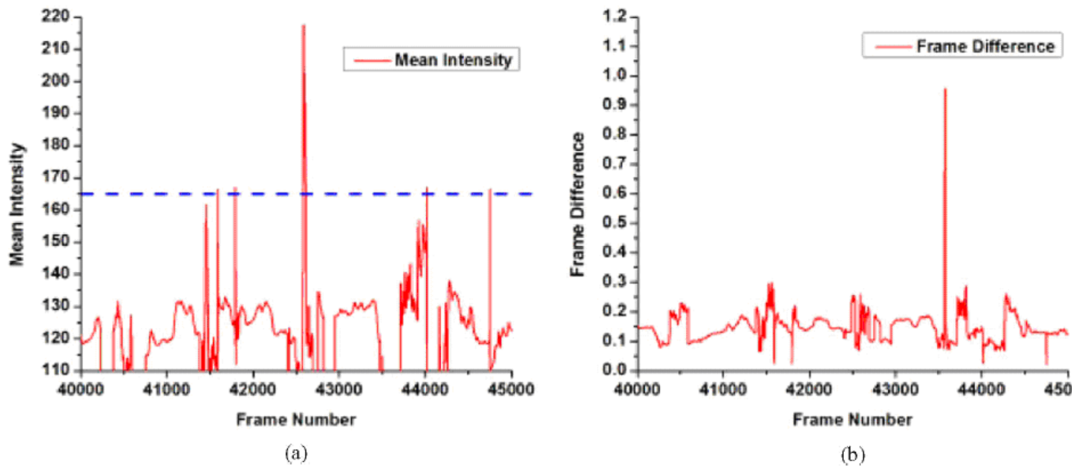


Fig. 2.15 Matching process.

(a) Mean intensity of some video frames, (b) Results after frame difference

2.4.4. Pairing logos

Finally, to detect the replay, a pairing method is used, but in this case it introduces some restrictions to improve robustness. When a logo transition is found the algorithm looks for some new transition in the following 2000 frames. If another transition is found a replay will be detected, otherwise it will be is indicated as a false detection. Moreover, the candidates must verify another constraint. Its luminance value should be included between two thresholds computed using the k-mean centroid selected previously.

2.4.5. Results

The experiment is carried out with halves of four games. The total length of test videos is about 180 minutes consisting of different leagues, including one European friendly game, one Spanish Soccer league, and two UEFA games. The ground truth of logo sequences and replay shot is labeled manually for each data set. The table 2.4 shows the results obtained.

Video cases	# total logos	# logos detected	# correct logos detected	# Total replays	# replays detected	# replays detected
England vs France	32	32	31	16	16	16
Barcelona vs Arsenal	36	36	35	18	18	18
Man.United vs Bay.Munich	62	62	60	31	30	29
Real Madrid vs Barcelona	60	57	57	30	27	27
Totals	190	187	183	95	91	90

Table 2.4 Test results of logo detection

The results presented by this method are the best ones of all papers exposed. About logo detection, the recall and precision values is 96.31% and 97.86% respectively. Finally, the success of replay detections is 94.73% for the recall and 98.9% for precision.

Chapter 3 Replay Detection based on automatic logo transition detection and template extraction

Once the main goal has been precisely defined, and knowing about recent research conducted in the video replay detection topic, this chapter is focussed towards new proposals, variants, experiments and partial results implemented from an starting point to the final solution.

The methodology followed corresponds to a trial and error method introducing improvements or changing directly some parts from the initial proposal. So along this chapter you could find all these justifications attaching the results of the experimentation. The data used for testing has been extracted directly from the TVC database, and converted from the DVCPRO25 format to MP4 through the FFMPEG libraries, and moreover decreasing the resolution to 360x288 pixels.

All the video used could be found on the APPENDIX A.

3.1 First proposal adapting the methodology seen on the state of the art

As the starting point, the first approach for solving the problem was to implement some of the methods presented in the state of the art section chapter.

The problem was divided again into four steps: First, to generate the logo candidate set, then to select the logo template among all the candidates. Once the logo template is chosen the next problem is matching the logos with the video contents, and finally to pair them as replays.

For faster processing, at the beginning of the implementation two videos were created from pieces of complete videos from the TVC database. Then, when the system worked correctly for these cases the real videos were tested. The figure 3.1 shows the chain of steps represented as big blocks.

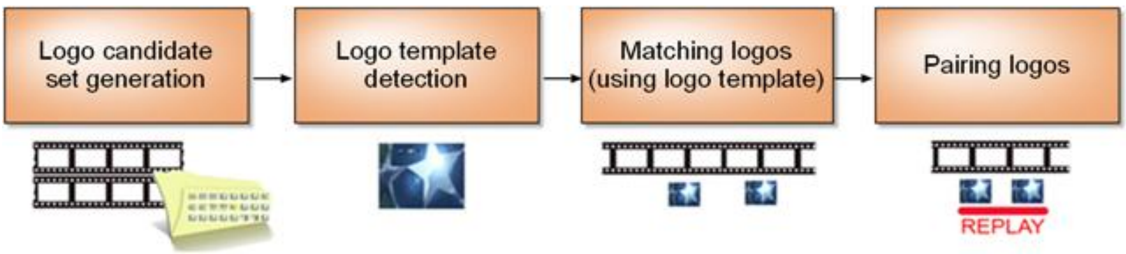


Fig. 3.1 Algorithm division into modules

3.1.1 Logo candidate set generation

As a first goal, the idea was to achive a set of candidates for the logo template. Therefore, applying the formulas (2.1) and (2.2) seen before the model computes the difference among adjacent histogram bins for each frame. The figure 3.2 shows on the left the histogram of the first frame of the video and on the right the accumulated bin differences of all the histograms of the video.

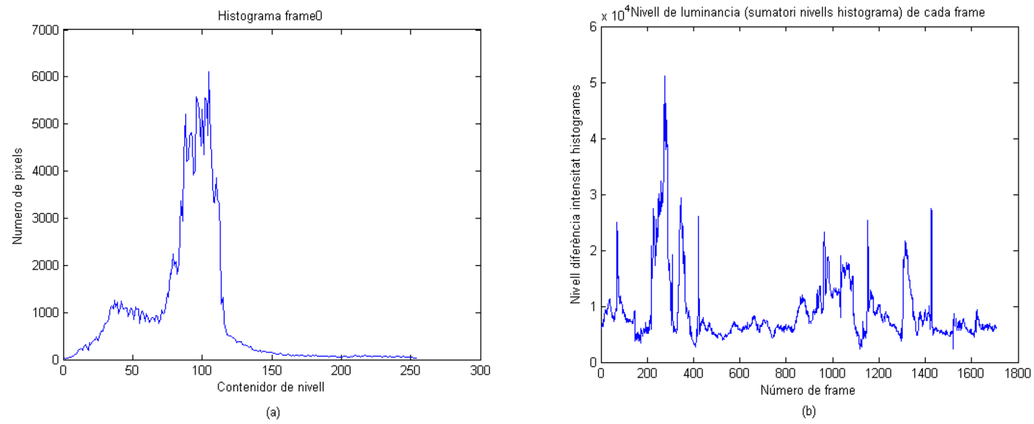


Fig. 3.2 Logo candidate set generation process (I)

(a) Histogram first frame of the video,
(b) Accumulated bin differences of all video frames histograms

Then, it computes the intensity frame difference between adjacent frames (Fig.3.3-a), and finally accumulates these differences in windows of 20 positions (Fig.3.3-b).

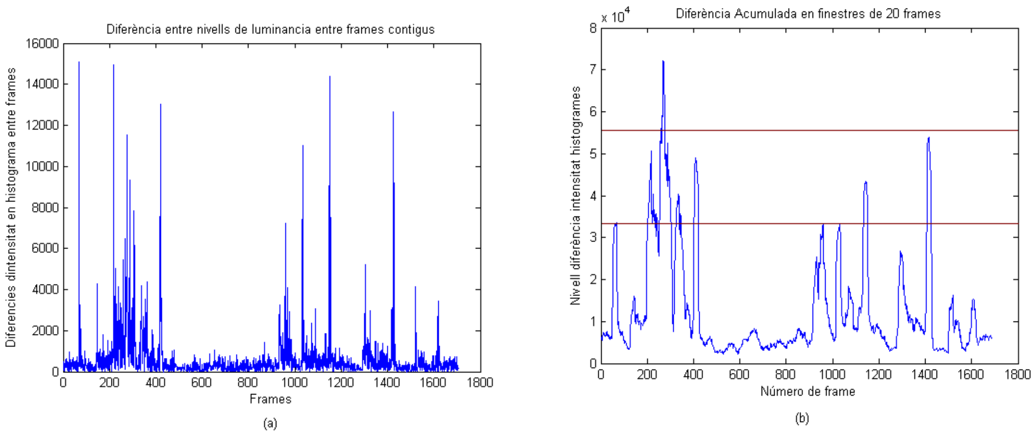


Fig. 3.3 Logo candidate set generation process (II)

(a) Frame difference of all video frames,
(b) Accumulated difference in windows of 20frames of all video frames

The candidate set results from filtering the last graph using two experimentally thresholds, in this case 35.000 and 55.000. So, at the end, the candidate set was around 70 frames from the 1700 frames of the complete video following the methodology exactly the methodology proposed on [11].

3.1.2 Logo template detection

At this point, one frame of the candidate set must be selected as a logo template. The mean luminance of each candidate was computed using the formula (2.3) and taking these values, the K-means process grouped similar frames in two different clusters. Finally, the cluster with higher centroid value was selected as the Final Candidate Set.

Next, computing the distance matrix among all them using the formula (2.4) the final logo template was chosen. In this case, the template detected is shown on the right of the figure 3.4.

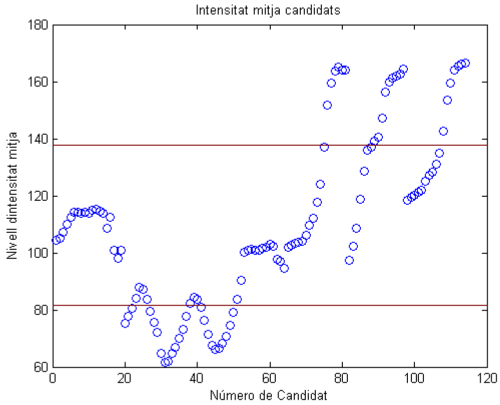


Fig. 3.4 Logo template detection process
Mean luminance of all the frames in the Candidate Set and
Logo template detected after K-mean process and distance matrix computation

3.1.3 Match logo with logo template

Once the logo template was chosen, a matching process was implemented with all video frames to find logo transitions.

The mean luminance values were calculated for the logo template and for all the frames of the video. Realizing subtractions all them were compared with the template. If the value was below a new threshold of coincidence, the frame was marked as logo transition.

This process produces a high number of contiguous frames which are marked as detections. So, to reduce this number to just the most similar frame in the transition, the threshold tolerance was reduced.

3.1.3.1 Solving Threshold problems

The reduction of the threshold tolerance seemed a good measure adopted at the beginning, but for logos that not full fit completely the frame, containing another randomly part, this solution instead of solving the problem, generated a new one: losses of logo detections. The figure 3.5 shows an example. The logo at the left does not present any problem in contrast with right logo where only the central part is invariant.

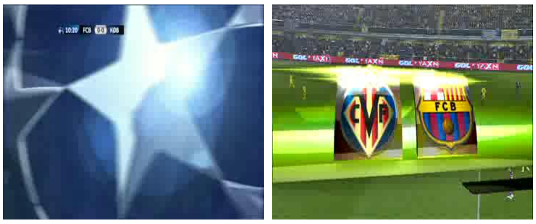


Fig. 3.5 Logo template examples

So, fixing a low tolerance value was not a good solution and a second strategy was applied to solve the problem with a minor modifications.

Instead of marking as a logo transition all frames that exceed the threshold, only the first one is chosen. Then, taking advantage of the characteristic length of the logo transitions that are never longer than 30 frames, the algorithm skips this number of frames before continue the matching. This improvement allows detecting only one representative frame for each logo transition, solving possible problems in the next step: pairing logos.

3.1.4 Pairing logos

Finally, last step consists into pairing the logo transitions detected taking into account the constraints explained in the previous chapter. So, two new experimentally thresholds were computed and if both values were exceeded by a couple of logo transitions they were marked as a replay.

3.2 Preliminary results

Once the theoretical model was implemented was time for testing and improvements. All the methodology described was tested using the two short videos composed by selected shots from two complete videos of the TVC database (videos A.7 and A.8). Moreover, a second test was realized with the full videos (videos A.1 and A.4). The figure 3.6 shows the two different logo templates found for the first test at the left and for the second test at the right.

Fig. 3.6 Logo templates found for the videos A.7, A.8 and A.1, A.4 respectively

Although the left image of the first couple was correct, its neighbour image was wrong. Furthermore, the second couple of images were incorrect both. So, it was clear that at least the logo template detection method of the algorithm needed to be improved. The best way to fine-tuning the method was treating the different steps seen inside the algorithm as independent modules. Thereby, any modification did not affect the other parts.

3.3 Improving the results: Trial and Error methodology

The following section explains the methodologies that have been implemented in the final algorithm. All the other methods and developments analysed and performed without successful results are detailed on the APPENDIX B at the end of this document.

3.3.1 Logo candidate set generation

The next improvements are focused towards one of the most important steps: to get the best possible images in the Candidate Set. The videos used correspond to the same videos used on the first proposal. The figure 3.7 indicates the module which is being improved.

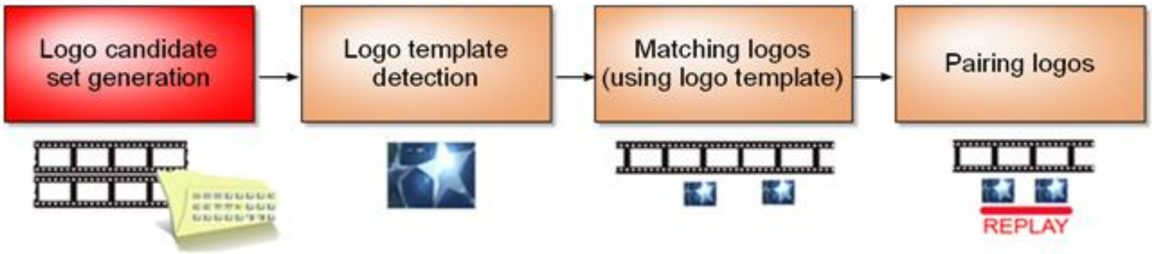


Fig. 3.7 Improvement of the first module

From the model seen above, the intention was to increase the number of positive candidates in the group of candidates.

The first modification (detailed in **Using RGB components instead of Gray**[B.2]) was to design an adaptive threshold instead of setting a default value to filter candidates. This did not improve the process and just filtered both good and bad candidates.

The second proposal (detailed in the section 2 of the APPENDIX B) was to use colour histograms instead of the gray histogram. This emphasized the peaks and highlight new peaks which had been masked. This increases number of positive candidates, but the group was too large to find the template.

To reduce the size, the idea (detailed in the section 3 of the APPENDIX B) was to not perform the step of accumulating the difference of intensities in windows of 20 frames, so only the peaks remain outstanding and not all the frames around them. However, the number of candidates was too high.

At this point, the solution was divided in two branches:

- First, was to investigate if it was possible to implement a method much faster with at least the same positive results inside the candidate set. This led to raise another system to detect candidates: the Shot Boundary detection method (explained below).
- The second was to improve next modules to distinguish between correct frames and false positives in the candidate set.

3.3.1.1 Shot Boundary detection method

Now, the main goal was to look for some method able to reduce the candidate set maintaining the correct results obtained until the moment and, moreover, reducing the computational cost. So, the solution adopted focuses the problem into getting the most relevant frames from the video and only working with them.

These frames were those corresponding to important visual changes in the video. For instance, when camera changes from close-up view to a long shot, or when a logo transition appears.

First changes were easily detectable because from previous frame to the next became usually a big variation, but for logo transition was not as easy. For example, logo transition could appear in slowly from one side of the frame to the other, or it could increase its size and decrease, etc. So, the method should detect when images were really changing or not.

The method implemented was extracted from the paper: Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing. [13]

The goal of this algorithm is to detect gradual transitions in videos in real time, and presents experimental results around 95% of precision and recall. So, it seemed a good alternative to acquire the candidates.

The main idea is to define a window composed of N frames, make a matrix of it and compute the Singular Value Descomposition (SVD). Then, it normalizes its eigenvalues with respect the largest one and traces the rank checking the eigenvalues that surpass a defined threshold.

At the end, the system decides if it is a shot boundary looking at the variation of the rank from the previous frame to the current.

It follows next rules:

- The rank is maximum at the start of a shot.
- If the rank(frameX) > rank(frameX-1), then the visual content of the current video frame is sufficiently different than the content of the previous frame.
- Else, if the rank(frameX) < rank(frameX-1), then the visual content of the video has been stable long enough to “forget” the previous shot and/or the digital effects during the transition.

So, if the rank(frameX) > rank(frameX-1) and the rank(frameX-1) = 1(the minimum value) that should be the end of the gradual transition.

The figure 3.8 shows the rank variation of a gradual transition. The transition increases the rank value, and when it ends, the rank recover the starting value.

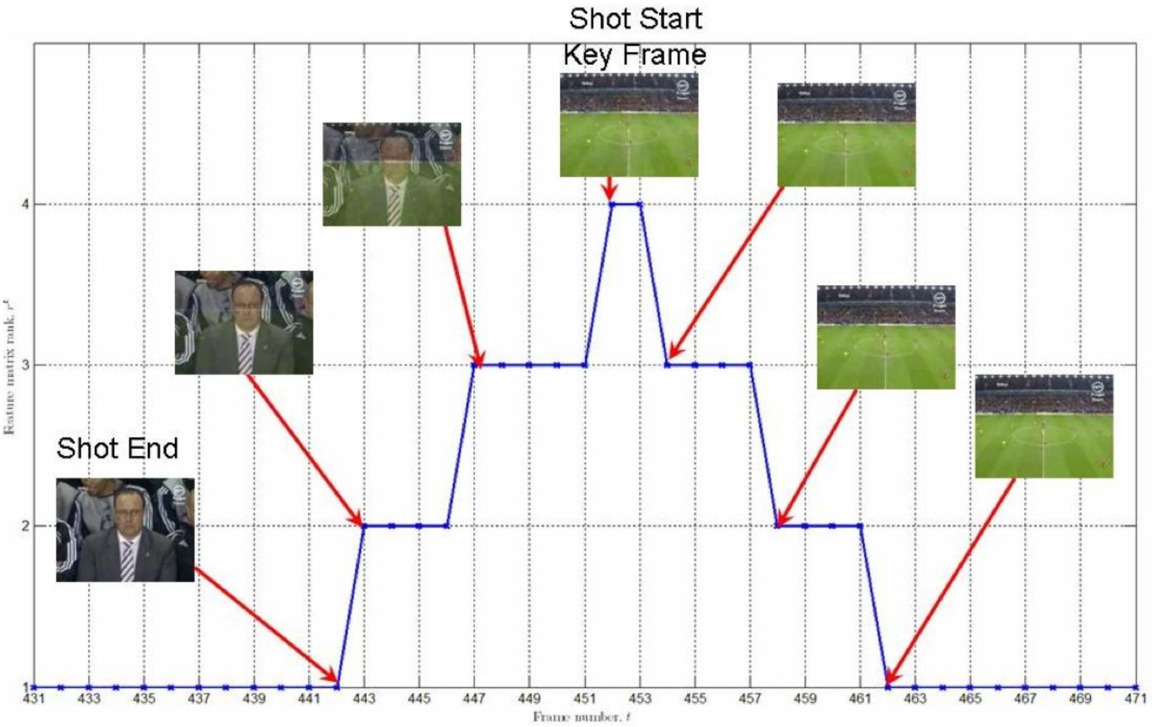


Fig. 3.8 Rank values for a gradual transition

Using this new methodology to detect transitions the results improved considerably.

The process reduced the computational time around 3 times the previous method. Furthermore, the candidate set contained all logo transitions among other candidates.

The principal advantage of this method was the improvement on adaptability to detect changes, being more robust than the adaptable thresholds method that some times, depending on the video, propose not optimized thresholds. So, finally the candidate set contained all desired frames.

In contrast, the number of candidates in the set was still too high. So, consequently, wanting to use this methodology, next step was to improve the template selection mechanisms.

3.3.2 Logo template detection

The second module was focused on extracting the logo template from the candidate set and it was the most difficult part of the algorithm. The figure 3.9 indicates the module which is being improved.

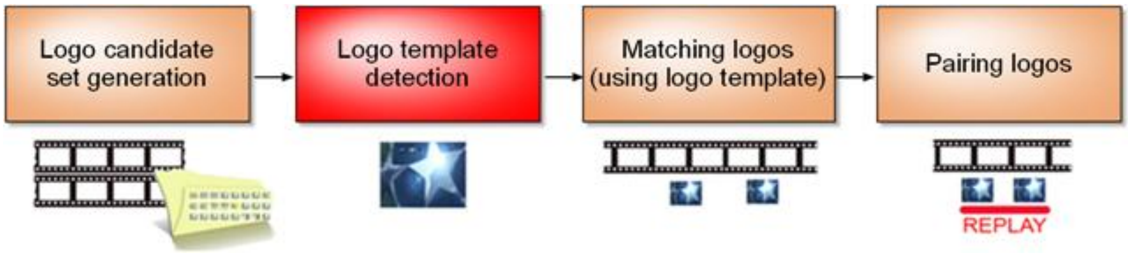


Fig. 3.9 Improvement of the second module

The previous step reduced considerably most of video frames, getting the most interesting ones. Now the main goal was to differentiate among the images that really were a logo candidate and those were anything else. So, the idea was to filter the candidate set taking into account the most relevant features of the logos.

3.3.2.1 Logo features

A set of characteristics were studied to determine which one was most relevant or representative from the others to use them in the filtering process. The figure 3.10 shows schematically some features associated to an image.

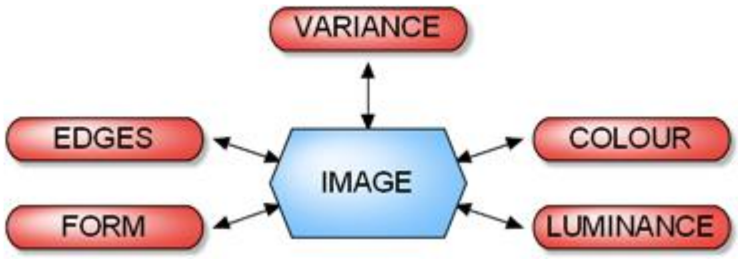


Fig. 3.10 Features of an image

3.3.2.1.1 Luminance

Normally, the logo has been created by graphical designers. So, the range of colors used is much more extensive than other real images found in a football stadium. Furthermore, many times, the logo template corresponds to the sponsor, like BBVA for "La Liga" in this current year, or the television ID, and it must be especially highlighted.

So, the luminance attached to a logo frame in the most of cases is higher than the mean luminance of the video.

3.3.2.1.2 Variance

The Variance is another parameter to take into account in logos. One of the objectives to include a logo transition between two shots, apart from highlight the moment, is to do a clean concatenation among the video broadcasting and the replay shot. It means that, while the logo transition is displayed filling the most part of the frame, the replay shot comes in, without seeing any sudden change other than the transition.

So, during this transition the logo usually fills the entire frame, and it means that these frames decrease its variance, at least, smaller than the mean variance of the whole video.

3.3.2.1.3 Colour, edges and form

The logos are always different depending on the football match. So, the color, edges and form are factors that cannot be taken into account.

However, the green component of the images could be relevant because it is useful to distinguish if the image probably becomes from a far shot containing majority soccer field or from a middle or close shot. So, it was used to make some tests that finally have not been included.

3.3.2.2 Candidate Set Filtering

Last point was useful to corroborate that luminance and variance were common factors on logos although the scalability of possible values.

So, a K-mean process was performed several times using the mean luminance and the mean variance of the images. The results were not good enough and the problem was raised again. (This method is detailed in the section 4 on the APPENDIX B)

Finally, a filter based in two thresholds was implemented following these rules:

- The first threshold limits the luminance to be at least 1/3 upper than the mean of the video.
- The second threshold limits the variance to be at least 1/3 lower than the mean of the video.
- And these two conditions should be achieved at same time.

All frames verifying these conditions formed the final candidate set.

Using this strategy, the results were successfully. The reduction of frames in the candidate set was very considerable. For instance, a candidate set around 30.000 candidates was reduced to 2000 candidates, or another example, a case with 22.000 candidates was reduced finally to 1200. Moreover, the resulting candidate set contained most of the logos got before filtering.

However, 2000 frames were too much for computing directly the differences matrix and guarantee successful results. So, another dividing step was needed to put together many candidates as possible.

The idea was to apply the k-means algorithm and to choose the group containing the logos. Now, applying k-mean successively was not required. Only dividing the candidate set once into an adequate number of clusters was enough.

3.3.2.3 K-mean process using 12 Centroids

At this point, the last subdivision should be done using the reduced candidate set achieved after filtering.

The features used until now to filter and reduce the candidate set were basically luminance and variance, so, all candidates shared similar values although be clearly different at a glance.

This fact shows that if finally the idea was to separate the candidates into groups of similar images, another factor should be took into account: the colour. But now, just divide the candidates in two groups was not enough to select after the template. Therefore, more than two groups should be created. After some tests the number was fixed into 12 centres, high enough to have one group composed basically by logos among the others.

To use the colour in the k-mean process, the RGB histograms were computed for each candidate. Computing the mean value of these histograms, each component was represented using only one value such a pixel. Then, only two of these values were considered, blue and red, and were subtracted respectively (blue - red) obtaining the value to take into account for the image.

The figure 3.11 shows an hypothetical example. So, if the highest value was the red, the result should be negative. In contrast, if the highest was the blue, the value should take positive values. Finally, if the both were similar, the result should be small and the value could adopt positive or negatives values (around the zero).

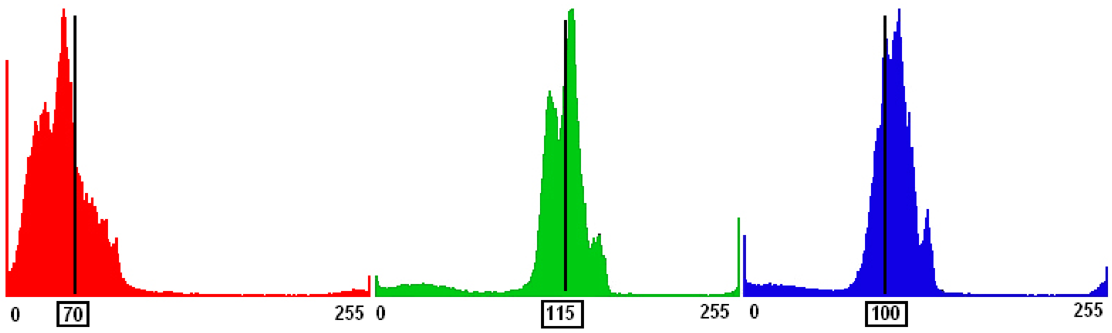


Fig. 3.11 Representative values of the RGB components of an image

Using these configurations the k-mean algorithm was performed, and then, the difference matrix selected the representative image of each cluster. The results were successful in most of the cases. A set of 12 images were proposed as final templates and among them, at least one corresponds to a logo.

So, the last step consists in choosing one of these figures as the final logo template. The figure 3.12 shows the final candidates found for one of the test videos from the TVC database. As could be seen the templates found represent the principal colours of the filtered candidate set. They were images representing the gray (which corresponds to the logo transitions), blue (players, banners and the sky), yellow (the referee) and green (the field).

Fig. 3.12 Final candidates to be logo template of the video A.2

3.3.2.4 Choosing the final logo template from 12 candidates

Finally, having 12 candidates, the last step consists into implement the methodology to choose the final logo template among them.

Analysing the groups, some conclusions were found:

- All the logos were placed at least in one group.
- If during the logo transition its frames change a lot they could be placed on different groups. So, it is possible to found more than one logo in the 12 final candidates.
- The division by colour of the groups made their respective images similar. But, those having the logos were much closer among them.
- The number of candidates per group was different for each one, but some of them have really few candidates inside.

So, taking into account these conclusions leads to establish the next two rules as the choosing criterion:

- First, the number of frames per group was normalized with respect the highest. Theoretically, if most of the frames of the candidate set were logos, they should be grouped together, forming one of the highest groups with respect the rest.
- Second, the averages from the difference matrices used to found the 12 candidates were normalized respect the minimum.
  So, the high value represents the image less different inside their candidate set respect all the groups and their respective template.
  Again, theoretically, if one of these 12 groups was composed basically by logos, all their frames should be very similar, and surely one of the most similar respects the others groups.

These two parameters were used to determine the final logo template.

The second rule about similarity inside the group seems logic. But, what happen with the first rule for those cases where the number of logos in the candidate set was very low?

Maybe, the group containing logos had a low number of candidates. Well, theoretically it must not happen, but to consider also these cases, a weighting was applied. So, the second rule doubles its weight with respect the first.

The table 3.1 shows the respective weightings for the 12 final candidates of the video A.2 (Getafe C.F versus F.C Barcelona).

Candidate	Mean Difference	Candidates number	Final weighing
1	0.743874	0.263076	0.195695
2	0.786848	0.0429352	0.0337835
3	0.741879	0.0374707	0.0277987
4	0.797049	1	**0.797049**
5	0.751512	0.12178	0.0915191
6	1.05997	0.132709	0.140668
7	0.54596	0.0640125	0.0349482
8	0.609314	0.0811866	0.0494681
9	0.696721	0.0491803	0.034265
10	2	0.0132709	0.0265418
11	0.674605	0.0039032	0.00263312
12	0.861056	0.143638	0.12368

Table 3.1 Weightings of the video A.2

As could be seen, the second column contains the normalized values of the mean difference of each group. It has a double weighing, that means that its high number takes a 2 instead of the 1. The third column shows the number of candidates per group normalized but without any scale factor. Finally, the last column presents the final weighing for each respective final candidate. On this example, the image selected as logo template was the fourth (Fig 3.13).

Fig. 3.13 Logo template detected of the video A.2

Even so, this methodology was not good enough for all the test videos proposed. The reason was the robustness of this method, fixing the number of centers to 12, without considering the possibility of separating them into more closing groups. Theoretically, the image to select as a logo template must appear several times in the video. So, they must form a set of images almost identical.

If instead of fixing the number of centers to 12 in the k-mean algorithm, it could increase depending on the similarity of the own candidates, and the threshold that determines this division was really high, one of the groups should be bigger than the others containing all this almost identical logos.

So, this reasoning opens the mind to news approaches for the candidate set generation process. Unfortunately it supposes to start a new line of research and experimentation which maybe would give rise to a new project associated with the current research. Otherwise, TVC has on his possession investigations and projects with some relation with the new idea presented. One of these projects is named **hierarchical cellular tree** abbreviated as HCT [14] and has been tested to prove some of the hypothesis raised.

This software is able to index all the images from a database using standardized visual descriptors offering after the possibility to do queries using an image of reference. The program builds a tree grouping similar images from the database according to the descriptors chosen. For instance, possible descriptors to select are the Color Layout, or the Color Structure. [15]

Then, with the tree build, the system accepts image requests moving over its branches in a fast way to give the most similar images to the requested. So, part of this software seems a good approach to auto generate groups of candidates, and look for correspondence among his members.

Two of the test videos were tested using this software. Their candidate sets have been used as databases. Then, the images were indexed according the Color Layout[10] descriptor first and the Color Structure[11] next.

The figure 3.14 shows the tree built using the candidate set of one of the TVC videos using the Color Layout descriptor.

As could be seen, each node color represents a level on a branch. So, from them hang other nodes until arrive to the yellow ones which represent the end points of the branches or groups of candidates. The red nodes are those groups containing logos. All they are placed on the same branch, and moreover, the figure 3.15 shows that are the most populated branches of the tree (node 53).

So, everything seems to indicate that the previous hypothesis of using mean differences and number of candidates could be a good strategy using this new grouping methodology.

---

[10] Color Layout (**CL**) is designed to capture the spatial distribution of colour in an image. The feature extraction process consists of two parts; grid based representative colour selection and discrete cosine transform with quantization.

[11] Color Structure (CS) encodes information about the spatial structure of the colours in an image as well as their frequency of occurrence.

Anyway, as it has been said, taking this approach carries a large process of improvements again, and it is just proposed as a possible solution to improve the current system in future works.

Furthermore, this solution is only applicable academically, because the time needed to export the candidate set, indexing their images and to built the tree is in the order of hours, and one of the goals of this project is to help to save time in postproduction.

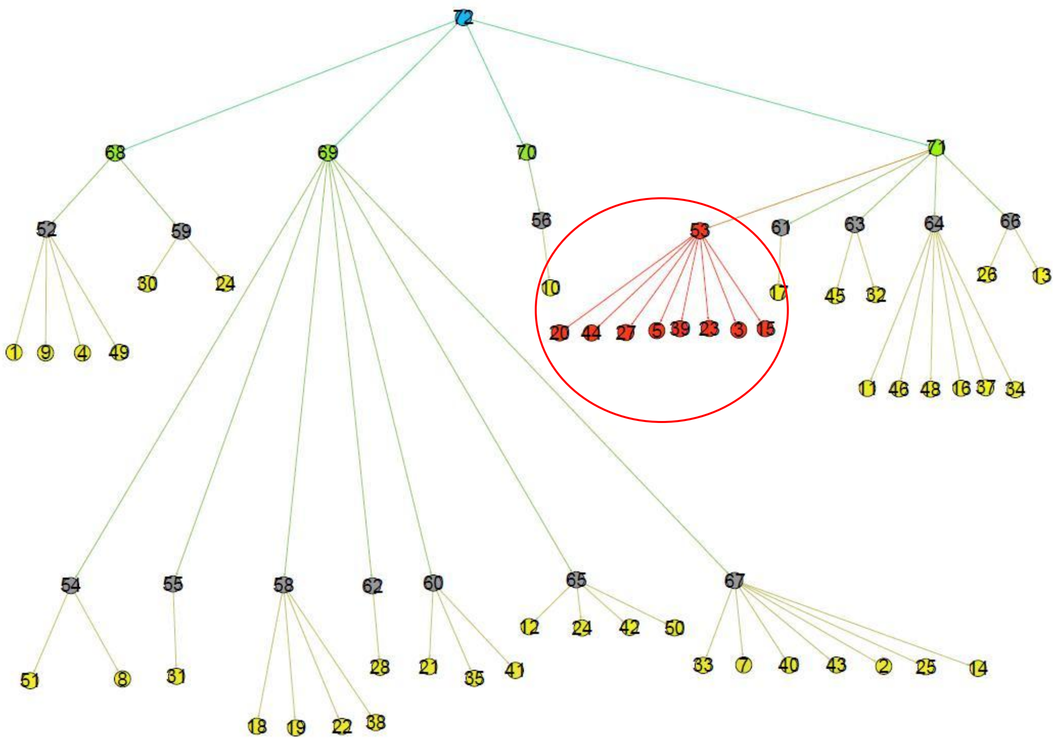


Fig. 3.14 Color Layout tree of the video A.2

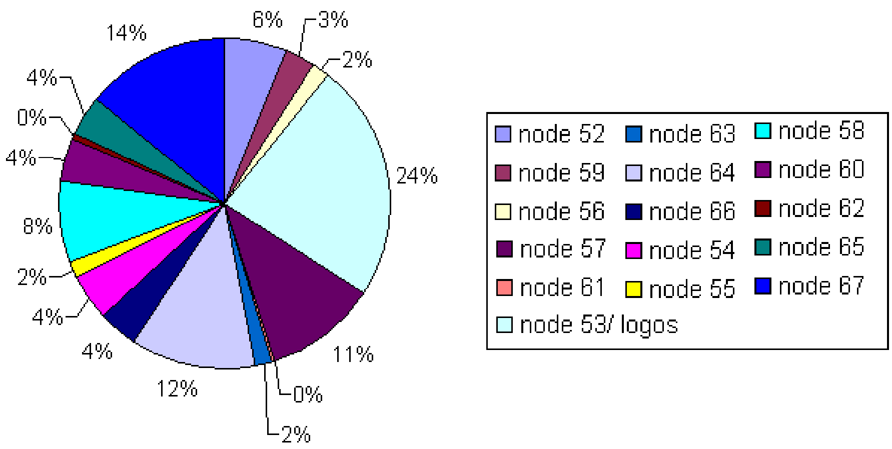


Fig. 3.15 Percentage of frames contained of ground level nodes.

3.3.3 Match logo with logo template

This module basically uses the logo template detected the step before for matching logo transitions along the video. The figure 3.16 indicates the module which is being improved.

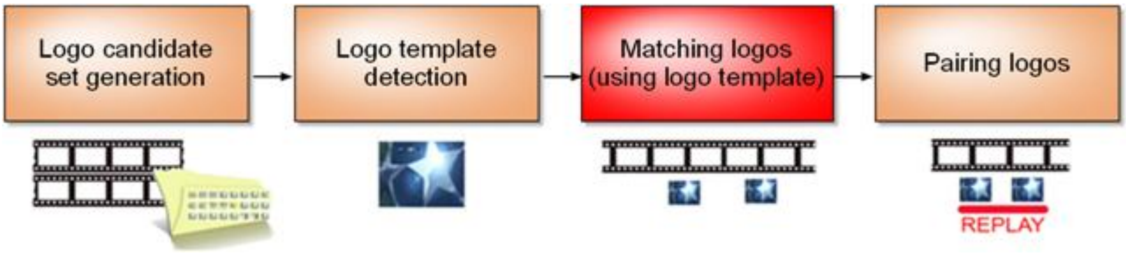


Fig. 3.16 Improvement of the third module

The method described on the proposal realizes a mean luminance comparison between frames and the template using a minimum threshold of verification. In fact, this methodology was not enough robust because it was comparing numbers, and the results can verify numerically the threshold although being a false case. So, the matching was implemented using another methodology.

3.3.3.1 Matching using frame to frame pixel difference

The method implemented is basically to use the logo template as a pattern. First, the template is converted from RGB to Gray and then it is considered as a matrix of values where each pixel varies from 0 to 255. Then, the same is done for all the frames of the video.

The process consists into subtract each time both matrices (M(i) and M(j)) and to check if the summation of the differences between them is less than a threshold computed previously (formula 3.1). This threshold is fixed as the 20% resulting of accumulate the values of the matrix corresponding to the template M(i).

$$\frac{\sum_{n=1}^{\text{numpixels}} M(i)}{\sum_{n=1}^{\text{numpixels}} M(j)} < 0.2 \sum_{n=1}^{\text{numpixels}} M(i)$$

(3.1)

The successfully frames were marked as logo transitions found. This method returns only almost identical images and therefore reduces the false positives.

The method had successful results for those logo templates that fill the entire frame because were invariant along all the frames of the video. But, for those logo templates which just full fits some part of the frame appear the problem of the background changing.

When two frames were compared, the matrices corresponding to their images were subtracted respectively, position by position. So, the pixel on the frame not covered by the template supposed in some cases too much difference.

The figure 3.17 shows a problematic logo template on the left against another without any problem on the right and the following section explains how the problem was solved.

Fig. 3.17 Logos to use as template

3.3.3.2 Fixing an area of interest

All logo transitions have in common that the logo is moved around the centre of the frame. So, to solve the problem of background changing, an area of interest was defined over the frames and the template.

The division was proportional following the same aspect of a matrix 3x3 and the area of interest correspond to the centre. Therefore, the background changing problem disappears and the threshold of similarity to accomplish could be maintained restrictive, improving the results of the matching. The figure 3.18 shows the new area of interest.

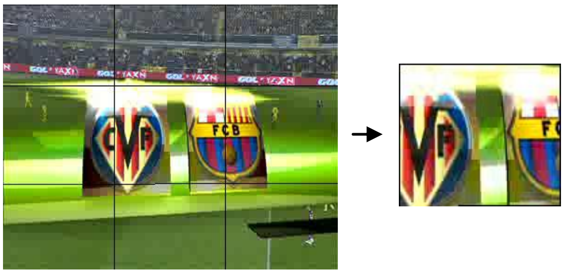


Fig. 3.18 Selecting an area of interest

3.3.4 Pairing logos

Finally, the last module also could be improved. The starting proposal suggests a pairing methodology with some constraints in position and the luminance value to discard false positives. The figure 3.19 indicates the module which is being improved.

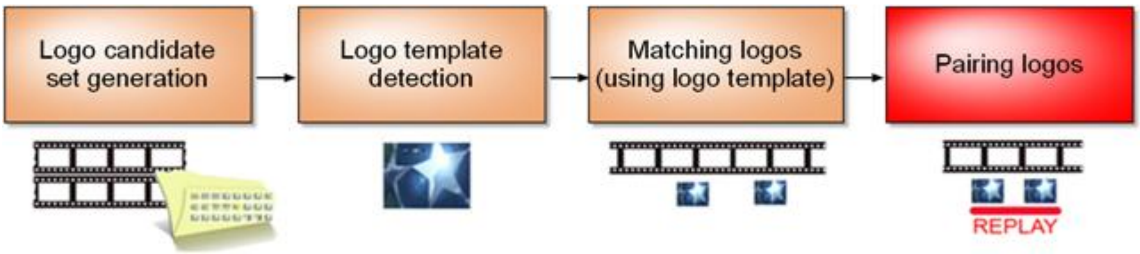


Fig. 3.19 Improvement of the fourth module

This method got successfully results for replays clearly far on time and for videos with an excellent broadcasting production. But, when a set of replays were placed closer on time or when some false detection appears the method often fails. So, a new methodology has been proposed for pairing detections taking into account additional parameters such as the proximity among detections.

3.3.4.1 Adaptable pairing

The first important difference introduced by the new method is that before to start assigning pairs, a first glance was done over the detections. For an easiest understanding a schematic example has been proposed (Fig 3.20).

Fig. 3.20 Theoretical example proposed

So, after processing the transitions must be pared as the figure 3.21 shows. Green arrows corresponds to the first frame of a replay, and red arrows the last. The yellow squares correspond to false positives detected.

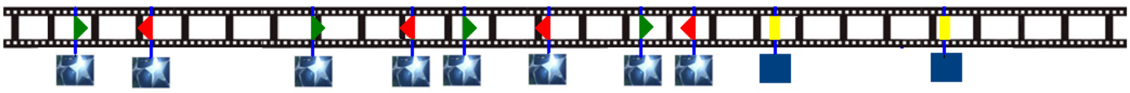


Fig. 3.21 Ideal result desired after pairing transitions

3.3.4.2 Check proximity among transitions

First step is based into check and consider the proximity among transitions. So, for each transition, the 2000 previous and following frames are reviewed.

Then, next considerations are done:

- If only one transition is detected on some of these checked periods, the detection analysed is marked as the starting or the ending point of the replay depending of their respective positions (green or red arrow)
- If more than one transition is found on the same period, only the nearest one is considered.
- If more than one transition is found in both periods, again the nearest one predominates over the other.
- Finally, if nothing is found in both periods the transition is marked as false positive (yellow squares).

The figure 3.22 shows the expected results of the example sequence proposed.

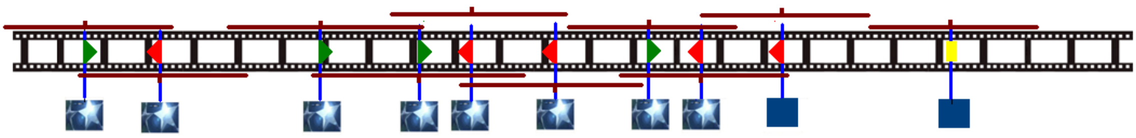


Fig. 3.22 First approach checking proximity

As could be seen, considering only proximity is not enough when a set of closer replays were placed consecutively. So, the system should consider a set of exceptions.

3.3.4.3 Discarding isolated detections as false positives

First exceptions analysed are those transitions that had not any detection around 2000 frames (before and after). All they are considered as false detections. These cases are signalized by yellow squares (Fig 3.23).

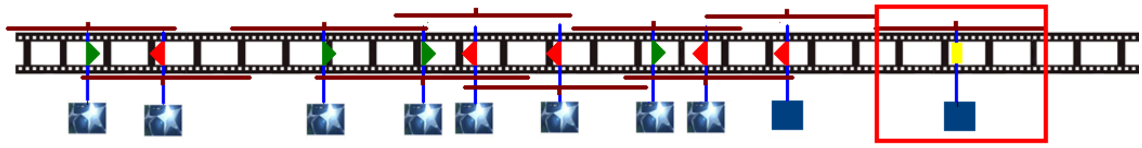


Fig. 3.23 Treating isolated transitions

3.3.4.4 Pairing couples of closer detections

The second exception concerns those pairs of transitions without any other transition found around them in 2000 frames distance. These cases are signalized using two opposite arrows, green and red respectively (Fig 3.24).

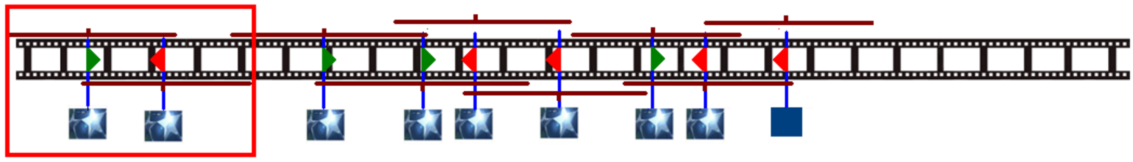


Fig. 3.24 Treating close transition couples

3.3.4.5 Pairing two couples of closer detections

The third exception is a bit more complicated. It concerns the cases where four transitions are closer. When two consecutives replays are placed nearest between them than between their respective transition couples, the arrows direction assigned by proximity are wrong. So, this exception is able to pair correctly these cases.

For that, any more transition has to found around them in 2000 frames distance. It is represented as shows the figure 3.25 and it is corrected and paired as shows the figure 3.26.

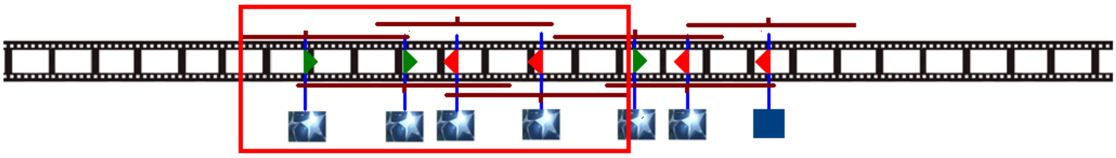


Fig. 3.25 Treating close couples of transitions

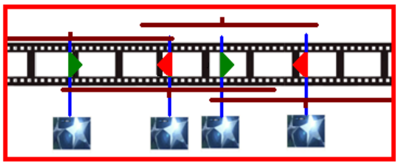


Fig. 3.26 Correcting close couples of transitions

3.3.4.6 Pairing detections for sets of three closer detections

Last exception considered concerns sets of three transitions found very closer where any more transition was found around them in 2000 frames distance.

The method pairs the two nearby transitions, and considers false positives the other one. This criterion has been applied because, theoretically, the number of false positives must be very low due the constraint parameter set before, and moreover, the replays usually have a duration of a few seconds instead of the minute and a half analysed. So, it is more probably that the two nearby detections correspond to the real replay, and the other one, to a false positive.

The figure 3.27 shows an example, and the figure 3.28 shows how it is solved.

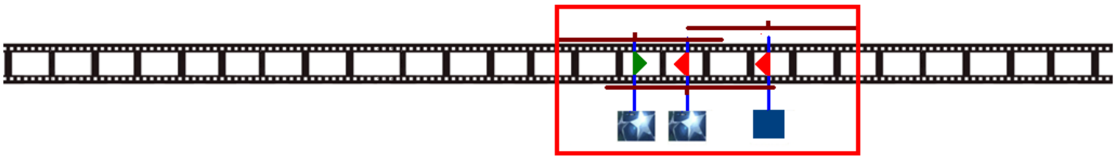


Fig. 3.27 Treating close trios of transitions

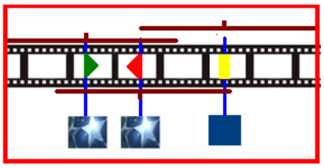


Fig. 3.28 Correcting close trios of transitions

3.3.4.7 Pairing the rest of detections

After these first considerations, the methodology has discarded the majority of cases, but for example, is possible to have more than two closer replays, and these cases are not evaluated yet. So, the system sorts again the arrows of the not discarded detections but consecutively and pairs them under this criterion. So, at the end, all transitions were classified.

This methodology of pairing transitions improves considerably the initial method. However, the best way to guarantee successful pairings is having the least number of false positives. In addition, an excellent production of the video is needed that means without problems of transition missing.

Chapter 4 Implementation of the Replay Detection Application

All methodology seen in the previous chapter takes the form of an algorithm. Along this chapter the main structure which has been implemented in C++ is detailed.

To begin with, an overall block diagram (Fig 4.1) shows the overview and the different steps of the algorithm. Then, the different modes of operation are presented and finally, the resulting XML exported file is described to understand the results achieved.

4.1 Block diagram

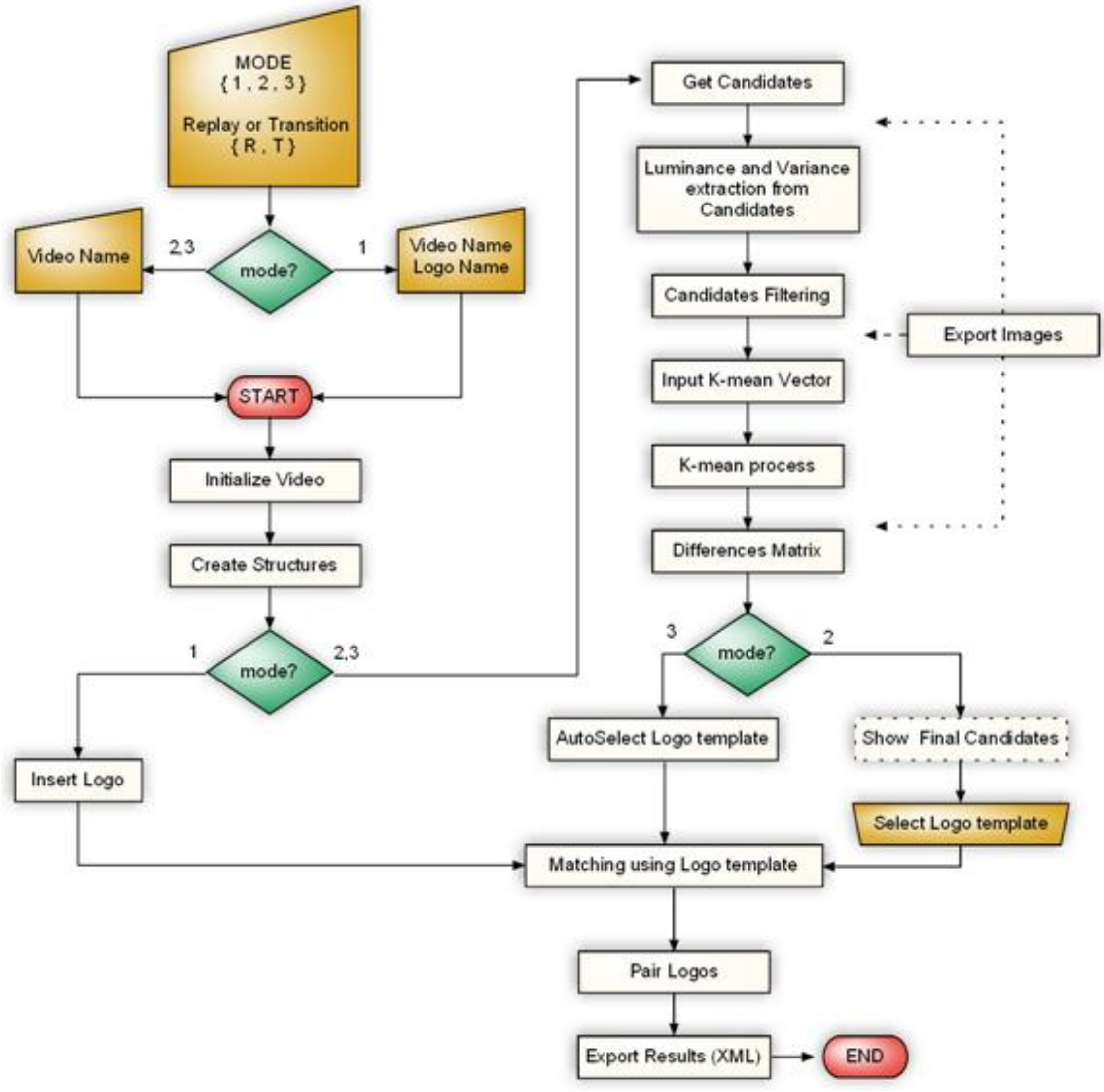


Fig. 4.1 Block diagram of the algorithm

The brown blocks indicate some manual parameter to introduce to the program. Basically, they are next:

- **Mode**: The mode of operation. There are three different possibilities of execution. All they will be described in the paragraphs below.
- **R o T:** Should be indicated what is wanted: to detect replays or just the transitions. This is useful for videos with only input logo transitions that should be treated differently because they could not be paired.
- **VideoName:** This corresponds to the name of the video to analyse. It must be indicated just the name, without *path and extension*. These parameters will be indicated in the configuration file explained below after the operation modes section. The video must present the extension "AVI".
- **LogoName:** This only will be requested if mode 1 is selected. It corresponds to the name of the Logo to use as template. It must be indicated just the name, without *path and extension*. The image must be coded using the 'JPEG' format and the path is not needed because it must be in the same directory as the videos.

The green blocks will review this input data to decide what to do, and finally the white blocks are just different methods implemented that process the video.

4.2 Operation modes

The algorithm is executed using the command line and allows launching batch processes. As the goals said on Chapter 1, the process must be efficient and the design of a graphical interface was not necessary.

It offers two modes of execution:
The first mode launch the application introducing all the data required in the same line as a batch process.
On the other hand, the second mode executes just the application without any input data and an interactive process is performed.

4.2.1 Batch Process

To run the application as a batch process the following data is required:

ReplayDetector.exe R || T Video Logo \\ M \\ A

- **ReplayDetector.exe:** Executable of the application.
- **R || T:** Pairing method:
    - **R:** Detect Replays and pair them.
    - **T:** Detect Transition without pair them.

  Only one parameter has to be chosen, "R" or "T" respectively. This parameter is needed because some replays have a logo transition at the initial point but any transition at the end and this fact could produce erroneous XML at the output. They should be differentiated.
- **Video:** Name of the video to analyse without any Path. It must be indicated on the configuration file explained below.

- **Logo || S || A:** Mode of operation.
  - **Logo:** It will execute mode 1. It must be indicated the name of the logo to use as template without any path and extension because it uses the same as the video indicated in the configuration file. The image of the logo must be included in a folder named LOGOS in the same directory as the video folders.
  - **S:** It will execute mode 2. The character 'M' means Semi-automatic mode.
  - **A:** It will execute mode 3. The character 'A' means Automatic mode.

4.2.2 Interactive Process

To run the application as an interactive process the following files are required:

ReplayDetector.exe

- **ReplayDetector.exe:** Executable of the application.

A main menu will appear offering the three different function modes:

First mode, accessible by putting a "1" through the keyboard, requires putting the name of the video to analyze, the pairing method and finally the image to use as logo template. This is the faster function mode because skips all the steps related to acquire the template, and is focused only on matching the logos using the template, pair them and generate the XML. So, this part will be common in all the modes.

The second mode, accessible by putting a "2" through the keyboard, requires putting the name of the video to analyze and then the pairing method (R or T).
At this moment the blocks sequence seen in the block diagram (Fig.4.1) are performed. First, the algorithm gets the candidates from the video and processes them extracting their respective luminance and variance values. Then, a filter is applied and with the resulting candidates a k-mean process is performed fixing 12 centers.

Next, the difference matrix proposes the final candidate of each group and which is stored on the same figure as a matrix 3x4. The user has to choose which one will be used as logo template. Finally, from this point to the end the algorithm concludes exactly as the mode "1".

The third and last mode, accessible by putting a "3" through the keyboard, requires the same input data as the previous mode. However, the proposal of the logo is fully automatic proposing the final image without any intervention of the user. The figure 4.2 shows the interactive interface using command line. As could be seen, after introducing the function mode, in this case "2" or what is the same "Semi-automatic" and introducing the pairing method "R", the name of the video to analyse is requested and then starts the process. Finally, the results are saved independently generating a new directory which indicates the operation mode used.

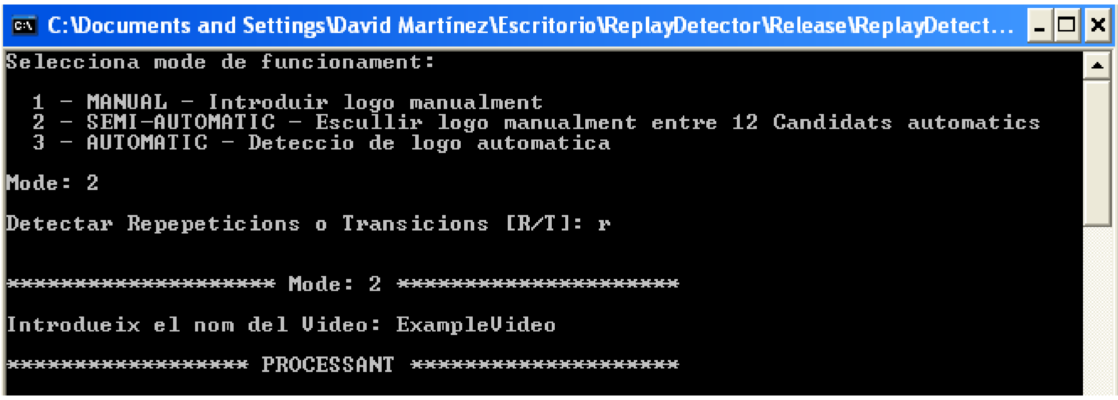


Fig. 4.2 Command line interface

4.3 Configuration File

During the process, the algorithm is able to export the current state of the candidate in each step.

This data normally will not be necessary because only the XML contains the relevant information. However, a configuration file has been implemented to offer the possibility to see the process more detailed.

Furthermore, it contains important parameters to edit if the paths of the data change. The parameters to configure are:

- **Path_dades**: Its value is “D:/BUSCAMEDIA” by default. It corresponds to the Path where the video folders are placed. Each video must be inside a folder sharing name.
- **Candidats**: Its value is “false” by default. It generates a folder containing all candidates before the filtering.
- **Filtrat**: Its value is “false” by default. It generates a folder containing all candidates after the filtering.
- **Agrupacio**: Its value is “false” by default. It generates 12 folders containing all the filtered candidates distributed by the K-mean process.
- **Candidats_finals**: Its value is “false” by default. It saves the 12 representative images from each group.
- **1Imatge_12Candidats**: Its value is “true” by default. It saves the 12 representative images using a common window.
- **Imatge_Seleccionada**: Its value is “true” by default. It saves the final logo template selected.
- **Transicions_i_Replays**: Its value is “true” by default. It generates 2 folders grouping the false logo transitions detected and the positives which are paired as replays.

On the APPENDIX C you can find a configuration example of this file.

4.4 XML Export

At the end, the algorithm generates an XML. The most important tags are those named "nid", "tcin" and "tcout".

- **Nid:** Unique identifier for each replay. Correspond to an incremental number.
- **Tcint:** Starting point of the replay. The format corresponds to hours, minutes, second and frames HH:MM:SS:FF.
- **Tcout**: Ending point of the replay. The format corresponds to hours, minutes, second and frames HH:MM:SS:FF.

The other tags are needed for other TVC proposes and projects, but are not relevant for be explained in this thesis.

So, as could be seen on the figure 4.3, each replay will produce a new entry named "strata" [12] on the xml.

```
<?xml version="1.0" encoding="UTF-8" ?>
 <xml>
   <stratas>
     <strata>
        <nid>0</nid>
        <description>Repeticio del partit</description>
        <strata_id>10001</strata_id>
        <tcin>00:21:33:05</tcin>
        <tcout>00:21:38:12</tcout>
        <tc_origin>00:00:00:00</tc_origin>
        <tcin_absolute>00:00:00:00</tcin_absolute>
        <tcout_absolute>00:00:00:00</tcout_absolute>
        <field_strata_id_pare_value>0</field_strata_id_pare_value>
     </strata>
     <strata>
        <nid>1</nid>
        <description>Repeticio del partit</description>
        <strata_id>10001</strata_id>
        <tcin>00:25:18:34</tcin>
        <tcout>00:25:45:23</tcout>
        <tc_origin>00:00:00:00</tc_origin>
        <tcin_absolute>00:00:00:00</tcin_absolute>
        <tcout_absolute>00:00:00:00</tcout_absolute>
        <field_strata_id_pare_value>0</field_strata_id_pare_value>
     </strata>
   </stratas>
 </xml>
```

Fig. 4.3 XML example

[12] A stratum is the name used for referencing each entry of the XML. So, each asset is described by a set of stratas.

Chapter 5 Experimental results

This chapter shows the final results obtained testing a set of videos from the TVC database. First, the results got by the three function modes for the soccer videos analysed are shown independently. Finally, three Formula 1 videos have been also tested. The features of all these videos are detailed on the APPENDIX A.

5.1 Testing the algorithm using mode 1 (“Manual”)

The first mode takes as input an external logo. The figure 5.1 shows all the logos used respectively. They have been extracted from the test videos using the software VirtualDub[13].

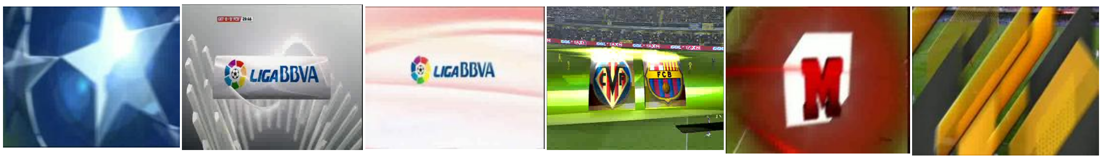


Fig. 5.1 Logos from game 1 to game 6 respectively

The table 5.1 contains the transitions and replays to find and really found for each video respectively using this mode. The totals are computed separately: first division and the others, and finally all them are accumulated to show a unique global value.

Video	# total logos	# logos detected	# correct logos detected	# total replays	# replays detected	# correct replays detected
Video A.1	108	108	108	54	54	54
Video A.2	79	79	79	38	38	38
Video A.3	163	165	163	81	80	80
Video A.4	50	50	50	-	-	-
Totals first division	400	402	400	173	172	172
Video A.5	87	87	87	43	42	42
Video A.6	20	20	20	10	10	10
Totals second/third division	107	107	107	53	52	52
Total	507	509	507	226	224	224

Table 5.1 Transitions and replays detected using mode 1

As could be seen on the table 5.1 the number of false positives and logos missed is very low. So, it seems that this mode works successfully. However, it will be analyzed on the Chapter 6.

[13] Virtualdub is a free video tool for basic editing and encoding, including batch processing, mainly geared to AVI files.

5.2 Testing the algorithm using mode 2 (“Semi-automatic”)

The second mode proposes 12 final candidates to select manually the logo template. The figure 5.2 shows the candidates proposed and the image chosen for each video.

Fig. 5.2 12 final candidates proposed for each video using mode 2 and template selected manually

The table 5.2 shows the results obtained for this second mode.

Video	# total logos	# logos detected	# correct logos detected	# total replays	# replays detected	# correct replays detected
Video A.1	108	108	108	54	54	54
Video A.2	79	79	79	38	38	38
Video A.3	163	182	163	81	88	79
Video A.4	50	50	50	-	-	-
Totals first division	400	419	400	173	180	171
Video A.5	87	32	32	43	6	6
Video A.6	20	1	1	10	0	0
Totals second/third division	107	33	33	53	6	6
Total	507	452	433	226	186	177

Table 5.2 Transitions and replays detected using mode 2

As could be seen, the figure 5.2 shows that this method recovers at least one logo among the final candidates. However, some cases have not a representative candidate enough good to find all the transitions with guarantees.

So, the number of false positives and missed replays increases respect the first mode. Again, it will be analyzed on the Chapter 6.

5.3 Testing the algorithm using mode 3 ("Automatic")

Finally, the third mode was tested. This time, the algorithm decides which candidate use as logo template. The figure 5.3 shows these selections:

Fig. 5.3 12 final candidates proposed for each video using mode 3 and template selected automatically

The results got using the third mode are exactly the same as the second mode for those logo templates selected correctly. So, only the second and third case obtains successfully results. The other cases should be discarded because are not matching the logo desired.

5.4 Testing the algorithm using Formula 1 videos

Finally, the algorithm was tested using three videos from another kind of event in order to check the exportability of the implementation. So, the three modes were tested in a set of Formula 1 videos [A.9] sharing the same logo transitions. The figure 5.4 shows this transition, and highlights the frame used as template.

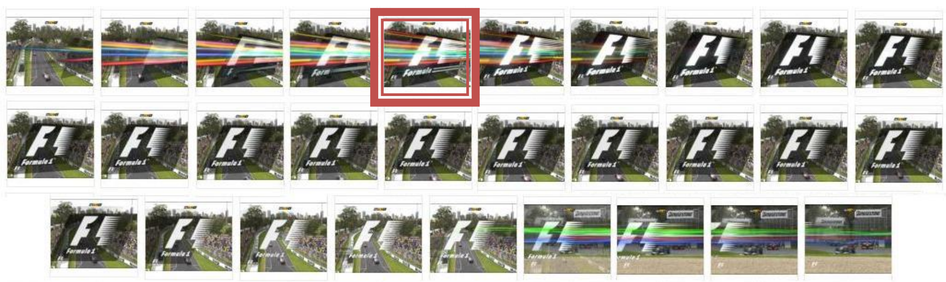


Fig. 5.4 Logo transition used in Formula 1 videos

The table 5.3 shows the results of the first operation mode:

Video	# Total logos	# logos detected	# correct logos detected	# total replays	# replays detected	# correct replays detected
Australian Grand Prix	54	54	54	27	25	25
Catalunya Grand Prix	64	65	64	32	29	27
Monaco Grand Prix	38	39	38	19	17	17
Totals	156	158	156	78	71	69

Table 5.3 Transitions and Replays detected using mode 1

The second mode proposes the sets of candidates shown in the figure 5.5.

Fig. 5.5 12 final candidates proposed for each Formula 1 video using mode 2

Only the second case contains logos among the final candidates, but they are not enough significant to find coincidences offering successful guarantees. So, finally the third mode was not tested.

The three operation modes will be analyzed on the Chapter 6.

Chapter 6 Final conclusions and future work

This chapter concludes the project analyzing the results shown on the Chapter 5. The analysis consists of calculating the values of precision and recall of each mode respectively, and moreover reasoning the cases which show errors.

6.1 Analyzing the soccer videos

As could be seen in the table 5.1 of the first operation mode on the Chapter 5, there are 507 logo transitions to detect in total and 226 replays to find.

So, the first conclusion is that the production concerning to the replays of these videos is not perfect, because each transition must be sandwiched between two logo transitions and so the total has to be even, and it is not. In addition, the number of replays to find not corresponds exactly to the mid of the transition total. The missed replays correspond to transitions not paired due to the distance between them is greater than the maximum distance considered to pair transitions.

The recall and precision values are shown in the table 6.1:

Video	Logos recall	Logos precision	Replays recall	Replays precision
First division	100%	99.5%	99.42%	100%
Second / third division	100%	100%	98.11%	100%
Total	100%	99.60%	99.12%	100%

Table 6.1 Precision and recalls values using mode 1

The matching logo process presents a recall of 100% because all the transitions are found and moreover it has a precision of 99.60% due to two false positives. After the pairing process the recall decrease a bit to 99.12% because it misses two of them but the precision is maximum (100%).

So, the **first operation mode** can be considered that **works successfully** in soccer videos.

On the other hand, the figure 5.2 of the second operation mode on the Chapter 5 shows the final candidates of the different soccer videos.The first four images correspond to the first division games, and next two to the second and third division respectively. As could be seen, the first cases propose at least one acceptable logo among the final candidates. So, choosing them as logo template the results achieved are successful as the mode 1.

The last two cases from second and third division have not the same luck. For instance, as shows the table A.5 on the APPENDIX A, the size of the logo “Marca” is considerably little in most of the transition, especially the frame selected as template. So, the background increases the difficulties of the matching process and the number of wrong detections increase.

The last case has another handicap. It presents a number of replays very low (only 10 replays for more than 2 hours and a half of content). So, the number of transitions is insufficient for propose a better candidate than the chosen. Furthermore, it presents the same problem than the case before, having only three frames during the transition filling completely the frame.

The table 6.2 shows the recall and precision values computed:

Video	Logos recall	Logos precision	Replays recall	Replays precision
First division	100%	95.46%	98.84%	95%
Second / third division	30.84%	100%	11.32%	100%
Total	85.4%	95.8%	78.31%	95.16%

Table 6.2 Precision and recalls values using mode 2

So, the **second mode** can be considered **useful only for the first division** cases, or in those videos having lots of replays filling completely the logo transition frames.

Finally, the third operation mode selects automatically the candidate to use as template. So, the final candidates are the same proposed in the second mode, and if the logo is chosen correctly, the results will be the same explained before in the table 6.2. But, as shows the figure 5.3 on the Chapter 5, it does not happen really with all the cases and only the second and third case select the logo template appropriately.

So, the recall and precision values are not computed for this mode because has not sense. Analyzing the differences features of these videos, the next conclusion is extracted:

The automatic selection of the template follows the weightings criterion explained on the Chapter 3. This takes into account the quantity of candidates in the group and the similarity among them.

The successful selections belong to those cases with long logo transitions having enough frames where the logo is completely filling them.

This way, the mean difference of the group which principally contains the logos decreases due the reduction of background changing. The other cases have not this casuistic, and therefore produce a wrong selection.

So, to improve the results the solution raised on Chapter 3 was to get better groups. Precisely, this is raised as future work in the last section of this chapter.

So the **third mode can not be used ensuring good results for whatever case** yet.

6.2 Analyzing the Formula 1 videos

Respect the formula 1 videos, as could be seen in the table 5.3 (see on Chapter 5), all transitions are detected of a total of 156, and only two of them are false positives. It supposes a recall of 100% and precision of 98.73%. Then, after the pairing process the recalls decreases to 88.46% and the precision is 97.18% (see the table 6.3).

Video	Logos recall	Logos precision	Replays recall	Replays precision
Total Formula 1	100%	98.73%	88.46%	97.18%

Table 6.3 Precision and recalls values using mode 1

The replays missed are those cases with logo transitions too separated in time between them. So, the **first operation mode** can be considered again **successfully** for be applied in F1.

Otherwise, the **second and third modes have not** the same **success**. As shows the figure 5.5 (see on Chapter 5), only one video presents at least one logo from the final candidates proposed, and furthermore, this has again the background problem seen on the final candidates proposed by the second operation mode for second/third division videos of soccer.

So, the final conclusion is that the software can be used in all kind of videos for the first operation mode. But, the second and third mode will be useful only for those videos which have many replays and its logo transitions have a high number of frames where the logo is filling them as much as possible.

6.3 Future Work

The tests done using the HCT software of TVC show an improvement on grouping (see on Chapter 3). The great advantage over the current methodology is the flexibility making these groups. There is no limit of clusters, and each time a new frame differs too much from the others composes a new group.

As shows the red nodes on the figure 3.14 (see on Chapter 3) one of the branches contains all the logos and therefore, theoretically, the method proposed to select the template among the candidates would produce better results.

Furthermore, in the HCT software each frame is described using standardized visual descriptors. On the current version the frames have been classified and filtered using the typical features of an image such as luminance and variance. So, another possible case to study could be using directly the visual descriptors as a filter.

References

[1]. Buscamedia: Cenit Buscamedia. (14 Jun. 2012) [Online]. Available: http://cenitbuscamedia.es/

[2]. Santiago Ontañón, Josep Lluís Arcos, Josep Puyol-Gruart, Eusebio Carasusán, Daniel Giribet, David de la Cruz, Ismel Brito, and Carlos Lopez del Toro, *“GENA: A Case-Based Approach to the Generation of Audio-visual Narratives”,* ICCBR 2012.

[3]. Arnau Raventós Mayoral, *“Descriptors de Moviment per l'anàlisi del contingut audiovisual”*, (25 jul. 2011) [Online].
Available: http://upcommons.upc.edu/pfc/handle/2099.1/12917

[4]. Miscrosoft Visual Studio: (14 Jun. 2012) [Online] Available: http://www.microsoft.com/visualstudio/es-es

[5]. OpenCV (Open Source Computer Vision): Library of programming functions for real time computer vision. (07 jun. 2012) [Online].
Available: http://opencv.willowgarage.com/wiki/

[6]. FFmpeg (7 jun. 2012) [Online]. Avialable: http://ffmpeg.org/

[7]. Zhonghua Dang; Jun Du; Qingming Huang; Shuqiang Jiang, *“Replay detection based on semi-automatic logo template”*, image and graphics, 2007, fourth International Conference, 22-24 Aug. 2007.

[8]. Liu Tingting, Zhang Liangpei, Li Pingxiang, *“REGION SEGMENTATION OF MULTI-SPECTRAL REMOTE SENSING IMAGES BASED ON JSEG ALGORITHM”*, Asian Conference on Remote Sensing Proceedings, 2007.

[9]. Xiaofeng Tong ; Hanqing Lu ; Qingshan Liu ; Hongliang Jin, *“Replay detection in broadcast sports videos”*, Image and Graphics, 2004, Proceedings, Third International Conference, 18-20 Dec. 2004.

[10]. Jinjun Wang; Engsiong Chng; Changsheng Xu, *“Replay detection: scene transition structure analysis”*, Acoustics, Speech, and Signal Processing, 2005. Proceedings (ICASSP '05). IEEE International Conference, 18-23 March 2005.

[11]. Wei Xu and Yang Yi, *“A robust replay detection algorithm for soccer video”*, Signal Processing Letters, IEEE, Sept. 2011.

[12]. Wikipedia, the free encyclopedia, *“K-mean clustering””* (14 Jun. 2012), [Online]. Available Available: http://en.wikipedia.org/wiki/K-means_algorithm

[13]. Abd-Almageed, W., *“Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing”*, ICIP 2008. (15th IEEE International Conference), 12-15 Oct. 2008.

[14]. S. Kiranyaz and M. Gabbouj, *“Hierarchical Cellular Tree: An Efficient Indexing Scheme for Content-Based Retrieval on Multimedia Databases,”* IEEE Transactions, Multimedia, vol. 9, no. 1, January 2007.

[15]. Xavier Giró and Ferran Marqués, *"System Architecture for Indexing Regions in Keyframes",* Poster and Demo Proceedings of the 3rd International Conference on Semantic and Digital Media Technologies (SAMT), Koblenz, Germany. 3rd-5th December 2008.

Acronyms

TVC	Televisió de Catalunya
GENA	Generating audio-visual narratives
FPS	Frames per Second
DCT	Discrete Cosine Transform
CBR	Case-Based reasoning
IEEE	Institute of Electrical and Electronic Engineers
IET	Institution of Engineering and Technology
DPA	Dynamic Programming Algorithm
XML	Extensible Markup Language
FFMPEG	Fast Forward MPEG
LS	Long shot
MS	Mid Shot
CV	Close-up View
SVD	Singular Value Descomposition
HCT	Hierarchical Cellular Tree

Glossary of figures and tables

APPENDIX A. Test Videos

This appendix shows the main characteristics of the videos used from the TVC database to test the algorithm. All they have indicated the length of the media, the transitions and replays contained and a visual display of an input and output logo transition. Usually, both are the same, but in some cases they differ.

The first video (Table A.1) is from the Champions League and the next three (Table A.2, A.3, A.4) are from the Spanish Championship first division "La Liga".

Champions League match (2010-2011): F.C Barcelona vs F.C Copenhagen	
Frames number	208.151 frames
Duration	02h 18min 46s 1frame
Transitions number	108 transitions
Replays number	54 replays
Length logo transition	17 frames
 **Logo transition In**	 **Logo transition out**

Table A.1 Video A.1

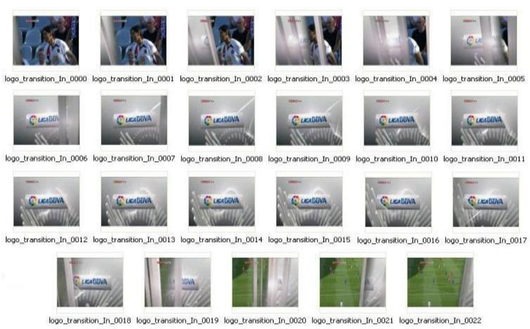
BBVA League match (2009-2010): Getafe C.F vs F.C Barcelona	
Frames number	183.907 frames
Duration	02h 02min 36s 7frames
Transitions number	79 transitions
Replays number	38 replays
Length logo transition	23 frames
 **Logo transition In**	 **Logo transition out**

Table A.2 Video A.2

BBVA League match (2009-2010): F.C Barcelona vs C. Athletico de Madrid	
Frames number	207.208 frames
Duration	02h 18min 08s 8frames
Transitions number	163 transitions
Replays number	81 replays
Length logo transition	16 frames
 **Logo transition In**	 **Logo transition out**

Table A.3 Video A.3

BBVA League match (2009-2010): Villareal C.F vs F.C Barcelona	
Frames number	238.962 frames
Duration	02h 39min 18s 12frames
Transitions number	50 transitions
Replays number	No paired
Length logo transition	21 frames
 **Logo transition In**	- **Logo transition out**

Table A.4 Video A.4

The next two videos (Table A.5, A.6) are from the second and the third divisions of the Spanish Championship respectively, and finally the following two videos (Table A.7, A.8) have been created by me selecting shots from the videos A.1 and A.4, and have been used only for the first tests.

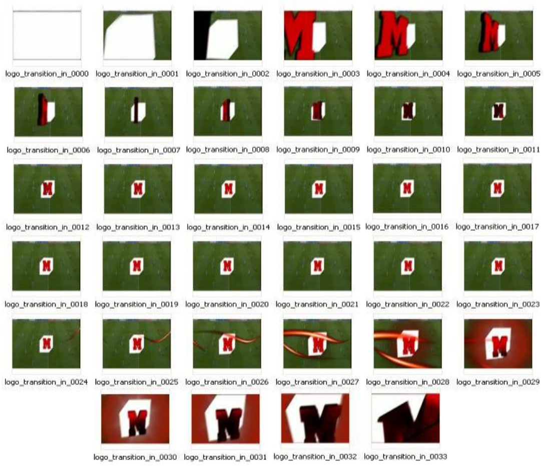
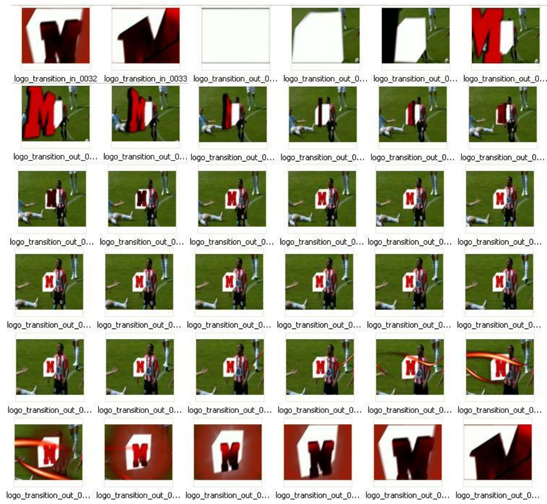
second division match (2009-2010): Celta de Vigo vs Girona F.C	
Frames number	304.496 frames
Duration	03h 22min 59s 20frames
Transitions number	87 transitions
Replays number	43 replays
Length logo transition	36 frames
 **Logo transition In**	 **Logo transition out**

Table A.5 Video A.5

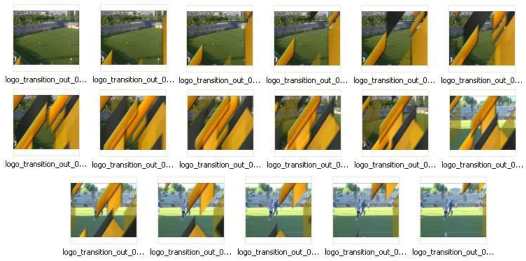
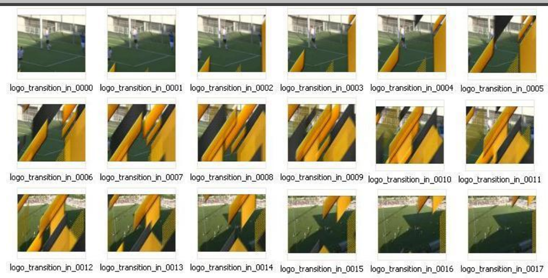
third division match (2009-2010): C.E.Hospitalet vs U.E.Lleida	
Frames number	238.300 frames
Duration	02h 38min 52s
Transitions number	20 transitions
Replays number	10 replays
Length logo transition	23 frames
 **Logo transition In**	 **Logo transition out**

Table A.6 Video A.6

Own video created by the addition of shots from the Video A.1	
Frames number	1.706 frames
Duration	01min 08s 6frames
Transitions number	4 transitions
Replays number	2 replays
Length logo transition	17 frames
 **Logo transition In**	 **Logo transition out**

Table A.7 Video A.7

Own video created by the addition of shots from the Video A.4	
Frames number	1.338 frames
Duration	53s 13frames
Transitions number	5 transitions
Replays number	2 replays
Length logo transition	21 frames
 **Logo transition In**	- **Logo transition out**

Table A.8 Video A.8

The last three videos (Table A.9) are from different races of Formula 1.

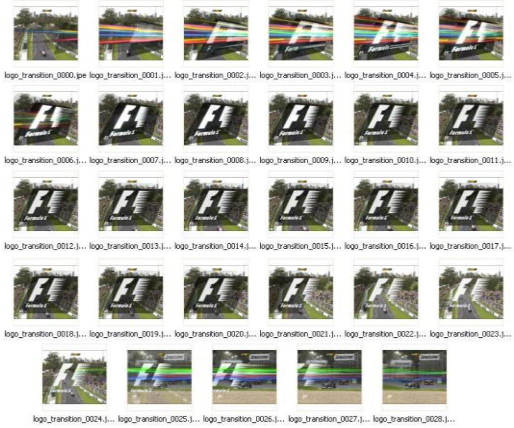
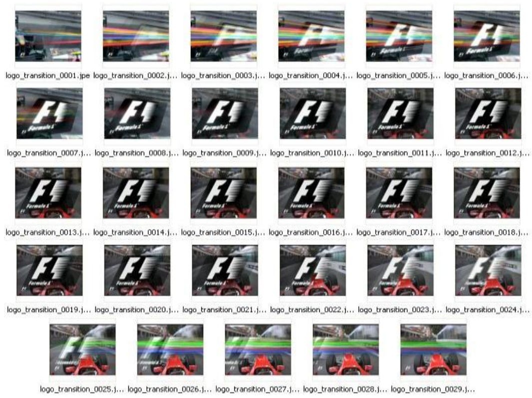
FORMULA 1 (2010): Australian grand prix	
Frames number	244.408 frames
Duration	2h 42min 56s 8frames
Transitions number	54 transitions
Replays number	27 replays
Length logo transition	29 frames
FORMULA 1 (2010): Catalunya grand prix	
Frames number	225.000 frames
Duration	2h 30min
Transitions number	64 transitions
Replays number	32 replays
Length logo transition	29 frames
FORMULA 1 (2010): Monaco grand prix	
Frames number	289.564 frames
Duration	3h 13min 6s 4frames
Transitions number	38 transitions
Replays number	19 replays
Length logo transition	29 frames
 **Logo transition In**	 **Logo transition out**

Table A.9 Videos A.9.

(a) Australian Grand Prix,
(b) Catalunya Grand Prix,
(c) Monaco Grand Prix

APPENDIX B. Methodologies tried without successfully results

This appendix details some of the methodologies proposed, studied and developed that finally have not been implemented in the final algorithm because their results were not good enough.

[B.1]. Changing fixed thresholds for adaptable thresholds

To get better candidates in the candidate set, the first approximation was to increase the thresholds which limit the candidate selection.

The reason was that watching accurately the previous results, was possible to see correct candidates in each set, but the number of false positives was too high in comparison with the corrects, and the system fails when tried to determine the template computing the differences matrix. Therefore it was necessary to include more positive candidates to the set or to reduce the false positives.

After performing several experiments the conclusion was that it had been a wrong decision because instead of improving results, as high threshold selected, more candidates passed the filter, but both, correct and incorrect candidates, and the problem was still there. So, next approximation was to establish an adaptive threshold.

To devise an adaptive method, it was necessary to identify how a logo transition differs from the others transitions of the same video and moreover, look for correspondence among logo transitions of different cases.

So, graphically the accumulated frame differences in windows of 20 frames were represented for the test cases used until now. Over them were identified all logo transitions using a couple of vertical black lines for representing each one.

Analysing the graph of the shortest video test it can be seen that the four transitions have similar levels because, in fact, the contents is nearly the same. Moreover, the transitions vary from a starting level to a different ending level. So, to assure the goal and have the maximum number of frames corresponding to logo transition in the set, the minimum and maximum threshold should include all them.

After experimenting with these thresholds a new conclusion was that it was not necessary to include in the set all the frames of the logo transition allocated between the thresholds, because it also includes more false positives and the results were not as good as expected.

So, the final thresholds should be compressed between the old ones. The figure B.1 shows the mean luminance values of the four transitions which compose the shortest video A.7 and the thresholds selected for this case. The upper and lower thresholds belong to the first approximation and the others to the second. Both determine finally the area of interest as shows the figure B.2 just below.

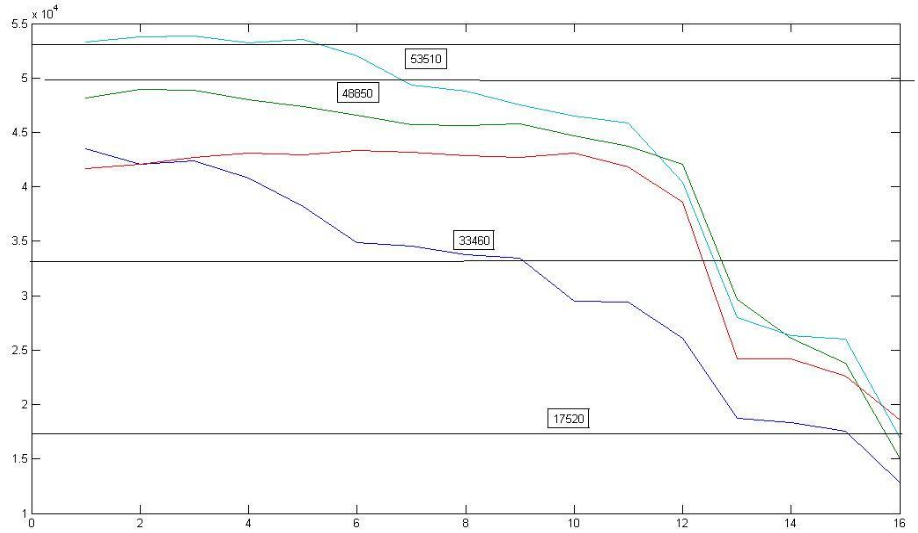


Fig. B.1 Thresholds computed for the video A.7

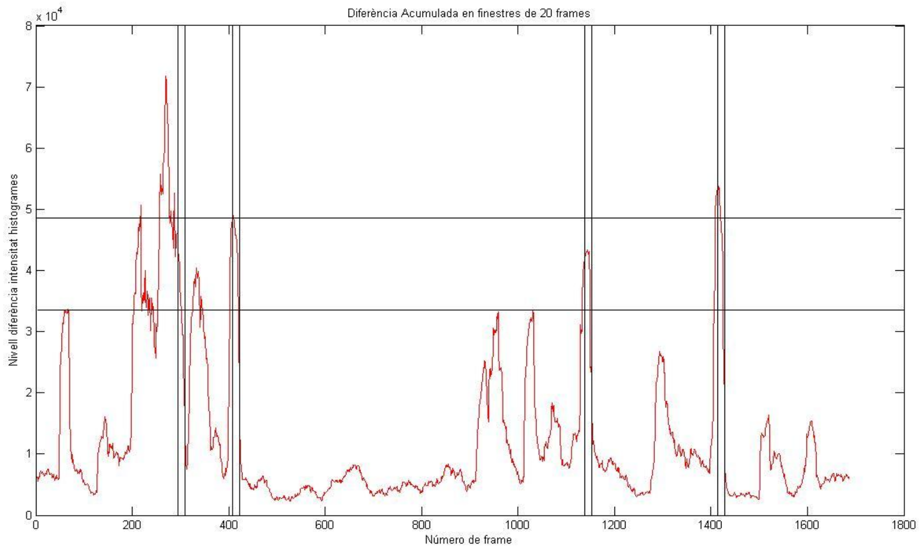


Fig. B.2 Resulting area of interest for the video A.7

The same process was performed with the complete videos A.1 and A.4 of the second test to confirm if the position of the thresholds selected in the shortest video A.7 and A.8 were more or less the same in the largest.

The figure B.3 shows the theoretical thresholds to select and the figure B.4 the resulting area of interest.

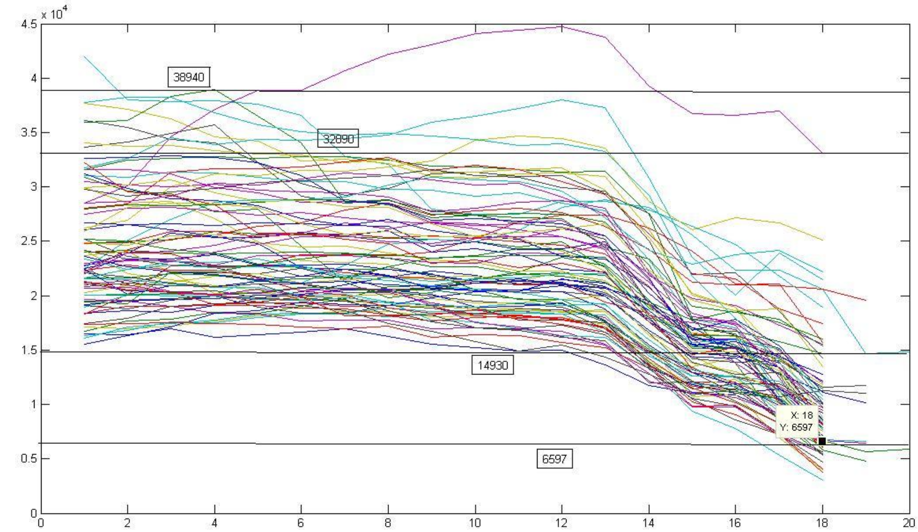


Fig. B.3 Threshold computed for the video A.1

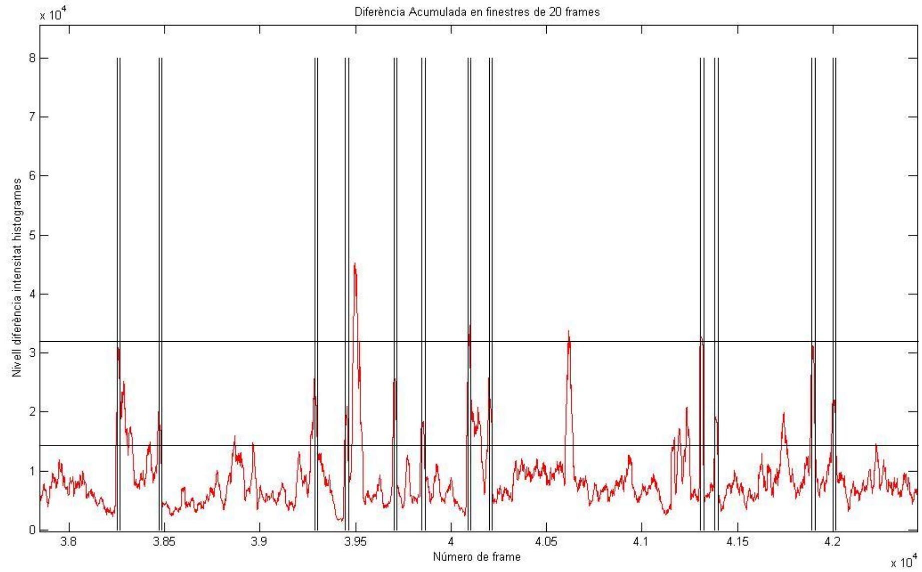


Fig. B.4 Resulting area of interest for a segment of the video A.1

Applying the same methodology to both cases the conclusion was that always the area of interest was in the upper part of the graphs although the range variation of the y-axes. So, the thresholds were computed automatically placing them in this region.

Testing again the shortest and the largest videos, the logo templates obtained were next respectively:

Fig. B.5 Frames selected as logo template for the videos A.7 and A.8

Fig. B.6 Frames selected as logo template for the videos A.1 and A.4

The final conclusion was that this methodology worked fine with short videos due to the poor variation in luminance of those frames which not correspond to the logo transitions. But in the full video, the number of shots increases and also number of frames with high differences in luminance, so the number of candidates included in the set was too much high and this produces wrong template detections in the next steps.

So, next step was focused again towards improving the selected images in the candidate set, especially for the largest videos.

[B.2]. Using RGB components instead of Gray

Until now, all graphs show the accumulated frames difference in windows of 20 frames. So, the theory says that all peaks correspond to the biggest changes on the video frames.

First change was to increase the windows up to 30 frames. The reason was that if logo transitions have durations approximately of 20 frames, the bigger changes should be at the beginning and the end of the transitions due the contrast with the normal frames of the video. So, accumulating these two big changes in only one position the value must be much greater than if not.

The second fine-tuning idea was focused on the first step of the process, to get the histogram and compute frame differences. Until now the differences were computed using initially the gray histogram of the images. This caused that some changes in colour could not be detected for having the same luminance level. So, the process changed and this time, instead of using the gray histogram, the histogram of each RGB component was computed. Therefore, the accumulated difference was the addition of the difference of all them. The figure B.7 shows these increments. The blue line shows the results got on the previous step (without using colour) and the red line the new ones.

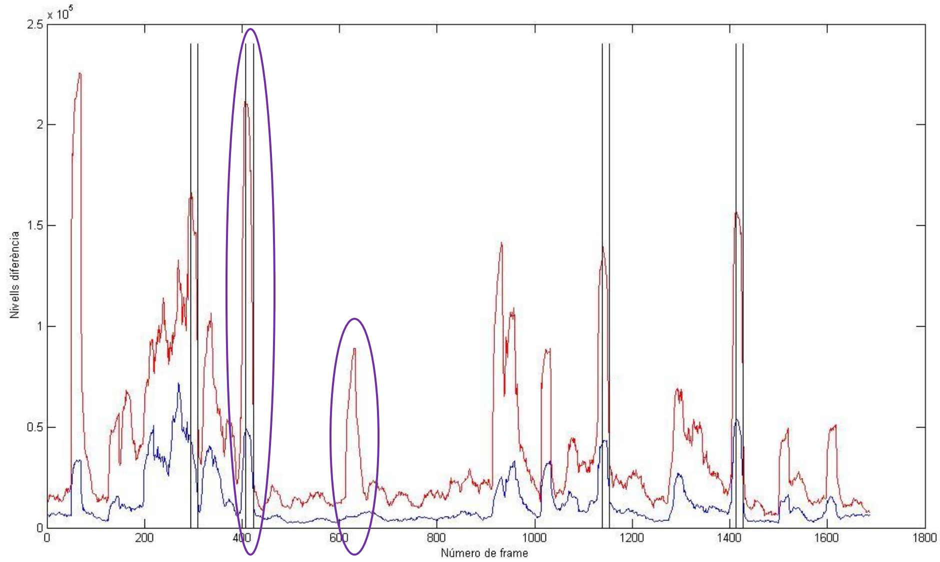


Fig. B.7 Gray and RGB accumulated difference of the video A.7

For instance, in the figure B.7 the region placed around the frame 600 was masked computing the gray histogram, and in fact, has a reasonable change using the colour histogram. In addition, other transitions like the placed around the frame 400 corresponding are emphasized a lot. The same happened with the video A.1 as shows the figure B.8.

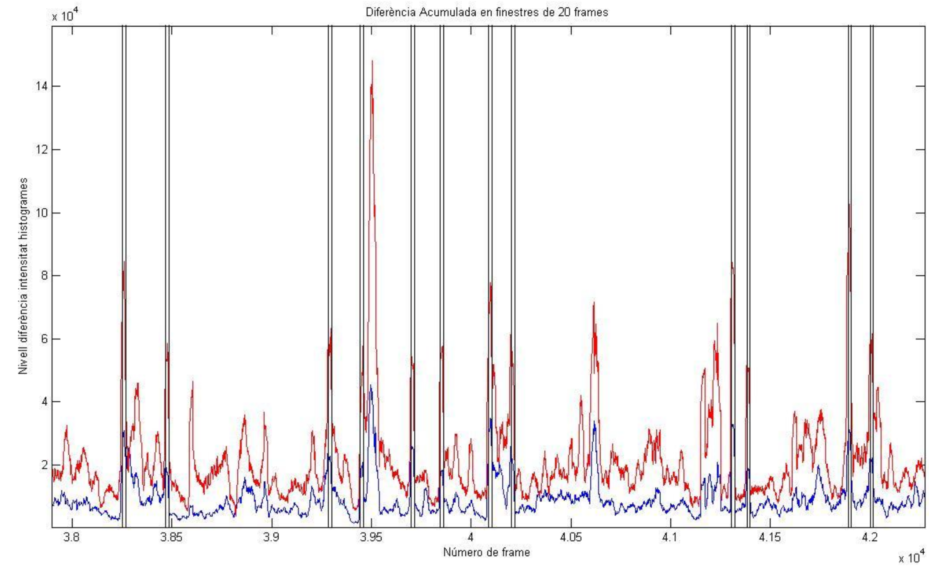


Fig. B.8 Gray and RGB accumulated difference of a segment of the video A.1

At the end, the conclusion was that computing the frame difference using the RGB components instead of the Gray, really all the significant changes were emphasized. This fact increases a lot the number of correct candidates included in the set, but it was not as good as expected, because this was not the only think that increase. The high number of changes emphasized produced that more images were between the adaptive thresholds shown, and the candidate set number increase too much. So, finally after applying the k-mean clustering process the system fails when trying to obtain the logo template among all the candidates. The next step was focus towards reducing the candidate set number while maintaining the number of correct candidates included.

[B.3]. Computing just accumulated frame differences

To reduce wrong cases in the candidate set was necessary to step back.
Although the accumulated frame differences in windows of 30 positions emphasize a lot the logo transition values, all frames around have values too much high. So, using just the RGB frame difference without any accumulating window, the resulting peaks were thinner (formed by less frames), and also highlighted than the rest. The figures B.9 and B.10 show clearly these differences for both tests. As could be seen, around logo transitions the frames highlighted were lower than before reducing a lot the candidate number.

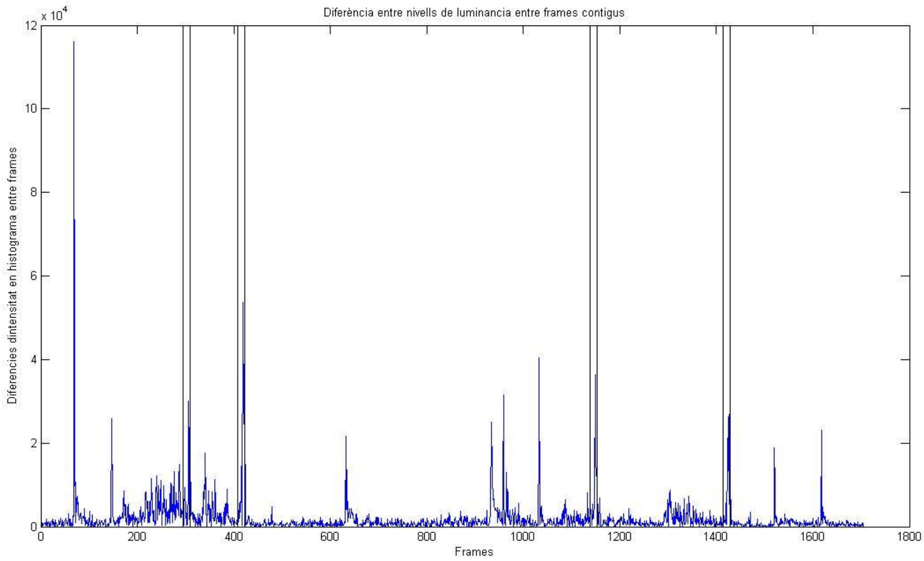


Fig. B.9 Frame difference of the video A.7

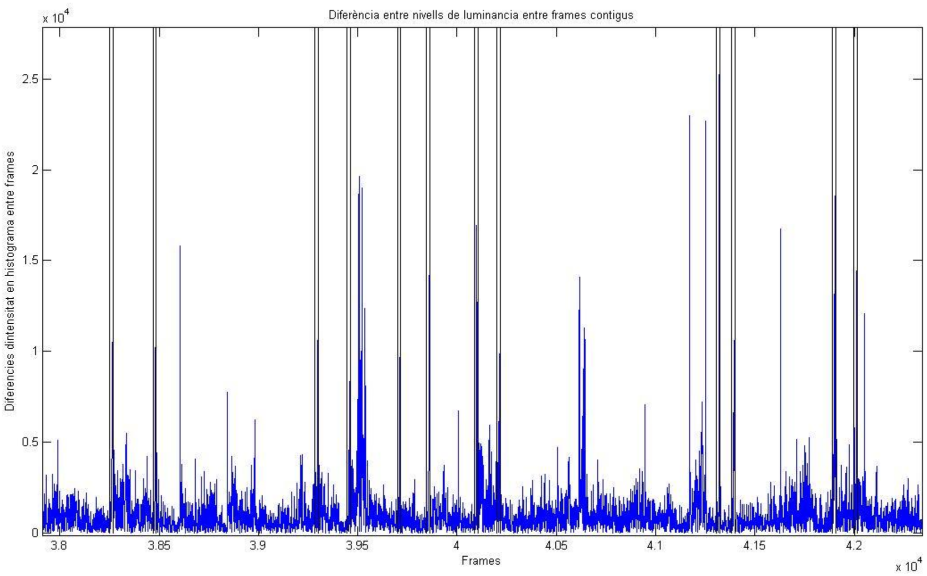


Fig. B.10 Frame difference of a segment of the video A.1

This time, the results were tested using the other TVC videos detailed on the APPENDIX A. After testing and obtaining the logo templates the conclusion was that this system works successfully for the test videos used until now, but for the other Videos from the TVC database with duration much longer around 2 and 3 hours the number of candidates in the set was huge. So, this fact made the process too slow when computing the matrix differences and majority achieving wrong results.

Therefore, the solution was divided in two branches:

- First, was to investigate if it was possible to implement a method much faster with at least the same positive results inside the candidate set. This led to raise another system to detect candidates: the Shot Boundary detection method explained on Chapter 3.
- The second was to improve next modules to distinguish between correct frames and false positives in the candidate set.

[B.4]. Successive K-mean using Luminance and Variance

The first approximation was to perform the k-mean process several times using the mean luminance values again and selecting each time the group which correspond to the higher center until reduce the number of candidates to less than 4500 candidates.

Then, a second k-mean process is performed several times again using the mean variance of the images, but selecting each time the group which correspond to the lower center until reduce the number of candidates to less than 500 images. So, the methodology just executed successively the k-mean algorithm reducing the candidate set. Next image shows it graphically.

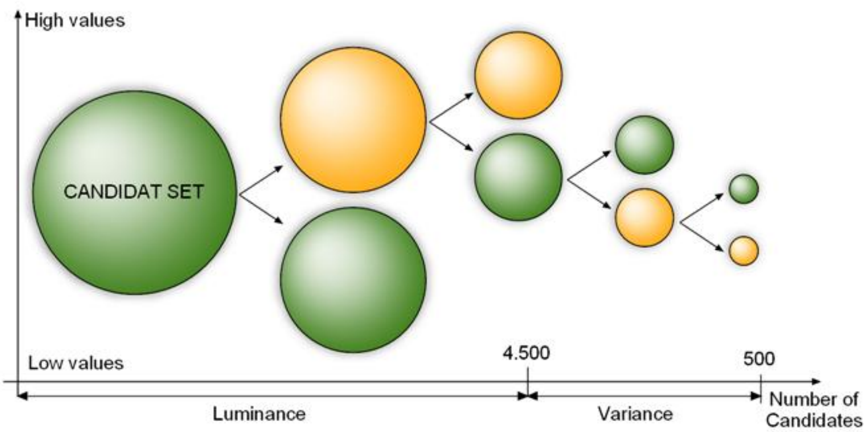


Fig. B.11 Group division

Analysing the final candidate set, the results were not good enough. Those logos with luminance and variance levels really high and low respectively were detected and included without problems. But those logos without these emphasized features were not included. So, the idea of reduce the candidate set was still present, but the problem must be restarted. The implemented alternative is explained on the Chapter 3.

APPENDIX C. Example of configuration file

This appendix shows the default configuration of the configuration file explained on Chapter 4.

In this case, if the first mode is selected at the output will be the resulting XML and the frames detected as logo transitions.

On the other hand, if the second or the third mode is selected, at the output will be in addition the image containing the 12 final candidates.

```
Path_dades             = D:\BUSCAMEDIA;
Candidats              = false;
Filtrat                = false;
Agrupacio              = false;
Candidats_finals       = false;
1Image_12Candidats     = true;
ImatgeSeleccionada     = true;
Transicions_i_Replays  = true;
```