# Segmentation and structuring of video documents for indexing applications

Ruxandra Georgina Tapu

## ▶ To cite this version:

Ruxandra Georgina Tapu. Segmentation and structuring of video documents for indexing applications. Economics and Finance. Institut National des Télécommunications, 2012. English. NNT : 2012TELE0050 . tel-00843596

**HAL Id: tel-00843596**

**https://theses.hal.science/tel-00843596**

Submitted on 11 Jul 2013

**THESE DE DOCTORAT CONJOINT TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE CURIE**

**Spécialité : Informatique et Télécommunications**

**Ecole doctorale :** Informatique, Télécommunications et Electronique de Paris

**Présentée par**

**Ruxandra Georgiana TAPU**

**Pour obtenir le grade de**
**DOCTEUR DE TELECOM SUDPARIS**

# SEGMENTATION ET STRUCTURATION DE DOCUMENTS VIDEO POUR L'INDEXATION

**Soutenue le : 07 Décembre 2012**

**devant le jury composé de :**

| | | |
|---|---|---|
| **Président de jury :** | **Madame le Maître de Conférences, HDR** | **Catherine ACHARD** |
| **Rapporteur :** | **Monsieur le Professeur** | **Azeddine BEGHDADI** |
| **Rapporteur :** | **Monsieur le Professeur** | **Philippe JOLY** |
| **Examinateur :** | **Monsieur le Professeur** | **Teodor PETRESCU** |
| **Directeur de thèse :** | **Monsieur le Professeur** | **Titus ZAHARIA** |

**Thèse n°:**2012TELE0050

# SEGMENTATION AND STRUCTURING OF VIDEO DOCUMENTS FOR INDEXING APPLICATIONS

# Acknowledgement

First and foremost, I am especially grateful to my thesis supervisor, Professor Titus Zaharia, for his guidance throughout my PhD thesis, and especially for giving me the freedom to pursue my own ideas in the area of computer vision (video indexing). Professor Titus Zaharia offered me invaluable help, starting from his academic advice and very helpful hints in the research proposed in this thesis.

Special thanks I would like to address to the members of my Ph.D. defense committee.

First, I would like to express all my gratitude to Professors Azeddine Beghdadi from University Paris 13 and Philippe Joly, from Institut de Recherche en Informatique de Toulouse for accepting the task of being reviewers. Their reviews, comments and fruitful suggestions helped me to improve the manuscript and to give its final shape.

To Professor Catherine Achard, from Université Pierre et Marie Curie – Paris VI, I would like to express all my thanks for her interest in my research work.

My warm thanks are also going to Professor Teodor Petrescu, from the University Politehnica of Bucharest, for his support and advices during all these years.

Being a part of the ARTEMIS Department within Institut Mines-Télécom/Télécom SudParis was for me a great experience. I have worked closely with many members of the department. In particular, I would like to thank here Raluca, Adriana, Afef and Andrei for providing me with such an enjoyable and stimulating working environment. Also, I would like to thank Marius Preda, Mihai Mitrea and Cătălin Fetiţa, Associate Professors within the ARTEMIS department, for their attentive listening and stimulating help.

To Madame Evelyne Tarroni I would like to express my gratitude for her help, patience and inexhaustible energy in solving all administrative problems. Also, I would like to thank Madame Marilyn Galopin from the Doctoral School EDITE de Paris, for her invaluable help in administrative issues.

I would finally like to thank my family and friends, both near and far, for providing me endless support, motivation, and inspiration. The unconditional love of my parents and brother was an unwavering source of strength throughout the writing of this thesis. Last, but not least, I would like to acknowledge the never-ending patience and support of my fiancé, Bogdan-Cosmin. Without his support, none of my achievements would have been possible.

# Table of contents

# List of Figures

# List of Tables

# 1. INTRODUCTION

Recent advances in telecommunications, collaborated with the development of image and video processing and acquisition devices has lead to a spectacular growth of the amount of the visual content data (still images, video streams, 2D graphical elements, 3D models...) stored, transmitted and exchanged over Internet. Within this context, elaborating efficient tools to access, browse and retrieve video content has become a crucial challenge.

Existing approaches, currently deployed in industrial applications are based mostly on textual indexation, which shows quickly its limitations, related to the intrinsic poly-semantic nature of the multimedia content and to the various linguistic difficulties that need to be overcome. The textual annotation has as main objective to associate a set of keywords to each individual item. In the case of huge repositories of audio-visual content, like those currently existing over Internet such a procedure requires a tremendous effort of human, manual annotation. In addition, the indexation process implies various annotators which may have different perceptions and sensibilities, which leads to subjective interpretations of the content. Finally, the multi-lingual aspects cannot be treated in a straightforward manner.

This shows the interest of developing content-based indexing and retrieval systems, able to interpret and describe the semantics of the multimedia content in an automatic manner. Still, to obtain viable and effective representations, a number of problems need to be overcome, including the identification of the pertinent items to be described, the selection of appropriate attributes and the resolution of the so-called semantic gap [Zhu09].

In particular, when considering the specific issue of video indexing, the description exploited by the actual commercial search engines is monolithic and global, treating each document as a whole. Such an approach does take into account neither the informational and semantic richness, specific to video documents, nor their intrinsic spatio-temporal structure. As a direct consequence, the resulting granularity level of the description is not sufficiently fine to allow a robust and precise access to user-specified elements of interest (*e.g.* objects, scenes, shots, events, chapters…). Within this framework, video temporal segmentation and structuring represents a key and mandatory stage that needs to be performed prior to any effective description/classification of video documents.

The process of automatic analysis of video content can be summarized into the following steps [Hanjalic02]:

- Video decomposition into various structural elements (object, keyframe, shot, scene...).
- Video indexing, *i.e.*, the process of automatically assigning content-based or high level descriptors to the detected video elements.
- Browsing and retrieval, *i.e.*, the process of presenting to the end user the search results, based on the considered descriptions.

This analysis shows that, when developing video indexing and retrieval applications, we first have to consider the issue of structuring the huge and rich amount of heterogeneous information related to video content. From the spatio-temporal structural point of view, a digital video can be decomposed into four different levels of detail (Figure 1.1), corresponding to scenes/chapters, shots, key-frames and objects:

- *Scene level* – corresponds to a group of video shots that are homogeneous with respect to a semantic criterion. A scene needs to respect three continuity rules in space, time and action.
- *Shot level* –defined as the sequence of successive frames from the moment a camera starts recording until it stops. The obtained video has the property of being visual continuous;
- *Keyframe level* – expressed as a set of representative images in each considered shot, able to summarize its visual content;
- *Object level* – corresponding to the spatial or spatio-temporal regions of arbitrary shapes and related to the salient objects of interest evolving in the sequence.

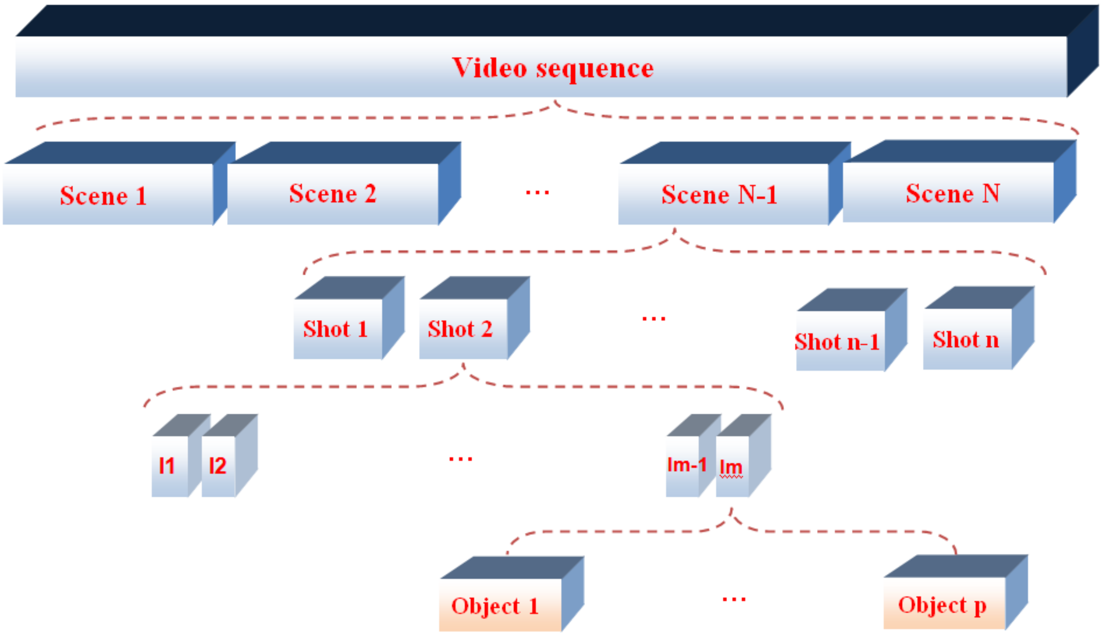


Figure 1.1. The structure of a video sequence.

The detection of the structural elements represents a key and mandatory stage that needs to be performed prior to any effective description/classification of video documents. In this thesis, we notably tackle this issue, and propose solutions for the detection of each of the above-mentioned structural elements involved.

The rest of the manuscript is organized as follows.

Chapter 2 tackles the issue of shot boundary detection. The shot segmentation algorithms have been intensively studied in the last two decades, as testifies the extremely rich literature dedicated to the subject, which is first presented and analyzed in this chapter. The high interest in the field can be explained by the fact the shot is generally considered as the “atom” which provides the basis for the majority of video abstraction and high level semantic description applications. The analysis of the literature shows that the challenge is to elaborate robust shot detection methods for both abrupt and gradual transitions, which can achieve high precision and recall rates whatever the movie quality and genre, the creation date and the techniques involved in the production process, while minimizing the amount of human intervention. In the second part of this chapter, we introduce and validate a novel shot boundary detection algorithm able to identify both abrupt (*i.e.*, cuts) and gradual transitions (*i.e.*, fades, wipes…). The technique is based on an enhanced graph partition model, combined with a multi-resolution analysis and a non-linear filtering operation. In order to reduce the computational complexity, an initial segmentation step of the input stream is performed with the help of a sliding window that selects a constant number of $N$ frames from the original video signal centered on the current frame. We focused next in reducing the global computational complexity by implementing a two-pass approach. In a first step, the algorithm detects time intervals which can be reliably considered as belonging to the same shots. Abrupt transitions considered as certain are also detected in this stage. In a second step, the multi-resolution graph partition algorithm is further performed only for the remaining uncertain time intervals.

In Chapter 3 the video abstraction problem is considered. In our case, we have developed a keyframe representation system that extracts a variable number of images from each detected shot, depending on the visual content variation. The proposed technique is based on a leap-extraction algorithm that consider for analysis only the images located at integer multipliers of window size (used of shot detection) and not the entire video flow. By computing the graph partition within a sliding window, our method ensures that all the relevant information is taken into account. Let us also note that the number of detected keyframes set per shot is not fixed *a priori*, but automatically adapted to the content of each shot.

Chapter 4 deals with the issue of high level semantic segmentation into scenes. First, the most relevant methods introduced in the last years are presented and discussed. Then, a novel scene/DVD chapter detection method is introduced and validated. Here, spatio-temporal coherent shots are clustered into the same scene based on a set of temporal constraints, adaptive thresholds and with the help of a new concept - neutralized shots. Concerning the keyframe visual similarity involved in the above-described process, we consider two different approaches, based on the chi-square distance between HSV color histograms and the number of matched interest points extracted described with SIFT descriptors.

Chapter 5 considers the issue of object detection and segmentation. The main concept developed here is the one of visual saliency. After reviewing and analyzing the state of the art techniques, we introduce a novel spatio-temporal attention model. The spatial saliency map it is based on an enhanced stationary technique called region-based contrast (RC) that separates large scale objects from their surroundings. The temporal model relies on interest points correspondence, geometric transforms (*i.e.*, homographic motion model), motion classes estimation (using agglomerative clustering) and temporal consistency. The proposed technique is extended on 3D videos by representing the stereoscopic perception as a 2D video and its associated depth. The approach is robust to complex background distracting motions and does not require any initial knowledge about the object size or shape. The various experimental results and comparisons with existent methods demonstrate the effectiveness of the proposed technique.

Finally, Chapter 6 concludes the manuscript and highlights the main contributions proposed in this work. Some perspectives of future research are also highlighted, in terms of both methodology and application for novel multimedia services.

# 2. SHOT BOUNDARY DETECTION

**Summary:** In this chapter, we consider the issue of shot boundary detection. We first review the state of the art, and highlight, for each method, principles, advantages and limitations. The analysis of the literature shows that the challenge is to elaborate robust shot detection methods, which can achieve high precision and recall rates whatever the movie quality and genre, the creation date and the techniques involved in the development process, while minimizing the amount of human intervention.

In the second part of this chapter, we introduce and validate a novel shot boundary detection algorithm, based on the graph partition (GP) model and scale-space filtering. Within this framework, the detection is performed on the weighted minimum vector of the local derivatives. We focus next in reducing the global computational complexity by developing a two-pass approach. The experimental results demonstrate the superiority of the proposed approach with respect to other state of the art algorithms, with average gains in precision and recall rates of 8%, for 25% savings in computational time. The method achieves high precision (superior to 90%)) and recall (superior to 95%) rates whatever the movie quality and genre and for both abrupt and gradual transitions.

**Keywords:** shot boundary detection, abrupt and gradual transitions, graph-partition model, scale-space filtering.

The shot, defined as a video interval corresponding to a continuous camera capture, constitutes a fundamental item in the film production process. Thus, movies are created by capturing a set of shots, which are then assembled by juxtaposition during the editing stage, in order to create the full video document. Various transitions between shots can be considered, in order to reinforce motion, ideas and movement within the artistic production process. The various types of transitions that are usually considered are briefly recalled in the following section.

## 2.1. TYPES OF TRANSITIONS

There are two basic types of shot transitions: abrupt and gradual. Abrupt transitions, so-called *cuts,* correspond to the direct juxtaposition of two different shots taken with different cameras. Some examples of cuts are illustrated in Figure 2.1.a.

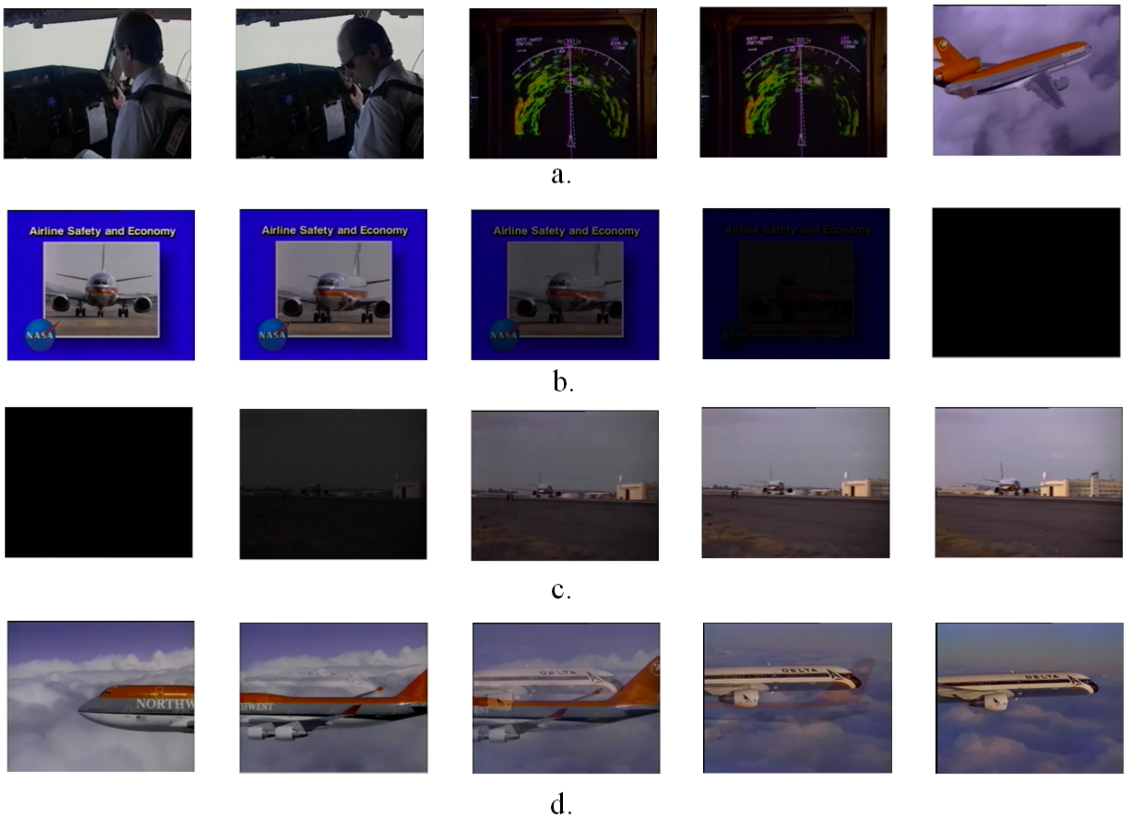


Figure 2.1. Example of the most encountered transitions in a video stream:
(a) Cut; (b) Fade-out; (c) Fade-in; (d) Dissolve.

In this case, the individual frames of the two different shots are simply concatenated, without considering any specific video processing techniques. *Cuts* represent the majority of transitions encountered in video sequences (85%).

Let us denote by $t_{cut}$ the instant where the cut between two shots $S_1(x, y, t)$ and $S_2(x, y, t)$ occurs. Here, ($x$, $y$) denote the spatial coordinates, while $t$ stands for the temporal one. After the concatenation of the two shots, the resulting video sequence, denoted by $S(x, y, t)$, can be described by the following equation [Lienhart01]:

$$S(x,y,t) = (1-u(t-t_{cut}))\cdot S_1(x,y,t) + u(t-t_{cut})\cdot S_2(x,y,t) \tag{2.1}$$

where $u(t)$ is the unit step function:

$$u(t) = \begin{cases} 1, & for\ t \geq 0 \\ 0, & else \end{cases} \tag{2.2}$$

By definition, a *cut* transition has the property of introducing a visual discontinuity in the video sequence, indicating a change in space and time.

A second family of shot transitions, so-called *gradual transitions*, may also be considered when editing a given video document, in order to achieve various artistic effects, corresponding to smoother visual transitions. Such effects are used to enhance the impact of the modality (*i.e.* refers to how viewer performance depends on the presentation mode of studied items), to add meaning or to accentuate emotions.

A gradual transition combines two shots by using chromatic, spatial or spatio-chromatic effects. Various classes of gradual transitions can be considered, including *fade-in, fade-out, dissolve, wipe, morphing...* The most commonly used (90%) [Petersohn08] ones are the fade-in/out and the dissolves.

A *fade-out* is a gradual decrease in brightness of a considered frame, resulting in a black frame (Figure 2.1.b). On the contrary, a *fade-in* is a gradual increase in intensity starting from a black image and up to a given frame (Figure 2.1.c). The *fades* are used in order to mark a distinct break in the movie, usually indicating a change in time, location, or subject matter. Most films present various *fade-in/fade-out* effects used together, one after the other, or begin with a *fade-in* and end with a *fade-out*. A special effect can also be created by combining a *fade-in/out* structure. In this case, the transition is called a *fade group*.

According to [Lienhart01], a fade sequence $S(x,y,t)$ of duration $T$ can be created by gradually modifying the pixels color/intensities as described in the following equation:

$$S(x,y,t) = f(t)\cdot S_1(x,y,t)\ ,\ t \in [0,T] \tag{2.3}$$

where $f(t)$ is a temporally monotone transfer function which satisfies the following conditions:

- for a *fade-in,* $f(0)=0$ and $f(T)=1$,
- for a *fade-out,* $f(T)=0$ and $f(0)=1$

In most of the cases, the transfer function $f(t)$ is linear, defined as:

$$f_{fade-in}(t) = \frac{t}{T} \tag{2.4}$$

$$f_{fade-out} = 1-\frac{t}{T} \tag{2.5}$$

Another family of popular shot transitions concerns the so-called *dissolves*. *Dissolve* transitions consists of blending the final images of a first shot with the first frames of the successive, second shot. Figure 2.1.d illustrates an example.

The dissolve transformation is performed in the space of pixels intensities and does not use any geometric transform. *Dissolve* provides a smooth transition, its speed affecting the overall mood and flow of the video sequences. *Dissolves* are often encountered in some specific video genres such as dance and music pieces or drama. They are also used in live sports to separate slow motion replays from the live action.

A *dissolve* sequence can be mathematically described [Ngo03] as a sequence $S(x,y,t)$ of duration *T*, created by combining two distinct video shots $S_1(x,y,t)$ and $S_2(x,y,t)$. The blending process is defined as described by the following equation:

$$S(x,y,t) = f_1(x,y,t)\cdot S_1(x,y,t) + f_2(x,y,t)\cdot S_2(x,y,t)\,,\ t \in [0,T] \qquad (2.6)$$

Here, the transition functions (also called *intensity scaling function*) $f_1(x,y,t)$ and $f_2(x,y,t)$ characterize the blending process and can be either linear or non-linear.

In the general case, the main constraints to be respected are:

1. , $0 < f_2(x,y,t) < 1$
2. $f_1(x,y,t+1) \le f_1(x,y,t)$, $f_2(x,y,t) \le f_2(x,y,t+1)$.

However, the following simplifying assumptions are frequently considered [Ngo03]:

$$f_1(x,y,t) = f_1(t)\,,\ f_2(x,y,t) = f_2(t)\,,\ S_1(x,y,t) = g(x,y) \text{ and } S_2(x,y,t) = h(x,y)\,.$$

This case corresponds to a dissolve between two static frames which may represent the last and first frames of the considered two adjacent shots in the [0, T] time interval. Equation (2.7) can be in this case re-written as

$$f(x,y,t) = f_1(t)\cdot g(x,y) + f_2(t)\cdot h(x,y) \qquad (2.7)$$

The most common dissolve types correspond to cross-dissolves. In this case, the transition function is defined as:

$$f_1(t) = 1 - f_2(t), \quad f_2(t) = \frac{t}{T}, t \in [0,T], \qquad (2.8)$$

Let us also mention a second type of dissolve transitions, so-called additive (Figure 2.2.b) can be considered. An additive dissolve is defined with the help of the following function:

$$f_1(t) = \begin{cases} 1, t \in \left[0,\frac{T}{2}\right] \\ 2\left(1-\frac{t}{T}\right), t \in \left[\frac{T}{2},T\right] \end{cases}, and\ f_2(t) = \begin{cases} \frac{2t}{T}, t \in \left[0,\frac{T}{2}\right] \\ 1, t \in \left[\frac{T}{2},T\right] \end{cases} \qquad (2.9)$$

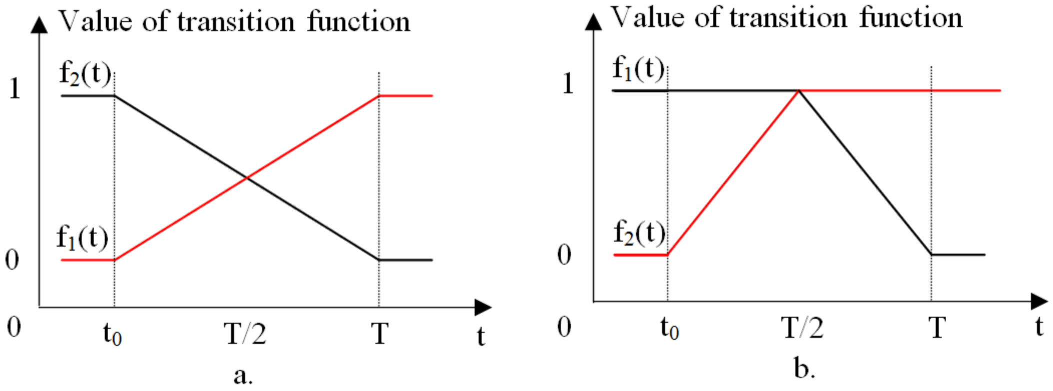


Figure 2.2. Intensity scaling function for two dissolve type: (a) cross-dissolve; (b) additive-dissolve.

Wipes distinguish from other types of gradual transitions by affecting the spatial distribution of exiting/entering edge pixels. In this case, each frame will have a portion of the old image and a region of the new, entering shot (Figure 2.3). Between adjacent frames, a single strip of the image will change, which determines a major variation of pixels inside the strip and lower variation of edge pixels in other areas of the image [Zabih95].

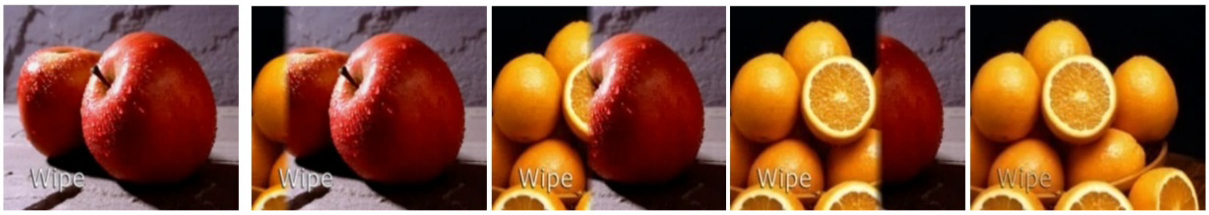


Figure 2.3. Wipe transition.

A morphing transition can be defined as the construction of sequence corresponding to a gradual transition between a source image and a target one. Generally, the morphing effects are obtained with the help of the cross-dissolve or fading techniques which permit to achieve a smooth change of image content (*i.e.* texture and/or color) from source to target frames. The color of each image pixel is interpolated over time from the first image value to the corresponding second image value. Most often, linear interpolation is utilized. This process is called cross-dissolve interpolation [Gomes99] and is illustrated in Figure 2.4.

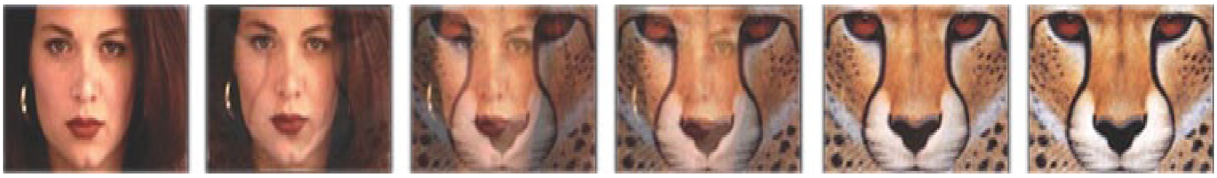


Figure 2.4. Morphing transition

A lot of research effort has been dedicated to the automatic detection of shot breaks, during the last two decades. Let us analyze how the state of the art methods consider this issue, for both abrupt and gradual transitions.

## 2.2. RELATED WORK

Historically, the first shot detection methods have considered the issue of identifying *cut* transitions. In this case, the potential strong variation of the visual content associated to two consecutive shots can significantly simplify the problem. More recent techniques are focused on detecting gradual transitions with special effects.

In all cases, a set of audio-visual features need to be used in order to detect transitions. In our work, we have considered uniquely methods based on visual features. The various types of visual features exploited are described in the next section.

### 2.2.1. Visual features

The objective is to describe a given frame with a compact, but discriminant set of visual attributes. Thus, a set of salient characteristics are extracted either from the entire frame or only from a region of interest supposed to be detected.

Such features include:

- Pixel intensities,
- Color histograms and moments,
- Contour/edge descriptions,
- Compressed-domains characteristics,
- Motion features

#### *2.2.1.1. Methods based on pixel intensity analysis*

One of the very first shot detection approaches relies on the differences in intensity values of corresponding pixels between two successive frames [Otsuji91], [Choubey97]. No color/motion/shape information is considered in this case, but solely the luminance channel of the considered videos.

If the number of pixels $N_{pixels}$ which change from one image to another exceeds a certain threshold ( $T$ ) a shot transition is declared. The number of changing pixels can be mathematically expressed as:

$$N_{pixels} = \frac{\sum_{x,y=1}^{X,Y} D(x,y,t,t+1)}{X \cdot Y} \cdot \quad , \tag{2.10}$$

where

$$D(x,y,t,t+1) = \begin{cases} 1, & if \ \left| S(x,y,t) - S(x,y,t+1) \right| > V \\ 0, & otherwise \end{cases} \tag{2.11}$$

Here, $X$, $Y$ denote the horizontal and vertical image sizes, and $V$ is a threshold specifying the minimum absolute difference value starting from which a pixel is considered as changed.

Such a direct approach implicitly considers relatively static, with no or relatively poor camera/object motion. However, in the presence of large objects with high amplitude motion or in the case of camera motion, such an approach will obviously lead to false detections. Moreover, in the case of gradual transition, the underlying pixel intensity constancy assumption does not hold. Such problems are illustrated in Figures 2.5 and 2.6.

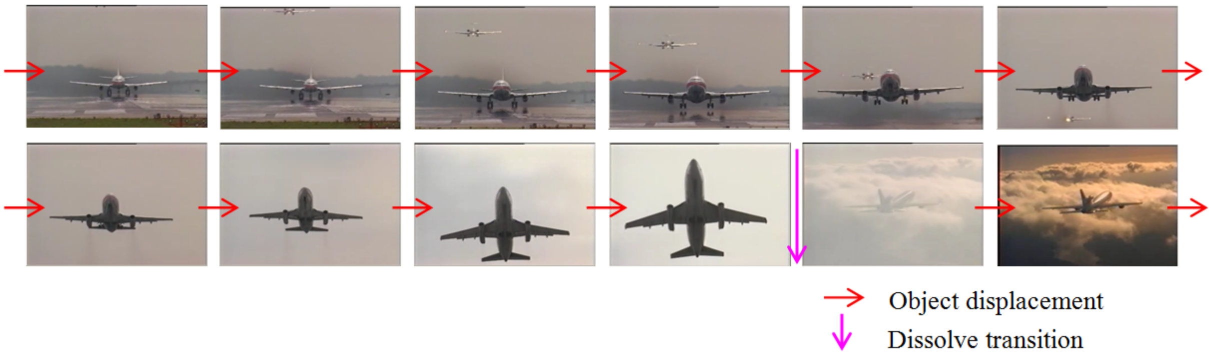


Figure 2.5. Large object motion followed by a dissolve transition.

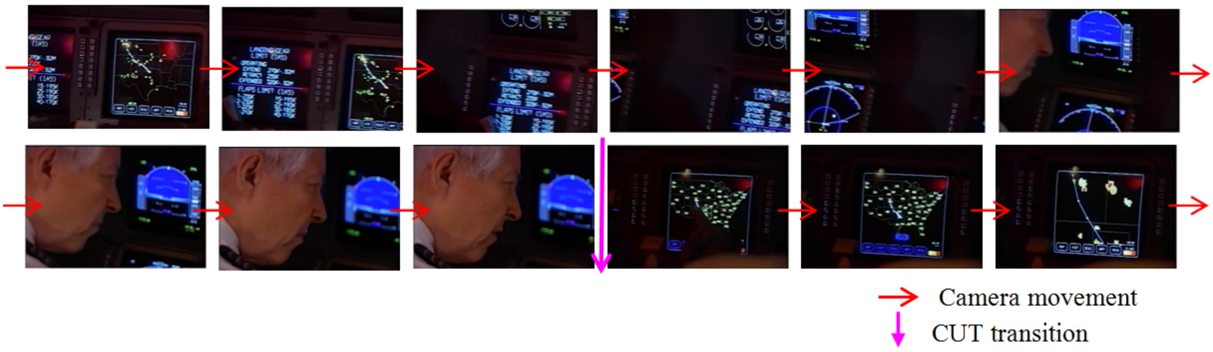


Figure 2.6. Camera motion followed by a CUT transition.

In addition, such methods are sensitive to several factors, including noise introduced by the acquisition process, often present in low resolution videos.

In order to increase the robustness to camera and object motion an enhanced method was proposed in [Zhang93], which introduces an additional, pre-processing step in the pixel intensity analysis. They use of an $3\times3$ averaging spatial filter before the comparison. The algorithm shows superior performances with respect to the baseline method in [Nagasaka91].

However, detecting shot changes by exploiting an analysis at the pixel level is obviously not a reliable approach.

For this reason, the method presented in [Shahraray95] considers the analysis at the level of image blocks. Each frame is partitioned into 12 rectangular blocks. The algorithm aims to determine the best match for each block of a considered frame, in the respective neighborhood of the previous image based on the pixels intensity values. Next, a non-linear statistics is used to combine the matched values, *i.e.* the weight of the match in Equation (2.12) will depend on its order in a predefined match value list:

$$N_{pixels} = \sum_{k=1}^{B} c_k \cdot D(k, t, t+1) \tag{2.12}$$

where $c_k$ is a predefined coefficient for the $k$ block, $B$ is the total number of blocks in the frame and $D(k, t, t+1)$ is a partial match value between $k^{\text{th}}$ blocks between successive frames.

The decision is made based on the assumption that, if the information in the current image can correctly predict the following image, both images belong to the same shot.

A different technique introduced and validated during the TRECVID 2004 campaign is presented in [Jaffre04]. First, low resolution versions of the original frames are created by sub-sampling the frames with a factor 1:8. Next, the resulted images are converted from RGB to HSV color space and only the V component is kept for further analysis. With every new frame, the absolute difference between pixels intensity is computed. Then, the method starts counting the number of pixels which are different of more than 128 grey levels. If the resulting value is superior to the average values, obtained for each frame since the last detected shot, a cut transition is identified. Regarding the gradual transitions the method can detect only dissolves and fades. In this case, authors propose counting only pixels with a constant sign of the intensity variation over the duration of a transition. If this number is above a threshold the beginning of a gradual transition is declared. The effect is considered ended when this number becomes lower then a second threshold. Finally a validation step is introduced. For the first and last frame of a candidate transition the absolute difference is computed and a new binarized image is created. In order to reduce noise a morphological dilatation with a 7x7 mask is applied. A candidate frame transition is validated as a transition if the number of pixels that have changed after the dilatation is above a threshold.

In [Lienhart97], authors propose an algorithm that first locates all monochrome frames in the video as potential start/end points of fade transitions. The monochrome frames are identified as images in the video sequence where the standard deviation of pixels intensities is close to zero, with respect to a given threshold. A fade transition $E(x, y, t)$, of length $d^i$ starting from a shot $S_{i-1}$ is modeled as described by the following equation:

$$E(x, y, t) = S_{i-1}(x, y, t) \cdot \left(1 - \frac{t}{d^i}\right), t \in \left[0, d^i\right] \quad . \tag{2.13}$$

The standard deviation of pixel intensities is then defined as:

$$\begin{aligned}\sigma\big(E(x, y, t)\big) &= \sqrt{\mu(X^2) - \mu^2(X)} = \\ &= \left(1 - \frac{t}{d^i}\right) \cdot \sqrt{\mu\left(S_{i-1}^2(x, y, t)\right) - \mu^2\big(S_{i-1}(x, y, t)\big)}\,,\end{aligned} \tag{2.14}$$

where $\mu$ denotes the statistical expectation operator, applied to the pixel intensities of the considered frame.

Starting from the detected monochrome images, a search in both temporal directions is performed in order to check for a linear increase in the standard deviation of pixels intensity. The method starts by computing the line of regression over the existent monochrome frames. With any new detected monochrome frame a novel regression line is computed only if the correlation does not decrease with more than 3% or if the line slope is not halved. In order to identify a transition, two measures are evaluated: the absolute value of the correlation, between the standard deviation of two successive monochrome frames, and the slope of the computed fitting line. If both values are superior to user-specified thresholds, and if the duration of the candidate fade transition lies within the considered typical transition ranges (*i.e.*, 4 to 100 frames), a fade transition is identified.

Another fade detection algorithm is introduced in [Alattar97]. For an ergodic video sequence $f$ the method starts by identifying all the negatives spikes of the second order difference derivative of the image intensity variance function $\sigma_f^2$. Fade-in and fades-out are modeled as described in the following equations:

$$d_{2(fade-in)}(n) = \begin{cases} \dfrac{2\sigma_f^2}{M^2}, & 1 < n < M \\ \dfrac{\sigma_f^2}{M^2} \cdot (2 - M), & n = M \\ 0, & M < n < N \end{cases} \tag{2.15}$$

$$d_{2(fade-out)}(n) = \begin{cases} 0, & 1 < n < M \\ \dfrac{\sigma_f^2}{(N-M)^2} \cdot (1 - 2(N-M)^2), & n = M \\ \dfrac{\sigma_f^2}{(N-M)^2}, & M < n < N \end{cases} \tag{2.16}$$

where $d_2$ represent the second order derivatives of the $\sigma_f^2$ variance function and $[M, N]$ is the temporal interval of the considered fade effect. The variance function $\sigma_f^2$ is defined as:

$$\sigma_f^2 = \frac{1}{X \cdot Y} \sum_{i=1}^{X} \sum_{j=1}^{Y} \left(a_{i,j} - m_f\right)^2 \tag{2.17}$$

where X, Y denote the horizontal and vertical image sizes, $a_{i,j}$ is the pixels intensity value and $m_f$ is the mean value.

The method is based on the assumption that during a fade transition the second order difference function remains constant. Moreover, at the beginning/end of a fade-in/-out transition the function presents a large variation (spike) due to scene changes. From Equation (2.15) and (2.16) it results that a fade-in interval ends, at location $n = M$, with a negative spike of magnitude $\frac{\sigma_f}{M^2} \cdot (2 - M)$. Also, in the case of a fade-out, the transition starts with a negative spike of magnitude $\frac{\sigma_f^2}{(N-M)^2} \cdot (1 - 2(N - M)^2)$, at location $n = M$.

In the neighborhood of a negative spike the algorithm determines the existence of a fade transition as well as its size.

A fade region is detected if $\Delta I(n)$ returns near a negative spike a positive constant value:

$$\Delta I(n) = \begin{cases} \left|\frac{m_f - C}{M}\right|, 0 < n < M \\ 0, M \leq n \leq N \end{cases}, \tag{2.18}$$

where *C* is the begin/end color intensity of the fade transition. For this purpose, the right mean ($m_r$) and variance ($\sigma_r$) of $\Delta I(n)$ within an adjustable window on the right of the negative spike are evaluated. Similarly, authors determine the left mean ($m_l$) and variance ($\sigma_l$) on the left side of the negative spike. The window size is adaptively selected such that the algorithm can include for analysis all frames with a positive absolute change in luminosity greater than a predefined threshold. If $m_r > m_l$ , a fade-out is detected. Otherwise if $m_r < m_l$ a fade-in is identified. If both values are equal the side with the smaller variance gives the transition type.

An improvement of the above-presented algorithm is presented in [Truong00]. The method starts by identifying the monochrome frames of a video sequence. They differentiate between quick fades that last only a few frames (3 to 5), and slow fade effects with a duration exceeding 100 frames. In the second step, they compute the second order derivative of the luminance variance (Equation (2.15) and (2.16)) in order to identify a negative spike situated close to a fade-in/out transition.

However, let us note that large object displacements as well as any kind of camera motion can produce negative spikes in the derivative signal similar to those given by a fade-in/out shot transition.

In [Truong00,] authors propose to further extend the method in order to detect dissolve transitions. The approach is based on the assumption that a dissolve can be described as a combination of a fade- out and fade-in techniques, superimposed on the same film strip. As in the case of fade transitions, for “ideal” dissolves, the luminance mean curve changes linearly, while the luminance variance has a parabolic shape. So, if they consider the second order derivative of the luminance variance curve, two large negative spikes should appear at the start/end of a dissolve. However, in practice, the two negative spikes at the beginning and end of a dissolve are often poorly pronounced due to noise and motion in video.

A twin comparison model has been introduced in [Zhang93]. The technique requires two thresholds in order to detect any type of transition: a higher one, $T_h$used to identify hard cuts, and a lower one $T_l$ dedicated to gradual transitions. They begin by detecting high discontinuities values, corresponding to hard cuts, based on $T_h$, and then the threshold $T_l$ is applied to the rest of the discontinuities values. If a discontinuity is higher then $T_l$ then the start of a gradual transition is considered.

An enhanced version of the above-mentioned technique is proposed [Zheng05]. In this case, the classical twin comparison method is employed for detecting short transitions (under 5 frames) and a novel approach, that adaptively determines the lower threshold is introduced for identifying the other types of transitions. The lower threshold is fixed according to the amount of motion present in the video, estimated with the help of the motion vectors, associated to each macro-block, included in the MPEG compressed domain.

For the $I$ type frames the macro-blocks are intra-frame encoded and thus do not contain motion vectors. In this case the motion feature is computed via interpolation from the forward and backward frames of type $B$ or $P$. In addition, for the frame of type $B$ or $P$, some of their macro-blocks are not encoded through motion compensation. Only the macro-blocks with motion vectors are used to estimate the global amount of motion, which is characterized by the mean of the absolute motion vectors in horizontal and vertical directions:

$$MV_h = \frac{1}{N}\sum_{i=1}^{N}\left|mv_h^i\right| \ , MV_v = \frac{1}{N}\sum_{i=1}^{N}\left|mv_v^i\right| \qquad (2.19)$$

where $N$ is the total number of macro-blocks.

In this case the lower threshold ($T_l$) is determined using a linear equation:

$$T_l = \alpha + \beta \cdot (MV_h + MV_v) \qquad , \qquad (2.20)$$

where $\alpha$ and $\beta$ are fixed coefficients determined heuristically.

In [Volkmer04], authors propose a gradual transition detection method, based on a sliding window centered on the current frame. A distance between the current frame and all the other frames in the sliding window is then computed. The average values of the whole set of distances provides an inter-frame difference measure, which is finally exploited for shot detection purposes (based on a thresholding process). The distance between frames is computed based on the Manhattan (city-block) measure between pixels intensities values, defined as:

$$L_1\left(f_i, f_j\right) = \sum_{k=1}^{X}\sum_{l=1}^{Y}\left|f_i(k,l,t) - f_j(k,l,t)\right|, \qquad (2.21)$$

where $f_i$ and $f_j$ defines two successive video frames.

A particular strength of the approach is its capability of accurately detecting the start and end of gradual transitions.

One of the first algorithms proposed to detect gradual transitions (fade/dissolve) based on pixel intensity variation is described in [Hampapur94]. The model exploits the chromatic scaling model, described in Equation (2.22).

$$E(x, y, t) = S(x, y, t) \cdot \left(1 - \frac{t}{l_0}\right) \tag{2.22}$$

where $S(x, y, t)$ denotes the image sequence, $l_0$ its temporal length and $E(x, y, t)$ represents a frame in the editing interval (*i.e.*, the set of frames generated during a transition between two shots).

In this case, the fade-in/out operations are represented as some combination of chromatic scaling operation. The fade-in ($E_{fi}(x, y, t)$) and fade-out ($E_{fo}(x, y, t)$) are determined as:

$$E_{fo}(x, y, t) = S_1(x, y)\left(\frac{l_1 - t}{l_1}\right) \; and \; E_{fi}(x, y, t) = \; S_2(x, y)\left(\frac{t}{l_2}\right), \tag{2.23}$$

where $l_1$and $l_2$ are the fade-out/in rates in terms of the number of shots.

The chromatic scaling models [Hampapur94] are used to classify dissolve effects in the video production process. In this context, a dissolve transition is modeled as a linear combination of two shots, as described in the following equation:

$$E_d(x, y, t) = S_1(x, y)\left(\frac{l_1 - t}{l_1}\right)_{(t_1, t_1 + l_1)} + S_2(x, y)\left(\frac{t}{l_2}\right)_{(t_2, t_2 + l_2)} \tag{2.24}$$

where $l_1$and $l_2$ are the dissolve lengths in both shots, $t_1(t_2)$ is the starting time of shot $S_1$ (respectively, $S_2$).

During dissolves and fades, the chromatic image is assumed to have a reasonably constant value. Unfortunately, this technique is very sensitive to camera and object motion. In practice, the assumption that no motion can be encountered during a dissolve transition is not satisfied. This causes high rates of missed detections and false positives.

In a general manner, methods based on pixel intensity variations are highly sensitive to noise and motion present in the video sequence.

In order to overcome such limitations, some more global representations are proposed, which include, in addition to the luminance value, colorimetric information. Such approaches are described in the following section.

#### *2.2.1.2. Methods based on color histograms*

The color histogram methods are based on the assumption that between two consecutive frames of a given video shot, the global color content presents relatively low variations. Such a color content can be effectively described with the help of color histograms.

In a basic form, the underlying principle can be stated as follows:

1. Describe each video frame with the help of a color histogram,
2. Evaluate the variation between each two consecutive frames as a distance/similarity measure between associated color histograms,

3. Identify shot transitions whenever the measure evaluated in step 2 is superior to a given threshold.

Color histograms offer the advantage of a compact representation, in terms of bandwidth/storage requirements. In addition, efficient similarity measures can be associated with. Finally, a color histogram presents a relative robustness to slight variation of the image content.

For all these reasons, color histograms are good candidates for representing the content of video frames within the context of shot detection applications. However, appropriate color spaces, color quantization schemes, histogram dimensions, similarity measures and inter-frame comparison strategies have to be considered. Let us analyze how such aspects are taken into account by the methods proposed in the literature.

In [Lienhart99], a color histogram is computed in the discretized RGB color space. A uniform quantization, performed marginally on each color component with a number of $B=7$ bits (*i.e.*, $2^B$-1 = 128 quantization levels on each color component) is retained. For each frame *i*, a similarity measure defined as the $L_1$ distance between the histograms associated to the current frame *i* and the previous frames (*i*-1) is computed. This leads to a 1D color histogram difference signal, denoted by $CHD_i$.

In order to identify shot transitions, the peaks in the CHD signal with an amplitude superior to a given threshold are considered as transitions boundaries. Let us note that selecting a “good” threshold that can hold in all cases is not straightforward, because of variations in content, noise, object/camera motion…

In [Furth95], authors propose a similar approach, with the difference that here the color histograms are constructed in the HSV color space (Figure 2.7). The principle of the HSV representation [Cardani01] consists of separating the intensity (luminance) channel and the chromaticity information, which is described in terms of two components:

- Hue – represented as an angular coordinate in a cylindrical coordinate system,
- Saturation – which measures the degree of white of a given color

The advantage of such a representation is related to the invariance property of the hue component with respect lightning intensity changes. Moreover, the similarity measures associated with this representation are more consistent with the human visual perception [Su11].

However, in [Mas03] several tests and comparisons between the HSV and RGB are performed within the framework of shot boundary detection applications. Authors conclude that the associated performances are quite equivalent.

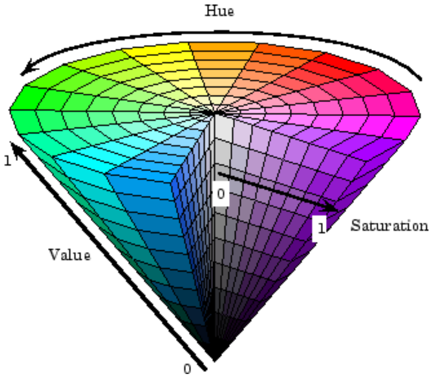


Figure 2.7. The HSV color space.

An extended evaluation of a significant number color spaces is presented in [Gargi00]. Color histograms are here computed for six different color spaces, including RGB, HSV, YIQ, L*a*b*, L*u*v* and Munsell. The following four different similarity measures are also retained for evaluation:

- Bin-to-bin histogram difference ($L_1$ distance):

$$D_{B2B} = \frac{1}{2XY} \cdot \sum_j \left| H_t(j) - H_{t-1'}(j) \right| \tag{2.25}$$

where is $X$ and $Y$ are respectively the image width and height, $j$ is the $j^{th}$ color in the representation (with respect to a considered color quantization scheme), and $H_t(j)$ is the number of pixels in frame *t* taking the color *j*.

- Chi-square test:

$$D_{\chi^2}(t,t-1) = \frac{1}{(XY)^2} \cdot \sum_j \frac{\left| H_t(j) - H_{t-1}(j) \right|^2}{H_{t-1}(j)} \tag{2.26}$$

Let us note that the $\chi^2$ test applied to image intensity histograms was one of the first similarity measures used for shot detection [Nagasaka92].

- Histogram intersection:

$$D_{HI} = 1 - \sum_j \frac{\min(H_t(j) - H_{t-1}(j))}{XY} \tag{2.27}$$

- The Kolmogorov-Smirnov measure, denoted by $D_{KS}$, which represents the maximum bin difference between cumulative histograms:

$$D_{KS}(t,t^{'}) = \max \left| CH_t(j) - CH_{t-1}(j) \right| \quad , \tag{2.28}$$

where $CH_t$ represents the cumulative histogram associated to the histogram $H_t$:

$$CH_t(j) = \sum_{0 \le i \le j} H_t(i) \tag{2.29}$$

The experimental results obtained demonstrated that the Munsell color space offers the best shot detection performances, at a moderate computational cost. The RGB, HSV, L*a*b*, L*u*v* color spaces showed quasi-equivalent performances. However, in [Gargi00] authors affirm that the choice of color space has less of an impact than the choice of the corresponding similarity measure and claim that the histogram intersection returns the best results. The chi-square test has significantly lower detection rates than the others and does not appear to be a suitable choice for coarse histogram comparison in the shot-change detection context. The Kolmogorov-Smirnov measure it is also insufficient to capture shot information, being completely inadequate for this application.

In [Nagasaka92], authors also propose a block-based histogram comparison, in order to obtain a more discriminative representation. Each video frame is divided into a set of non-overlapping blocks. The typical partition proposed is of 16 = 4 × 4 blocks. For each bloc, a histogram is associated to. The $\chi^2$ similarity measure is here exploited to compare different histograms, on a block by block basis. The regions corresponding to the eight largest differences are discarded, in order to reduce the effects of noise, object and camera motion.

In [Cernekova03], shot boundary detection is performed with the help of a singular value decomposition (SVD) applied to the individual frame histograms. For each frame $f_i, i = 1, \dots N$ of a video sequence, a $M$-dimensional feature vector $a_i$ is computed. Here, 3D normalized histograms, computed in the RGB color space, with 16 bins for each color component are considered. Thus, the dimensionality of feature vector is $16^3 = 4096$. Using $a_i$ as column an *NxM* matrix *A* is obtained:

$$A = [a_1] \dots [a_N] \tag{2.30}$$

The SVD of an *NxM* matrix *A* is a factorization of the form:

$$A = U \sum V^T \tag{2.31}$$

where *U* is an *NxM* column-orthogonal matrix (left singular vectors), *V* is an *NxN* column orthogonal matrix (right singular vectors) and $\sum = diag(\sigma_1, \dots, \sigma_R)$ is a diagonal matrix with non-negative elements (the singular values $\sigma_1 \geq \cdots \geq \sigma_R \geq 0$).

The color histograms (the column vectors of A) are projected onto the orthonormal basis formed by vectors of the left singular matrix *U*. The frame coordinates are given by the columns of $\sum V^T$. The row vectors (*i.e.* colors) of A are projected on the orthogonal basis using $V^T$and the coordinates are given by the rows of $U \sum$.

As a similarity measure the authors propose to exploit an angular correlation coefficient, defined as the cosine function of angle between the rows vectors $\widetilde{v_i}$ and $\widetilde{v_j}$:

$$\Phi(f_i, f_j) = \cos(\widetilde{v_i}, \widetilde{v_j}) = \frac{(\widetilde{v_i} \cdot \widetilde{v_j}^T)}{\|\widetilde{v_i}\| \cdot \|\widetilde{v_j}\|} \tag{2.32}$$

The technique proposed in [Cernekova06] is based on two parameters, corresponding to the mutual information (MI) and the joint entropy (JE) between successive frames $t$ and $t$+1 are determined, using the following set of relations:

$$MI_{t,t+1} \triangleq MI_{t,t+1}^{R} + MI_{t,t+1}^{G} + MI_{t,t+1}^{B} \qquad , \qquad (2.33)$$

where the mutual information for the red component $MI_{t,t+1}^{R}$ is computed as:

$$MI_{t,t+1}^{R} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C_{t,t+1}^{R}(i,j)\, log \frac{C_{t,t+1}^{R}(i,j)}{C_{t}^{R}(i)\cdot C_{t+1}^{R}(j)} \qquad , \qquad (2.34)$$

with $C_{t,t+1}^{R}(i,j)$ representing the probability that a pixel having a red level $i$ in frame $f_t$ to change into a red level $j$ in frame $f_{t+1}$.

The joint entropy is defined as:

$$H_{t,t+1} \triangleq H_{t,t+1}^{R} + H_{t,t+1}^{G} + H_{t,t+1}^{B} \qquad , \qquad (2.35)$$

where $H_{t,t+1}^{R}$is the joint entropy of the transition from $f_t$ to frame $f_{t+1}$, for the red component:

$$H_{t,t+1}^{R} = -\sum_{i=1}^{N-1}\sum_{j=0}^{N-1} C_{t,t+1}^{R}(i,j)\cdot \log(C_{t,t+1}^{R}(i,j)) \qquad . \qquad (2.36)$$

In equations above, $N$ denotes the number of quantization levels on each color component.

Small values of the parameter $\mathrm{MI}_{\mathrm{t,t+1}}$ indicate the existence of a cut between frames $f_t$ and $f_{t+1}$. Since $\mathrm{MI}_{\mathrm{t,t+1}}$decreases when the transmitted information from one frame to another is small (in case of fade transitions) the joint entropy (Equation (2.36)) is employed, in order to efficiently distinguish abrupt and fade transitions. The joint entropy measures the amount of information carried by the union of these frames. Therefore, its value decreases only during a fade, where a weak amount of inter-frame information is present.

The method offers the advantage of a high discriminative power, and of insensivity to translational, rotational, and zooming camera motions [Costaces06]. In particular, the method shows high detection rates (precision and recall rates of around 95% for abrupt transition and 83% in the case of fade transitions).

Another technique able to detect gradual transitions is based on the twin comparison algorithm presented in [Zhang93], which consists of comparing frame to frame color histogram differences with the help of two different thresholds. A first, higher threshold is used to determine cut effects while, a second lower one is representative of gradual transitions. The two thresholds are applied on a frame-to frame difference signal, computed based on color/gray level histograms. The HSV color space is here employed.

The method is highly sensitive to camera motions such as pan and zoom, which can generate an effect similar to the one caused by gradual transition. In order to improve the robustness of the proposed technique, authors introduce a new, user-dependent parameter, so-called called tolerance that allows the analysis of a number of consecutive frames before making a decision.

A third family of methods considers a content representation based on contour/edge features. Such methods are described in the following section.

#### *2.2.1.3. Methods based on edges/contours*

Edge/contour-based methods exploit the contour information present in the individual frames, under the assumption that the amount and location of edges between consecutive frames should not change drastically. Such methods aim at overcoming the limitations of color-based approaches, which suffer from illumination changes.

Thus, a cut transition produces a structural discontinuity at the image level. Based on this observation, in [Zabih95] authors propose an edge-based approach, which relies on the hypothesis that at the level of a shot boundary, new edges associated with the two shots are located at significantly different positions in the corresponding images.

The technique takes as inputs two consecutive frames *k* and (*k*+1). To detect the transition type the algorithm first performs a global motion compensation between successive frames [Zabih94]. The edges are detected with the help of the Canny's algorithm (Figure 2.8) which determines for each frame the associated binary image $e_k$ and $e_{k+l}$, respectively. Let $f_k^{in}$ denote the fraction of edge pixels in $e_{k+l}$ which are located at more than a fixed distance *r* from the closest pixel in $e_k$. Similarly, $f_k^{out}$ is the fraction of pixels in $e_k$ which are further than *r* from the closest pixel in $e_{k+l}$. The edge change ratio (ECR) parameter is defined as follows:

$$ECR_{k+l} = \max\left(\frac{f_k^{out}}{\sigma_k}, \frac{f_{k+l}^{in}}{\sigma_{k+l}}\right) \qquad , \qquad (2.37)$$

where $\sigma_k$ represents the number of edge pixels in the frame $k$.

Shot transitions are determined by examining the peaks in the EC sequence. A global threshold is considered. If the value of ECR exceeds the threshold, a scene brake is detected.

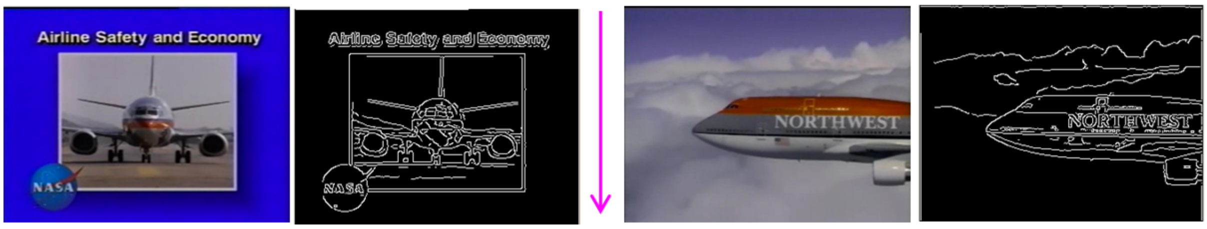


Figure 2.8. Transition detection based on edge variation.

A variation of the above approach is presented in [Lupatini98], where edge information (number of edge points that appears or disappears) is also used to detect scene breaks. Based on the two frames under analysis the algorithm firstly performs motion compensation using the block-matching technique introduced in [Liu93]. Next, camera/object motions of punctual changes in the image are eliminated by applying a low pass spatial filtering scheme. Edges are here extracted with the help of a Sobel operator. Finally, the difference between the numbers of edge points that change between two successive frames is determined. The method is more robust to camera and object motion than histogram-based approaches, but with an increase in the associated computational complexity.

The proposed technique is able to determine the type of transition existent between two video shots. If the ECR signal contains an isolated maximum, a hard cut is identified. Unlike hard cuts which lead to a single peak in the ECR signal, fades and other gradual transitions lead to an interval where ECR is elevated [Lienhart97]. In order to detect fade transitions two ECR parameters, corresponding to entering and exiting edges and respectively denoted by $ECR_k^{in}$ and $ECR_k^{out}$ are considered and analyzed. Let us note that during a fade-in (fade-out) transition, the number of entering (resp. exiting) edges is superior to the number of exiting (resp. entering) edges. In other words, a fade-in is identified if $ECR^{in} >> ECR^{out}$. Inversely, if $ECR^{out} >> ECR^{in}$ a fade-out transition is detected. All other maxima of ECR are automatically identified as dissolves.

Another approach is presented in [Yu97]. Authors exploit the fact that the ECR parameter should be close to zero at some instant, because either the initial or the final frame of a fade transition correspond to a blank image, with a number of edge pixels close to zero. Furthermore, the ECR parameter should present a gradual, monotonic variation. The method can be summarized by the following steps: first the frames are smothered using a spatial filter and then the $ECR_i$ between two consecutive frames $i$ and $(i+1)$ is computed. Next, the set of local minima int the ECR signal is detected. Solely the local minima with corresponding ECR value below a given threshold are here considered. For each retained local minimum, the first local maximum on the left ($i_0$) and right ($i_N$) sides of the minimum is determined. Then, if $\sum_{i\in(i_0,j)} \frac{ECR_i}{j-i_0}$ is smaller than a pre-establish threshold a fade-out transition is identified. Else, if $\sum_{i\in(i_0,j)} \frac{ECR_i}{i_N-j}$ is below a threshold then the transition is considered as fade-in.

During dissolves, object contours gradually disappear and new objects contours gradually show up. As a consequence, at the center of a dissolve transition the contour contrast becomes dimmer. The *ECR* parameter exploits the fact that the number of exiting edge pixels during the first part of a dissolve is large, while the number of entering edges pixels is large during the second half. For the dissolve detection the authors in [Yu97] define the double chromatic difference of a frame $f_i$ as follows:

$$DCD_{f_i} = \sum_{(x,y)} Th\left(\left|\frac{I(x,y,i_0)+I(x,y,i_N)}{2} - I(x,y,f_i)\right|\right) \quad , \qquad (2.38)$$

where $I(x,y,t)$is the intensity of pixel $(x,y)$ at time $t$ and Th(.) is a thresholding function. So, a dissolve is detected if the global minimum of $DCD_{f_i}$is bellow a fixed threshold.

The approach has high computational requirements but works well under the assumption that during the transition interval the sequence presents relatively slow motion.

A comparative study of both edge and histogram-based methods is proposed in [Lienhart01]. The following conclusions are highlighted:

- concerning the detection performance for hard cut transitions the color histogram-based algorithms generate better results than edge/contour techniques, with less computational time,
- the strength of the *ECR* feature comes from its ability of identifying any type of video transitions (fade, dissolve and wipe), but with a reduced performance (less than 70%.in precision and recall rates.

However, edge features are highly useful in removing the false alarms caused by abrupt illumination changes, as underlined in [Yuan07].

A fourth family of methods considers a set of features associated to the compressed and notably MPEG-compressed representation of videos.

#### *2.2.1.4. Temporal segmentation in the compressed domain*

Let us first mention the approach proposed in [Meng95], where the visual discontinuities are measured with the help of the MPEG coefficients stored in the compressed video stream.

The principle consists of analyzing the statistical characteristics of the Discrete Cosine Transform (DCT).

For an image block *f*(*x*, *y*) of size (*N* × *N*) pixels the DCT transform is defined as described by the following equation:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot cos\left[\frac{\pi(2x+1)u}{2N}\right]\cdot cos\left[\frac{\pi(2y+1)v}{2N}\right], \quad (2.39)$$

for $u, v = 0, \ldots, N-1$ and $\alpha(u), \alpha(v)$ defined as:

$$\alpha(u) = \begin{cases} \sqrt{1/N}, & for\ u = 0 \\ \sqrt{\frac{2}{N}}, & for\ u \neq 0 \end{cases} \quad (2.40)$$

In the case of MPEG videos, the DCT is computed on a (8 x8) block basis.

Two types of features can be used directly to detect shot change:

- The DC coefficients (*i.e*., corresponding to frequential parameters *u* and *v* equal to zero for each block), and corresponding to the average luminance/chrominance values of each block,
- The AC coefficients, which describe the signal variation within each block.

The set of all DC coefficients represent a down-sampled version of the original image. The majority of compressed-domain shot detection algorithms consider uniquely the DC coefficients.

Let us note that in this case, major savings in computational time can be achieved since such coefficients can be determined without completely decoding the entire video stream.

Another interesting parameter that can be exploited for shot boundary detection is the type of the MPEG encoded frame, which can be *I* (intra-coding mode), *P* (predictive coding mode) or *B* (bi-directional predictive coding mode) [Li04]. Such a parameter is used for shot detection purposes in [Calic02], [Pei00].

Another approach, introduced in [Fernando01], jointly exploits the number of macro-blocks encoded in B-mode and the motion vectors associated with each block in the MPEG-2 stream. The analysis is based on the statistical features (mean and variance) associated to DC coefficients in the compressed domain. However, the proposed method is implicitly based on the assumption that the video sequence is an ergodic process. This hypothesis is often violated in practice, with an impact on the associated shot detection performances.

In a general manner, shot boundary detection methods that directly exploit information included in the MPEG streams do not need to completely decompress the data and thus offer the advantage of speed. In particular, they are well suited for real-time applications. Such information makes it possible to detect discontinuities produced by both abrupt and gradual transitions. However, the associated detection performances are lower than the ones corresponding to methods that are performing the analysis in the original image domain.

A tradeoff between speed and detection performances can be achieved by considering a partial decoding solution, which makes it possible to carry out the analysis in both image and compressed domains [Bruyne08].

#### *2.2.1.5. Methods based on motion analysis*

Motion present in videos represents one of the major problems that impacts the performances of shot detection methods, because of the number of false alarms that are introduced in practice in the case of videos with a high amount of motion. The question to be solved can be stated as follows: how to distinguish between variations in the visual content due to camera/object motion and variations due to shot transitions?

In order to overcome such a problem, motion-based approaches have been proposed. The underlying hypothesis is that, in the case of a transition between two shots, a discontinuity in the associated motion flow is produced.

A block matching technique which aims at determining the motion between pairs of successive frames is proposed in [Lupatini98]. The resulting motion vectors are exploited within the framework of a motion compensation procedure. If the difference, in terms of motion compensation error, between the two considered frames is greater than a specified threshold, a shot boundary is identified. An average value of the luminance channel associated to each considered block is also considered, in the motion compensation stage, in order to ensure robustness to local motion.

A slightly different approach is proposed in [Porter00]. A spatial decomposition of video frames into non-overlapping blocks of 32 x 32 pixels is first performed. To each block from the current frame $k$, a best matching block is determined in frame $k+1$. The matching process is performed around a search windows centered on the considered block. The decision process takes into consideration the following two parameters: the normalized correlation between corresponding blocks and the location of the correlation coefficient with the largest magnitude. In order to reduce the computational complexity related to the computation of the normalized correlation function, a frequency domain technique is applied.

Advanced techniques that implement detection based on motion analysis use more sophisticate features to identify transition as: the changes affecting the optical flow [Akutsu92], the number and distribution of motion vectors [Lupatini98] as well as the strength of the residual [Shahraray95] given by the differences from the current and anterior position of the block.

However, as mentioned in [Gargi00], block matching methods remain inferior, in terms of performances to intensity/color histograms-based techniques. In addition, the motion estimation stage is a highly complex process, which limits the applicability of this family of approaches. Moreover, obtaining reliable motion estimations is still an open issue of research.

This concludes the analysis of how the main visual features are taken into consideration by the methods of the state of the art. Let us now analyze a second, important aspect, which concerns the specification of the temporal domain of the various continuity metrics involved.

### 2.2.2. Temporal domain of continuity metric

Whatever the visual features involved, the analysis process requires a comparison between frames within a given set. Such a set can be defined in various manners, as described here-below.

#### *2.2.2.1. Frame-to-Frame comparison*

The simplest way to detect a camera brake in a video stream is searching for high discontinuity values of a similarity measure between pairs of successive frames (denoted by $I(x,y,t)$ and $I(x,y,t+1)$), expressed as:

$$d_t = \Theta(F_t, F_{t+1}) \tag{2.41}$$

where $d_t$ denotes the similarity value at frame $t$, $\Theta$ is the similarity function, and $F_t$ and $F_{t+1}$ are the features considered in the two frames $I_t$ and $I_{t+1}$.

Pair-wise comparison schemes offer the advantage of simplicity. However, such approaches are highly sensitive to significant object/camera motion or in the case where different types of discontinuities occur. As a typical example, let us mention the flash lights that are present in certain videos and which lead to a punctual discontinuity that might generate false alarms. In order to overcome such a drawback, several optimizations have been proposed.

In [Zheng05] and [Leszczuk02], the second order derivative of the color histogram with 16 bins for each channel in the RGB color space was applied before analysis.

$$\Delta^2 F_t = \Delta F_{t+1} - \Delta F_t \tag{2.42}$$

where $F_t$denotes the features of the $t^{\text{th}}$ frame and $\Delta F_t = F_{t+1} - F_t$.

Authors have shown that using the second order derivatives can significantly improve the performances of the considered methods, in particular in the case of small object motions.

However, such approaches can provide only partial solutions to the problem, in some particular cases. A sounder framework to deal with such a problem is provided by methods performing the analysis within a sliding window, defined over a set of multiple frames.

### *2.2.2.2. N-frame window analysis*

The principle of the so-called *N*-frame window analysis approaches, first introduced in [Hanjalic02], is illustrated in Figure 2.9.

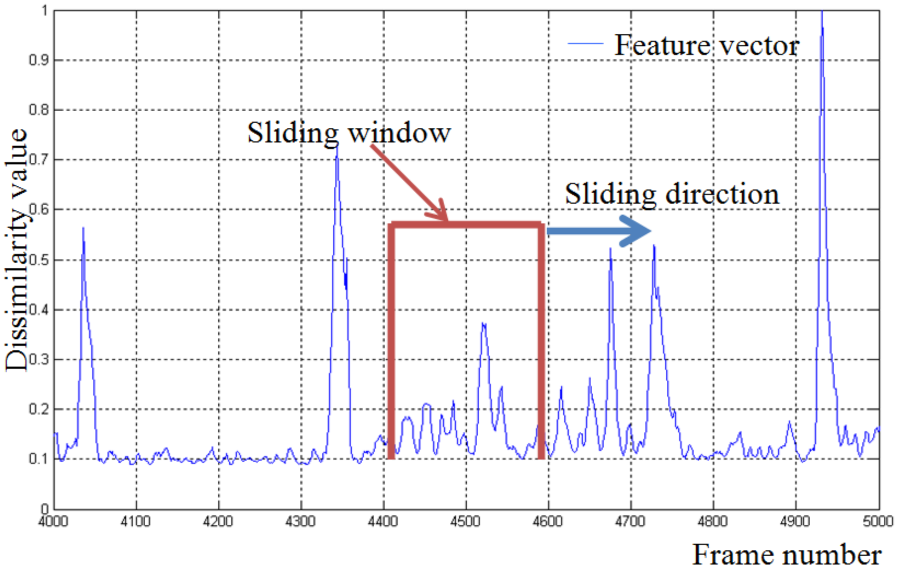


Figure 2.9. Illustration of the sliding window analysis approach.

In order to reduce the local or punctual variations in terms of dissimilarity value between successive frames, a set of frames, corresponding to a considered temporal interval (whose length gives the window size), is considered [Boccignone05].

The set of frames included in the analysis window is combined as described in [Cooper07]. First, low-level features are computed for each frame. The authors propose using the YUV color histograms ($F_t$) for $N$ frames. The chi-square similarity between images is used ($d$). Next, a similarity matrix is determined, which includes the similarity values between each two pairs of frames in the window. The similarity matrix, denote by *S*, is defined as described in the following equation:

$$S(i,j) = d\left(F_i, F_j\right) \tag{2.43}$$

Thus, the number of rows and columns of *S* is equal with the total number of frames included in the sliding window.

In this way, abrupt shot boundaries present highly distinctive similarity patterns in the matrix. Frames from the same shot are visually coherent, with high similarities, while frames from different, adjacent shots present low similarity. This produces a checkerboard pattern along the main diagonal of matrix *S*. Based on this observation the authors propose using matched filtering approaches (kernel cross-correlation) for boundary detection. The matched filter is a square kernel matrix, *K*, that represents the appearance of an ideal boundary in *S*. The frame index score is determined by correlating *K* along the main diagonal of the matrix *S*.

$$v(n) = \sum_{l=-L}^{L-1} \sum_{m=-L}^{L-1} K(l,m) S(n+l, n+m) \tag{2.44}$$

where $K(l,m)$ is a square matrix of size *2Lx2L*.

Different kernel functions can be considered, corresponding to scale-space analysis, diagonal cross-similarity, cross-similarity and full-similarity. They are detailed in [Cooper07].

Finally, the shot detection decision process is performed as follows. Maxima in the correlation measure (Equation (2.44)) correspond to locally novel frames and are good candidate for shot boundaries. The authors use them to form an intermediate-level feature set to comprehensively represent the local temporal structure and to perform boundary detection via supervised non-parametric classification. The temporal segmentation is formulated now as a temporal pattern classification for which they apply the k-nearest-neighbor (kNN) classifier.

A third approach to combine visual information from multiple frames consists of considering the whole temporal interval since the last detected shot boundary.

#### *2.2.2.3. Interval since last shot break*

In this case, the entire set of frames since the last detected transition and up to the current instant are considered and simultaneously analyzed.

Such approaches can be interpreted in the sense of a spatio-temporal image, so-called sequence matrix, summarizing the video flow.

An early example of such an approach is presented in [Han99]. The method starts by determining a feature vector for each image of the video sequence. Then, with the help of both horizontal and vertical projections of motion vectors the principal component dissimilarity vector is defined as:

$$D(j) = \sum_{i=2}^{N} |P(i,j) - P(i,j-1)| \ , \qquad (2.45)$$

where *P*(*L*x*M*) is the principal component of each feature vector and *N* is the total number of frames of the video stream.

The largest principal components *P(1,j)* are excluded to reduce significant changes in luminance level. In order to detect transitions the authors propose comparing $D(j)$ to a fixed threshold.

A so-called visual rhythm method is proposed in [Chung00]. Authors exploit the spatio-temporal image, constructed from pixels sampled uniformly along the main diagonal of each frame, to create different visual patterns adapted to various types of shot transitions. Cuts, wipes, and dissolves can then be identified by analyzing the resulting image. A cut appears as a vertical line, a wipe is a continuous curve, while a dissolve presents a linear increase or decrease of pixel values over a certain time interval.

In the above approach the spatial-temporal images are constructed in the spatial domain. Another technique [Guimaraes03] uses the visual rhythm variation by constructing the sequence matrix based on three measures: Soille’s gradient, multi-scale representation using ultimate erosion and thinning.

Whatever the features and similarity measures involved and whatever the type of the analysis interval used, it is necessary in all cases to threshold a similarity function in order to detect transitions. This aspect is discussed in the following section.

### 2.2.3. Thresholding and discontinuity detection

The simplest way is to consider a fixed, global threshold that is applied to the dissimilarity signal (Equation (2.43)) in order to detect shot breaks.

#### *2.2.3.1. Fixed threshold*

Selecting a “good” threshold is essential to detect transitions. The static threshold is the simplest decision method. In this case, the similarity/dissimilarity metric between pairs of consecutive frames is compared with the considered, fixed threshold (Figure 2.10), as described in the following equation.

$$SBD(t) = \begin{cases} 1, & if \ \left|S(x,y,t) - S(x,y,t+1)\right| < T \\ 0, & otherwise \end{cases} , \qquad (2.46)$$

where $SBD(t)$ is a binary variable associated with frame $t$ and indicating if frame $t$ corresponds to a shot boundary ($SBD(t) = 1$) or not ($SBD(t) = 0$).

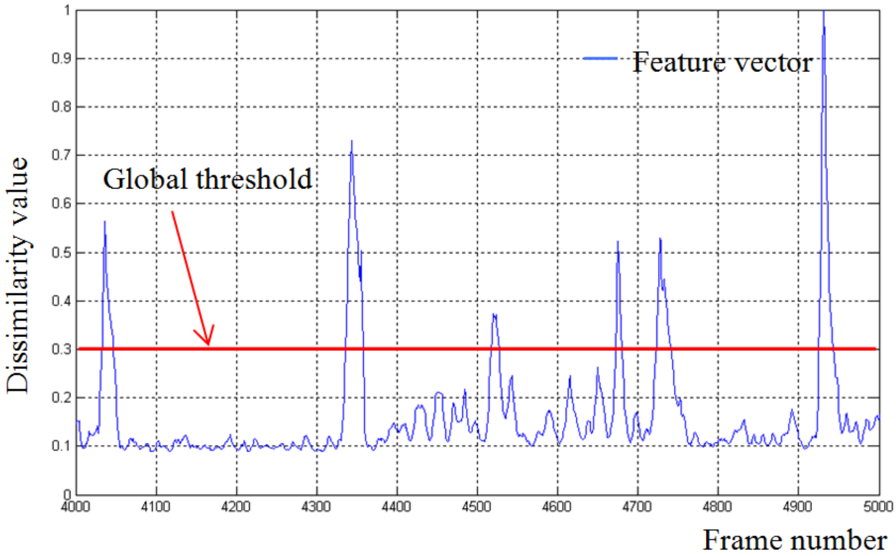


Figure 2.10. Transition detection based on a global threshold.

Most methods consider some heuristics in order to choose a global threshold. [Nagasaka92], [Arman93], [Fernando01]. However, it is impossible to determine a global threshold that may hold for all types and genres of videos.

An alternative solution is proposed in [Hanjalic02]. Here, the statistical distribution of the discontinuity values of pixels intensity variation between successive frames, within a fixed number of frames (*e.g.* 500) is measured. The obtained distribution is afterwards modeled by a Gaussian function with parameters $\mu$ and $\sigma$ , and the threshold value is computed as described in the following equation:

$$T = \mu + r \cdot \sigma, \qquad (2.47)$$

where $r$ is a parameter related to the user specified tolerated probability for false detection.

Such an approach takes into account the second order statistics of a video. However, the choice of parameter $r$ is not straightforward.

In order to overcome such limitations, in [Zabih95] authors require evaluating that the discontinuity values over the similarity signal obtained using pixels intensity variation.

In [Zhang93], authors first observe that *cut* transitions correspond to relatively high amplitude spikes in the feature dissimilarity signal, while gradual transitions lead to smaller amplitude values. The twin-comparison method proposed exploits this behavior by considering two global thresholds to detect transitions: a higher one to detect cut transitions and a lower one to detect any kind of gradual transitions. However, the approach suffers from inherent confusions between gradual transitions and variation due to camera/object motion.

In order to smooth the similarity signal, various pre-processing techniques can be utilized.

In [Otsuji94], authors consider morphological filtering with a flat structuring element of size $B$ frames. The dilation ($D_B\big(S(t)\big)$) and erosion $E_B\big(S(t)\big)$ of the similarity signal are computed as described in Equations (2.48) and (2.49).

$$D_B(S(t)) = \max S(t+r), \qquad r = -\frac{B}{2}, \pm 1, \dots, 0, \dots, \frac{B}{2} - 1, \tag{2.48}$$

$$E_B\big(S(t)\big) = \min S(t+r), \qquad r = -\frac{B-1}{2}, \pm 1, \dots, 0, \dots, \frac{B}{2}, \tag{2.49}$$

The authors propose applying morphological operators because in practice the erosion shrinks the peaks of the dissimilarity signal shorter than the structuring element, while the dilation produces the dual effect, enlarging the positive peaks. So, in this case a transition is detected by first applying a temporal closing that first eliminates negative spikes and afterwards a top-hat transform to extract positive peaks. The method brings some improvements especially when dealing with special effects such as animation, slow object or camera movement.

A nonlinear filtering approach is proposed in [Han99]. The objective is to distinguish between rapid, abrupt variation and smooth, gradual transition. Based on this constraint the authors propose introducing a progressive nonlinear filter (PNF) specifically designed for temporal segmentation. The PNF can suppress large variation in gradual transitions regions and preserve abrupt shot changes. The process involves two smoothing operation. In the first phase abrupt shot changes are preserved and noisy gradual transition are smoothed using the following equation:

$$F_i^1 = \frac{1}{N} \sum_{k=-N/2}^{k=N/2} F_i^0(k), \tag{2.50}$$

where $F_i$ are the features associated to frame *i*, and *N* is the size of the sliding window.

In the second stage a filtering operation eliminates insignificant impulse shot changes based on Equation (2.50).

$$F_i^2 = \begin{cases} \frac{1}{N} \sum_{k=-\frac{N}{2}}^{k=\frac{N}{2}} F_i^1(k)\,, & if\ \Delta F_i^1 < threshold \\ F_i^1\,, & otherwise \end{cases}, \tag{2.51}$$

where $\Delta F_i^1 = \left|F_i^1 - F_{i-1}^1\right|$. The method shows better performance than Gaussian, Wiener, and Wavelet-based filters.

In a general manner, algorithms based on global thresholds provide satisfactory results only in cases where the analyzed video exhibits similar characteristics over time in terms of content type and associated features. However, in practice, it is impossible to determine a unique, global threshold that can work with all kind of video material [Lienhart01]. Moreover, a same video document may include parts with highly different, both static and dynamic content. In such a case, considering a global threshold parameter at the level of the entire video is not appropriate.

#### *2.2.3.2. Adaptive threshold*

The adaptive thresholding is a natural solution to avoid the above-mentioned problems. The main principle consists of modifying the threshold within a sliding window, by taking into account the statistics of the similarity signal over time.

This principle is illustrated in Figure 2.11, where the average value of the similarity signal is considered. In a general manner, an adaptive threshold incorporates contextual information by considering the local activity of the content.

One of the first methods that adopt adaptive thresholding approach for shot boundary detection objectives is described [Yeo95]. The similarity signal at frame t is defined as:

$$S_{i,j}(t) = d\left(F_i, F_j\right), \qquad (2.52)$$

where $F_i$and $F_j$ are the feature sets extracted to characterize frames $I_i$and $I_j$.

A shot break is declared if the similarity measure between two consecutive frames satisfies the following conditions:

- (C1) - represents the maximum value within a symmetric sliding window of size $2N-1$, centered at the current frame $t$,
- (C2) - is t $\alpha$ times bigger than the second largest maximum in the sliding window, with $\alpha$ denoting a real-valued parameter.

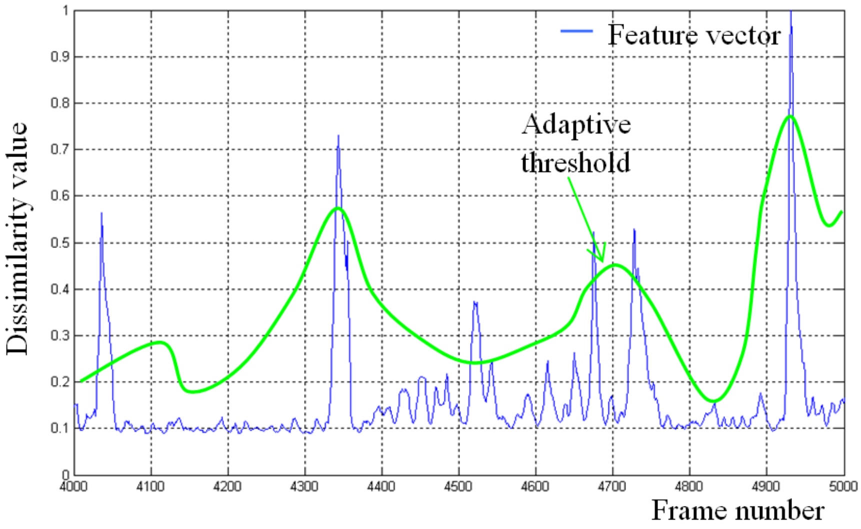


Figure 2.11. Transition detection based on an adaptive threshold.

The parameter $\alpha$ models, in a certain sense, the shape of the boundary pattern and implicitly incorporates the adaptive thresholding principle. Thus, a cut transition is characterized by an isolated, sharp peak in the similarity signal. Condition (C2) is useful to distinguish between shot boundaries and punctual distortions of the similarity signal, such as those caused by camera flashes: abrupt illumination introduces a rapid succession of at least two spikes.

An improved approach is proposed in [Hanjalic02]. The sliding window principle is here combined with a statistical analysis of the similarity samples within the considered window, which are modeled with the help of a Gaussian distribution. Instead of choosing a heuristic/empirical value, parameter $\alpha$ is determined indirectly, based on the pre-specified tolerable probability of falsely detected boundaries

Various authors have adopted adaptive thresholds in their approaches [Park05], [Robles04]. In [Park05] the authors propose to vary the threshold depending on the magnitude value of the motion vectors, while in [Robles04] the authors compute the threshold based on the average number of wavelet coefficients associated to each frame at different resolution.

In [Osian04], the evaluation of the similarity signal is performed within a sliding window of 15-20 frames. Several parameters are computed for each position of the considered window, including global motion compensation computed for each pair of frames and an affine transformation which warps the first frame into the second. The resulting difference is compared against an adaptive threshold, determined with the help of second order statistics.

A modified version of this approach is proposed in [Cernekova06]. Here, the average value of the difference signal is computed within the sliding window, without considering the current frame. The similarity value associated to the current frame is then compared with the average value. If their ratio exceeds a given threshold, a shot boundary is declared.

In [Urhan07], authors combine two different thresholds: a global and a local one. If the similarity value is superior to the global threshold, a cut transition is identified. In the other case, a local threshold, defined as the average of the dissimilarity signal within a sliding window, is computed. If the similarity value is above the local threshold than a gradual transition is identified.

The computation of adaptive thresholds/local statistical measures naturally leads to an increase in computational time. In order to speed-up, the process, in [Drew99], authors propose to perform the analysis at a coarser temporal resolution, obtained by sub-sampling the video. A fixed sampling step of 32 frames is here used.

A similar approach is also proposed in [Bescos00]. In order to detect gradual transitions, multiple sampling intervals, varying from 5 to 30 frames are here used (Figure 2.12). This leads to a multi-scale analysis process. A shot transition introduces a peak in the dissimilarity sequence at a given scale, and flatter variation patterns at the other scales. In the case of noise caused by large object displacement, camera movement or abrupt illumination two different peaks will result at all scales. This nice property is used to distinguish between transitions and the other events, thus increasing the detection efficiency.

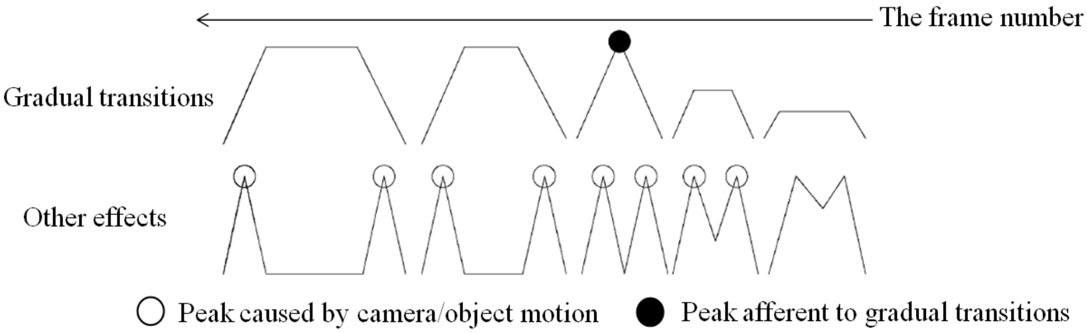

The frame number
Gradual transitions
Other effects
○ Peak caused by camera/object motion
● Peak afferent to gradual transitions

Figure 2.12. Transition detection using the technique [Bescos00].

Methods involving adaptive thresholds overcome a lot of the problems related to a global analysis and significantly increase the detection efficiency. However, the size of the analysis window is a highly important parameter. For relatively small values, it is impossible to capture all the dynamics of a gradual transition. In most cases, this will generate a missed detection. For large window size, multiple shots with different characteristics may be considered together, which leads to the same drawbacks as in the case of global thresholding. Typical values of window sizes encountered in the literature range in the 5-35 frames interval.

However, a “good” compromise should be determined, on an empirical basis or by injecting some strong *a priori* information within the analysis process. This can be done by considering a learning process, as in the case of the classification-based approaches described in the following section.

#### *2.2.3.3. Trained classifiers*

In order to overcome limitations related to threshold selection and parameter setting, machine learning techniques are based on a radically different detection strategy. The principle consists of considering supervised classification techniques, able to learn the considered transitions from a ground truth learning set and to generalize them to unknown video signals.

Among the classification techniques that can be uses to achieve such an objective, let us cite rule induction [Feng08], support vector machines [Anguita10], boosting algorithms (AdaBoost) [Freund97]…

Recent approaches [Yuan07], [Matsumoto06], [Chasanis08] use a support vector machine (SVM) classifier [Burges06] in order to both:

- Locate shot boundaries, *i.e.* provide, for each frame, a binary decision indicating if the considered frame correspond or not to a shot transition,
- Identify the corresponding transition type.

If we consider a set of training points $\{x_1,\ x_2, \dots, x_n\}$, where each input has $N$ attributes, the SVM technique divides the input data in two classes $y_i = -1$ or $y_i = +1$. In this case the data is considered linearly separable. Most methods are based on binary decisions and use a set of machine for detection (*e.g.* one to differentiate between abrupt and gradual transitions, one to distinguish dissolves from fades…).

In [Matsumoto06], features from both compressed (MPEG) and uncompressed domains are combined into a multi-dimensional global feature vector. The method is able to identify various types of transitions, including cuts, dissolves with various transition spans….

A SVM shot boundary detection strategy is also proposed in [Yuan07]. Multiples classifiers are manually trained with positive and negative examples for each type of transition. In this context, cuts are distinguished from the others via the shape of the valley in the similarity signal considered. Because a gradual transition may span a varying temporal length, authors affirm that it is almost impossible to predict all possible situations. Moreover, after analyzing the experimental results they claim that a more appropriate solution is to consider an adaptive threshold.

In [Chasanis08], authors consider a multi-scale representation, obtained by computing inter-frame dissimilarities at three sampling intervals of 2, 3 and 6 frames. A SVM classifier is here again considered in order to categorize the feature vectors, associated to each image of the video flow, in three classes corresponding to so-called normal sequences (*i.e.*, succession of frames without transitions), abrupt cuts and gradual transitions.

Classification-based approaches offer powerful solutions which make possible to:

1. Consider and combine various types of features for an increased discriminative power,
2. Inject a considerable amount of *a priori* knowledge in the shot detection process, with the help of the ground truth used in the training step.

However, in practice, the associated performances are strongly dependent of the learning set used to train the classifiers. Such a data set should be sufficiently important in terms of size, and variability of content/transition types, in order to ensure good generalization properties.

However, constructing such a ground truth data set requires a huge amount of human interaction. In addition, the categories of transitions and, more generally, the various types of video segments have to be carefully identified and taken into account for a successful classification process.

### 2.2.4. Performance evaluation

An important issue to be solved when considering the issue of shot boundary detection methods is the constitution of a representative test data set that can serve as ground truth for the various techniques. Such a ground truth needs to include a large variety of videos, with a wide range of content and transitions types. In addition, such a data set, that can include thousands of videos for a real-life set-up, has to be manually segmented by human experts, which is a tedious and time-consuming process. Moreover, precisely determining the transitions boundaries in the case of some gradual transitions is not straightforward and may depend of the subjective user perception.

The lack of common, widely recognized by the scientific community, and publicly available benchmark data set is a first and major problem encountered when evaluating shot boundary detection methods.

Numerous methods introduced in the literature propose evaluations on more or less *ad-hoc* data sets, which limit the pertinence of the obtained results. In addition, such test sets are generally not publicly available. As a consequence, objective comparisons with different methods are not possible.

Still, notable efforts have been made during the last decade, within the framework of various national and international evaluation campaigns dedicated, in general, to video processing/analysis techniques. As notable examples, let us mention the so-called ARGOS French campaign – “Campagne d'évaluation d'outils de surveillance de contenus video” [Joly07], and TRECVID – “Video Retrieval Evaluation” [http://trecvid.nist.gov]. Such campaigns propose to standardize the evaluation process, which is carried out as a contest, by developing a unique data base for tests and ground truth and by establishing a set of performance evaluation criteria.

In particular, the TRECVID approach consists of evaluating each submitted method against a large test set of commercial videos for which the ground truth data is manually established. Unfortunately, the TRECVID corpus is restricted only to the participants.

Concerning the performance evaluation metrics, two types of detection errors can be encountered:

- The first one corresponds to the so-called *missed detections* (or *false negatives*) and relates to the case where the detection failed for some transitions present in the video;
- The second error corresponds to the situation where some false shot boundaries have been erroneously detected. Such cases are called *false alarms* (or *false positives*).

When a ground truth test data set is available, such detection errors can be counted and globally described with the help of two error parameters, denoted by *MD* and *FD*, and respectively representing the numbers of missed detected and false detected transitions. Let us also denote by *D* the total number of correctly detected transitions. Based on these entities, the most often popular evaluation measures encountered in the literature are the so-called recall (*R*) and precision (*P*) rates, respectively defined as described in Equations (2.53) and (2.54):

$$Recall = \frac{D}{D + MD} \tag{2.53}$$

$$Precision = \frac{D}{D + FA} \tag{2.54}$$

The precision is given by the total number of correctly detected transitions divided by the total number of detected transitions (*i.e.* the sum of correctly detected transitions and false alarms). The recall parameter is defined as the number of true positives divided by the total number of elements that actually should be identified as shot boundaries (*i.e*. the sum of correctly detected and missed detected transitions). Precision can be interpreted as a measure of exactness or fidelity, whereas recall is a measure of completeness.

The recall and precision rates can be combined within a unique evaluation metric, denoted by F1 and defined as the harmonic mean of precision and recall rates:

$$F1 = 2 * \frac{P \cdot R}{P + R} \tag{2.55}$$

Ideally, the recall, precision and $F1$measures should be equal to 1, which corresponds to the case where all existing shot boundaries are correctly detected, without neither false alarms nor missed detections.

In [Sethi95], authors suggest that false-alarm errors should be ignored entirely, because in the author’s opinion the correct identification of shot boundaries is more important than any false alarms. Moreover, they argue that the false alarms are determined either by camera/object motion or by camera processing operations (*e.g.* zoom in/out, pan…) and it is natural to consider that part as distinct. In the view of [Gargi00], this is not desirable, since under such an evaluation scheme, an algorithm that detects a shot transition at every single frame would outperform a more conservative one.

The evaluation might also depend on the considered applications. For example, since shot detection is needed in the case of semantic video compression [Cotsaces06], too many shot changes will lower the efficiency of the representation.

Moreover, in the case of video summarization applications, false alarms are not desirable since the objective is to obtain compact representations, where the redundancies are minimized. Even with perfect precision, some types of video sequences have shot changes every few seconds, increasing this with false alarms, the resultant video summary would become useless for the end user.

Today, the widely recognized measures for evaluating shot boundary detection methods are the recall, precision and F1 rates. They will be subsequently utilized in our work.

## 2.3. THE PROPOSED SHOT BOUNDARY DETECTION SYSTEM

The proposed shot boundary detection method exploits and extends the graph partition (GP) model introduced in [Yuan07]. Let us first recall the basic principle of the graph-based representation model.

### 2.3.1. Graph partition method

Various real-world or physical situations can be mathematically modeled with the help of graphs. Let us first introduce the basic definitions necessary for our future developments.

#### *2.3.1.1. Background*

A graph $G$ is by definition an ordered triple $(V(G), E(G), \psi_G)$ consisting of a nonempty set of *vertices* (or *nodes*) $V(G)$, a set of *edges* $E(G)$ and an incidence function $\psi_G$ that associates with each edge of $E$ an unordered pair of (not necessarily distinct) vertices of $G$. If $e$ is an edge and $v_i$and $v_j$ are vertices such that $\psi_G(e_{i,j}) = \{v_i, v_j\}$, then $e_{i,j}$ is said to join $v_i$ and $v_j$ while the vertices are called the ends of $e_{ij}$ [Hendrickson00]. Two vertices incident to a common edge are called adjacent, or neighbors. Analogously, two edges sharing a common vertex are also called adjacent.

The number of vertices (edges) is denoted by $|V|$ (resp. $|E|$).

Let us illustrate this concept with the help of the following simple example. Let $G = (V(G), E(G), \psi_G)$ be the considered graph, with $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$ and $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ . The incidence function $\psi_G$ is defined by:

$$\psi_G(e_1) = \{v_1, v_2\},\ \psi_G(e_2) = \{v_1, v_1\},\ \psi_G(e_3) = \{v_2, v_3\},\ \psi_G(e_4) = \{v_3, v_4\}$$
$$\psi_G(e_5) = \{v_2, v_4\},\ \psi_G(e_6) = \{v_3, v_5\},\ \psi_G(e_7) = \{v_1, v_4\},\ \psi_G(e_8) = \{v_4, v_5\}$$

The diagram associated with the planar graph $G$ is illustrated in Figure 2.13.

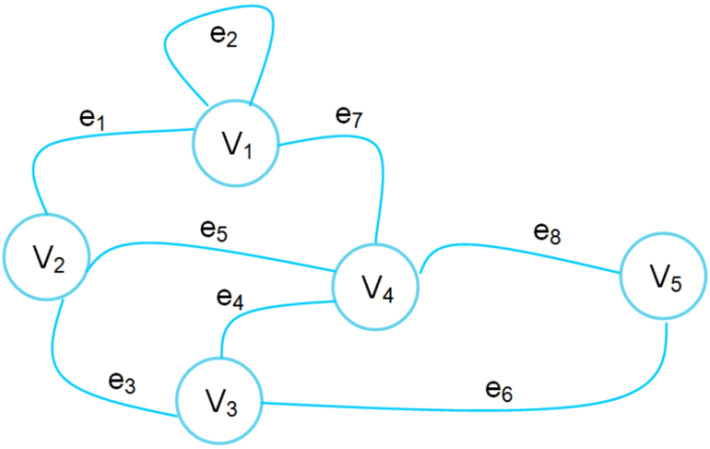


Figure 2.13. The associated diagram for a graph G.

An edge having two distinct ends is called a link, while in the case of identical ends is said to form a *loop* (the case of $e_2$ in Figure 2.13). A graph is called *simple* if it has no loops in its structure and no two links join the same pair of vertices.

The graph vertices may represent distinct entities, while the edges encode data dependencies. To each edge $e_{ij}$ , a weight $w_{ij}$ may be associated to. Such a weight might represent, for example, a degree of similarity between nodes *i* and *j*.

The connectivity of a given graph can also be described with the help of the so-called adjacency matrix, denoted by *A*. The adjacency matrix is constructed based on the vertices (or nodes) of a graph that are adjacent to which other vertices. For the graph presented in Figure 2.13 the associated adjacency matrix is described as follows:

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} , \tag{2.56}$$

Let us note that the adjacency matrix is symmetric, of size $|V| \times |V|$.

Let us now introduce the problem of optimal graph cuts.

For a given graph *G* with weights associated to each edge, the objective is to partition the nodes of *G* into two non-empty and disjoint subsets *A* and *B* such that a total cost function associated to the considered partition is minimized. Such a partition can be simply obtained by cutting the graph along a set of edges (*i.e.*, removing edges connecting the two parts *A* and *B*).

A graph partition application requires the specifications of the following issues:

- What is the best criterion that the partition has to satisfy?
- How such a partition can be computed efficiently?

In the literature, various partition criteria have been defined. Early studies proposed to use as an objective function the so-called minimum cut [Kernighan70]. In this case if we consider a set of edges *E* defined as: $E = \{(v_i, v_j) \mid v_i \in A, v_j \in B \; and \; (v_i, v_j) = e_{i,j}\}$ the cost function associated to the partition (*A*, *B*) is computed as:

$$cut(A,B) = \sum_{(v_i,v_j)\in E} w_{i,j}\,(v_i, v_j) \quad , \tag{2.57}$$

The cut function represents the sum of weights associated to edges that have been removed.

The objective then is to determine the partition minimizing Equation (2.57) (Figure 2.14). Although there are an exponential number of possible partitions, efficient minimization algorithms exist [Kolmogorov06].

However it has been proven that usually such an approach leads to skewed cuts [Yuan07] since the functional in Equation (2.57) tends to introduce a bias and favors small cuts corresponding to some isolated nodes in the graph.

In order the overcome such limitations, several other objective functions have been introduced more recently, including the ratio cut [Wang03], the normalized cut [Shi00] and the min-max cut [Ding01].

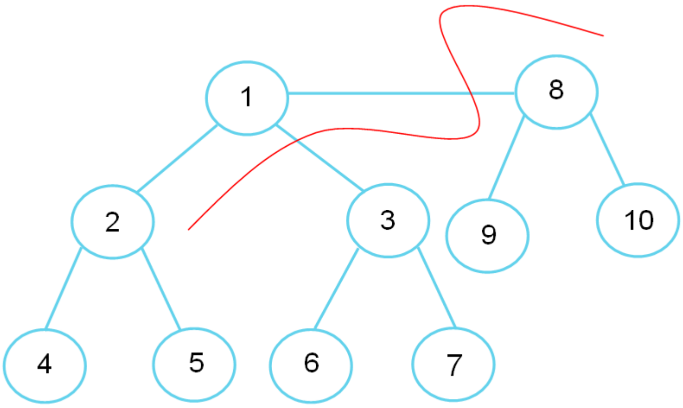


Figure 2.14. Graph partition model - schematic representation.

Such strategies require the definition of two extra measures called the subgraph association for $A$ and $B$, expressing the total connection between the vertices in $A$ to all the other nodes in the considered graph. If we consider two sets $E_1 = \{(v_i,)v_j \mid v_i \in A, v_j \in V\ and\ (v_i, v_j) = e_{i,j}\}$ and $E_2 = \{(v_i,)v_j \mid v_i \in V\ v_j \in B\ and\ (v_i, v_j) = e_{i,j}\}$, then the associations are defined as:

$$assoc(A,V) = \sum_{(v_i,v_j)\in E_1} w_{i,j}(v_i, v_j)\ , \tag{2.58}$$

$$assoc(B,V) = \sum_{(v_i,v_j)\in E_2} w_{i,j}(v_i, v_j), \tag{2.59}$$

One of the most popular approaches consists of, considering an objective function defined based on the dissociation between measures. This leads to the so-called *normalized cut* objective function, defined as:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \tag{2.60}$$

The normalized cut function makes it possible to overcome the limitations of the minimal cut, which tends to yields small sets of isolated nodes in the graphs. The normalized cut guarantees that a reduced number of isolated points will not determine small *Ncut* values, because in this case the cut is expressed as the total connection between the reference set of vertices and all the other nodes in the original structure.

In a similar manner, a new measure can be defined for the total association, called *normalized association* within groups for a given partition and defined as:

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \tag{2.61}$$

The normalized associations *Nassoc(A, B)* expresses how tightly the nodes clustered in a specific class are connected to other nodes grouped in other classes.

Finally, let us note that the normalized association and normalized cut functions are inter-related, and described in Equation (2.62):

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} = \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)} =$$

$$\ldots = 2 - \left( \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \right) = 2 - Nassoc(A,B) \qquad (2.62)$$

In our work, we have adopted a slightly different approach, corresponding to the so-called min-max objective function, introduced in [Ding01] and defined as:

$$Mcut(A,B) = \frac{cut(A,B)}{assoc(A,A)} + \frac{cut(A,B)}{assoc(B,B)} \qquad (2.63)$$

When graphs are used for video temporal segmentation, the pairs of weights associated with each edge $e_{i,j}$, are strictly positive $w_{i,j}(v_i, v_j) > 0$, and express the similarity/dissimilarity between two frames. Let us now describe how such a graph-based representation approach can be used for temporal segmentation purposes.

### *2.3.1.2. Min-Max graph cut for temporal video segmentation*

In this case, the graph *G* is constructed as follows. For each individual video frame *n*, a sliding window of size *N*, centered on the current frame is considered. To each frame in the current window, a node in the graph is associated to. Thus the set of nodes, denoted by $G_n$ will correspond to the set of frames in the considered window. The set of edges is defined by interconnecting each node to each other, in an exhaustive manner. The corresponding weights will correspond to a measure of visual similarity between corresponding frames.

A particular case of cuts is here considered. They correspond to the partition of set $G_n$ into two sets, denoted by $A_n^k$ and $B_n^k$ and associated to a frame *k* in the considered temporal interval. More precisely, $A_n^k$ and $B_n^k$ respectively represent the sets of nodes (frames) anterior and posterior to frame *k*.

The min-max objective function in Equation (2.63) becomes now:

$$Mcut(A_n^k, B_n^k) = \frac{cut(A_n^k, B_n^k)}{assoc(A_n^k, A_n^k)} + \frac{cut(A_n^k, B_n^k)}{assoc(B_n^k, B_n^k)} \qquad (2.64)$$

The *cut* and *assoc* functions can be written as described in Equations (2.67) and (2.68).

$$cut(A_n^k,\ B_n^k) = \sum_{i \in A_n^k, j \in B_n^k} w_{i,j}(v_i, v_j) \qquad (2.65)$$

$$assoc(A_n^k) = \sum_{i,j \in A_n^k} w_{i,j}(v_i, v_j); \quad assoc(B_n^k) = \sum_{i,j \in B_n^k} w_{i,j}(v_i, v_j) \qquad (2.66)$$

The analysis thus performed makes it possible to construct a local dissimilarity vector *v(n)* that stores, for each frame *n* the associated optimal cut measure $Mcut(A_n^k, B_n^k)$ (minimum value of the objective function defined in Equation (2.61)). For each frame *n*, *v*(*n*) is defined as:

$$v(n) = \min_{k\in\{1,\ldots,N-1\}} \{Mcut(A_n^k, B_n^k)\} \tag{2.67}$$

Figure 2.15 illustrates the dissimilarity vector obtained for a given video, with various sizes of the analysis window, ranging from 10 to 35 frames.

We can observer that larger window sizes lead to smoother dissimilarity vectors. The window size should be large enough to capture usual transitions that are greater than 10–15 frames. In our work, we have considered a value of $N = 25$ frames (Figure 2.15), that assures a good compromise between detection precision and computational time.

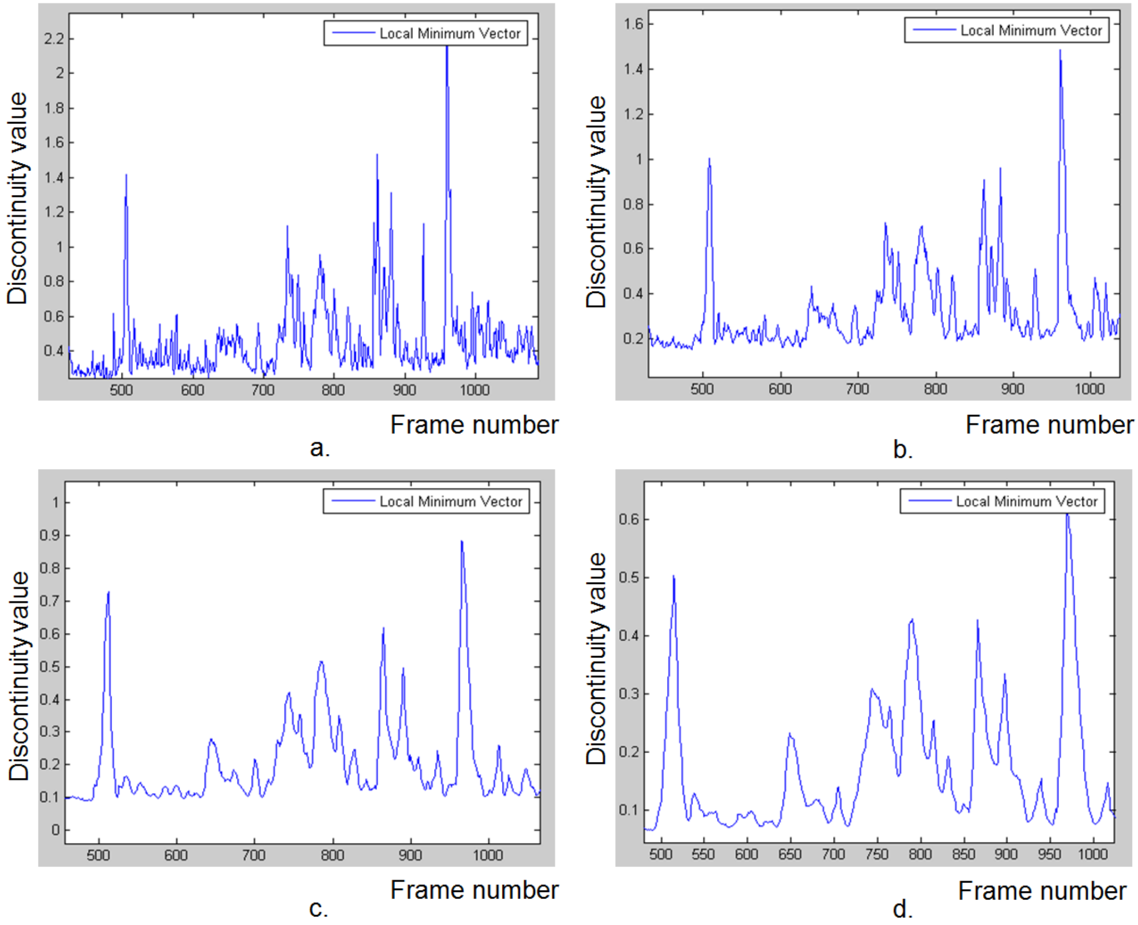


Figure 2.15. Local minimum vector variation with different window sizes:
(a) 10 frames; (b) 15 frame; (c) 25 frames; (d) 35 frames.

Concerning the weights $w_{i,j}$, they are defined as the chi-square distance between color histograms associated to corresponding frames, represented in the HSV color space.

$$w_{i,j} = \sum_k \frac{\left(H_k^i - H_k^j\right)}{H_k^i + H_k^j} \times e^{|i-j|} \quad , \tag{2.68}$$

where $\{H_k^i\}_{k=1}^{Nbins}$ denotes the HSV color histogram associate to frame $i$. The histogram can be mathematically expressed:

$$\{H_k^i\}_{k=1}^{Nbins} = \frac{1}{X \cdot Y} \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} \delta(I(i,j) - c) \qquad (2.69)$$

where $X \cdot Y$ is the frame size, $k$ is the current color index, $I(i, j)$ is the color of the pixel $(i, j)$ in the image $I$ and $d(x)$ is the Dirac function.

The exponential term in Equation (2.68) is used in order to take into account the temporal distance between frames [Ţapu10].

In our work, we have considered the HSV color space, in order to take advantage of the Hue invariance to common (weak) lightning intensity changes.

The video graph *G* is represented with the help of an *N x N* symmetric matrix, denoted by *S*, which stores the distances between the *N* nodes (frames) of the graph and which is defined as:

$$S = \begin{pmatrix} w_{1,1}, w_{1,2}, \dots, w_{1,N} \\ w_{2,1}, w_{2,2}, \dots, w_{2,N} \\ \dots \\ w_{N,1}, w_{N,2}, \dots, w_{N,N} \end{pmatrix}, \qquad (2.70)$$

The similarity matrix stores the cut and association values, as illustrated in Figure 2.16.

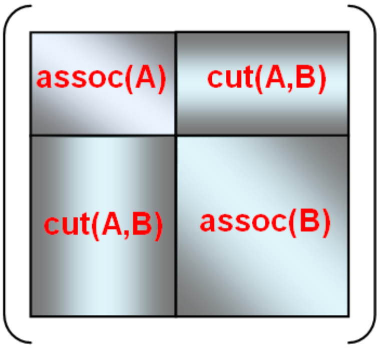


Figure 2.16. The similarity matrix *S* associated with graph *G*.

When displacing the analysis window from frame to frame, a new frame of the video is each time considered for analysis. The similarities between current image and all the other frames in the window are determined and stored in the matrix. At each iteration, the values of the similarity matrix are permuted from left to right allowing a continuous update in the graph partition (Figure 2.17).

After computing the local minimum vector for the input video stream the simplest way to identify a camera brake is determine if the discontinuity value:

1. exceeds a pre-established threshold, and
2. represents a maximum within the considered window [Ţapu11].

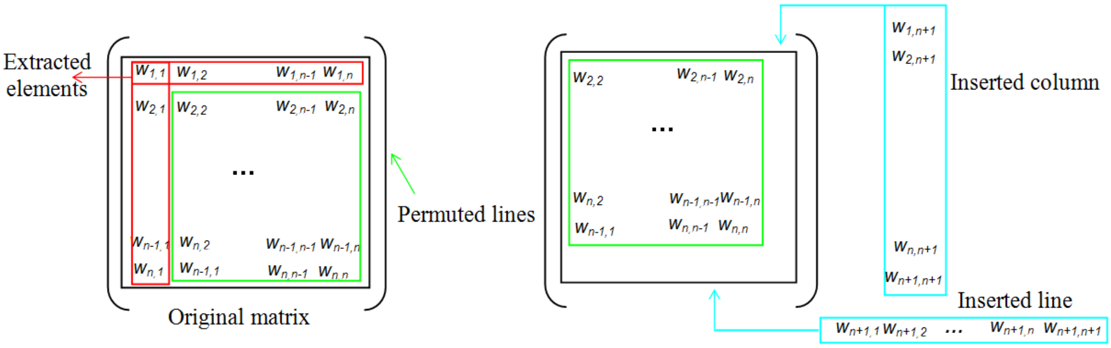

Extracted elements
$w_{1,1}$ $w_{1,2}$ $w_{1,n-1}$ $w_{1,n}$
$w_{2,1}$ $w_{2,2}$ $w_{2,n-1}$ $w_{2,n}$
...
$w_{n-1,1}$ $w_{n,2}$ $w_{n-1,n-1}$ $w_{n-1,n}$
$w_{n,1}$ $w_{n-1,1}$ $w_{n,n-1}$ $w_{n,n}$
Original matrix
Permuted lines
$w_{2,2}$ $w_{2,n-1}$ $w_{2,n}$
...
$w_{n,2}$ $w_{n-1,n-1}$ $w_{n-1,n}$
$w_{n-1,1}$ $w_{n,n-1}$ $w_{n,n}$
$w_{1,n+1}$
$w_{2,n+1}$
Inserted column
$w_{n,n+1}$
$w_{n+1,n+1}$
Inserted line
$w_{n+1,1}$ $w_{n+1,2}$ ... $w_{n+1,n}$ $w_{n+1,n+1}$

Figure 2.17. Updating process for the similarity matrix.

If both conditions are accomplished, then a shot boundary is identified. This process is illustrated in Figure 2.18.

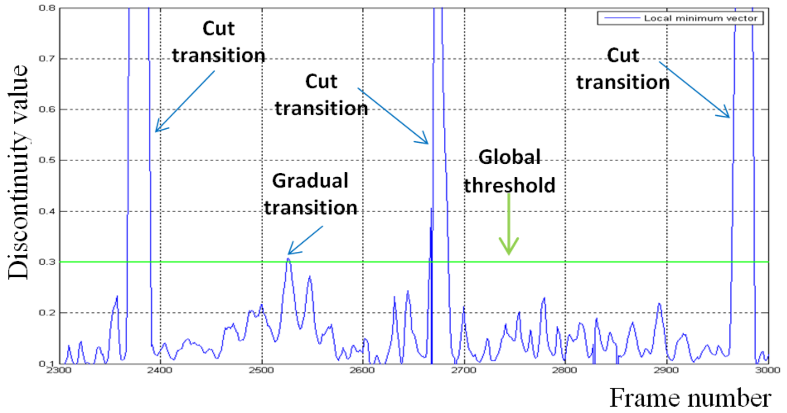
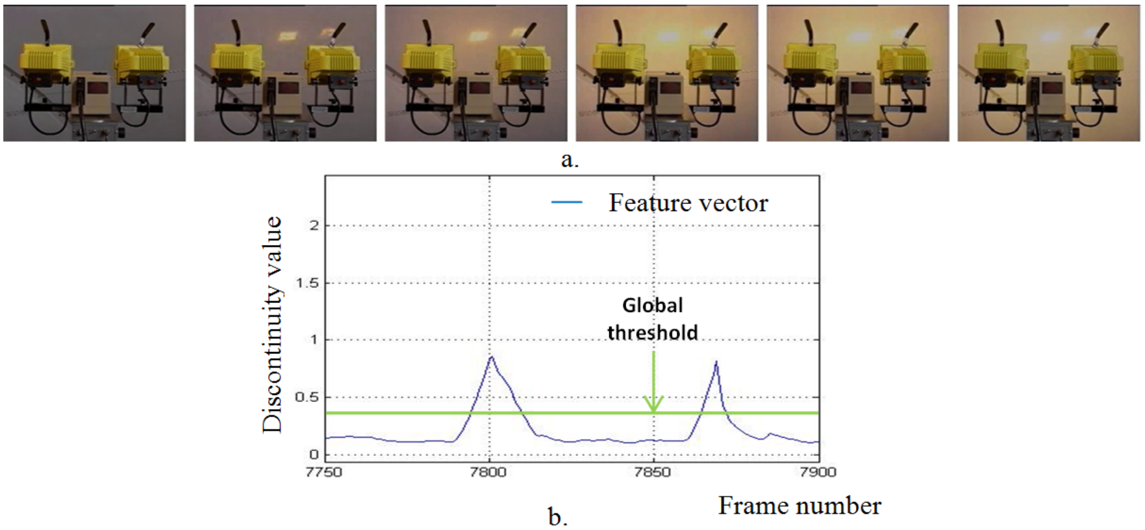


Figure 2.18. Transition detection using the local minimum vector.

However, determining an appropriate and global threshold is a difficult issue, since in practice, shots might exhibit camera or large object motions which can lead to both false alarms and miss detection of gradual transitions. This problem is illustrated in Figures 2.19, 2.20 and 2.21, where some false alarms, due to large camera motion and flash lights are detected, in the case of a real-life video document. Figure 2.19 presents the variation of the local minimum vector due to a flash light.


a.

b.

Figure 2.19. Local minimum vector variation for due to a flashlight: (a) Video sequence; (b) Local minimum vector variation.

In the case of large object movement the variations of the local minimum vector are not that pronounced as for flashlights as it can be observed in Figure 2.20. The camera motion has a similar impact on the local minimum vector (Figure 2.21).

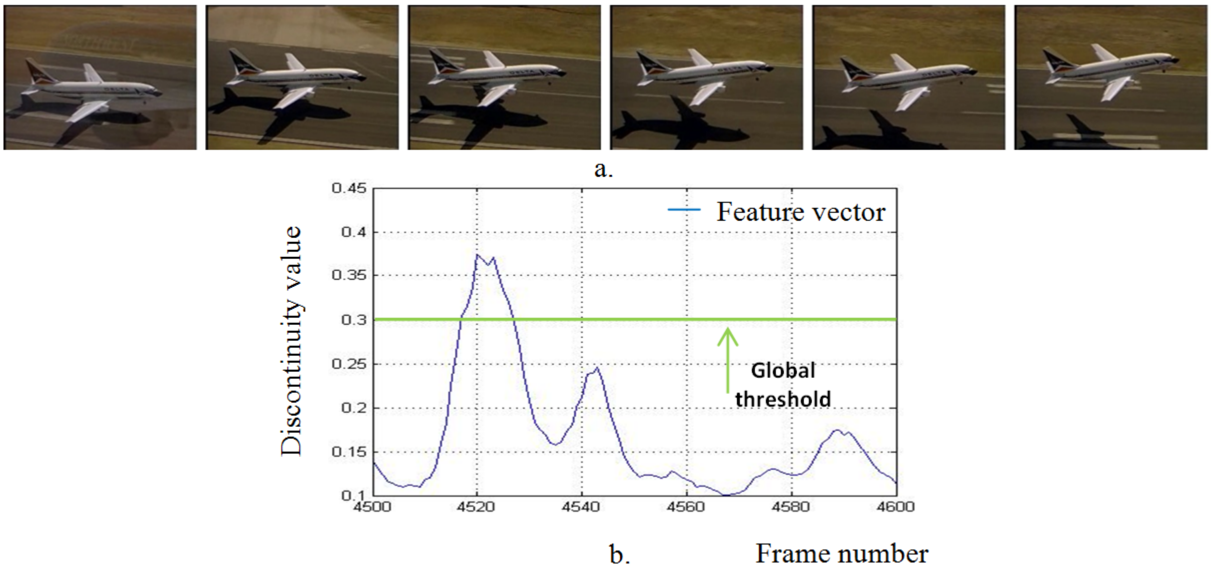


Figure 2.20. Local minimum vector variation due to large object displacement:
(a) Video sequence; (b) Local minimum vector.

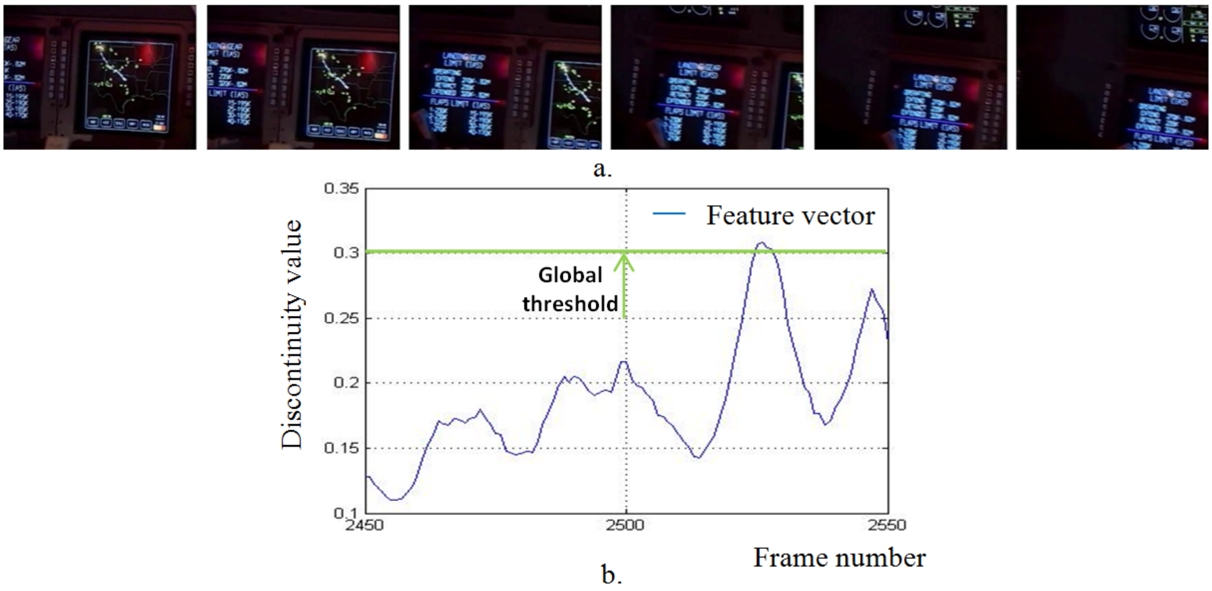


Figure 2.21. Local minimum vector variation due to camera movement:
(a) Video sequence; (b) Local minimum vector.

As it can be observed, the main limitation is related to the threshold parameter, the sensitivity to camera and object motion being very strong (Figure 2.22).

A current way to solve this problem is to replace it by an adaptive threshold inside a moving window. Introducing an adaptive threshold helps to decrease this sensitivity in a certain way, but obviously not completely. In this case, the window size becomes the main parameter to be tuned, directly influencing the detection performances.

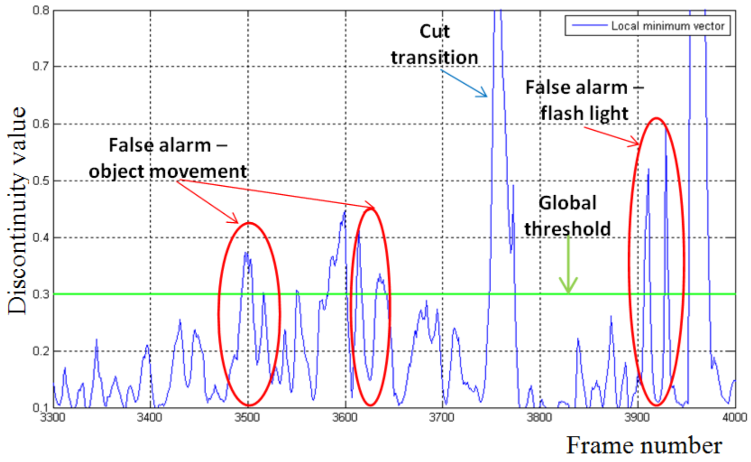


Figure 2.22. Temporal segmentation based on the local minimum vector.

In order to overcome such limitations, and to be able to deal with heterogeneous videos, we propose to perform a pre-processing step, aiming at reducing the possible perturbations that can present the local minimum vector $v(n)$.

This pre-processing step is based on a non-linear filtering in the space of scale space derivatives of the dissimilarity vector $v(n)$.

### 2.3.2. Pre-processing with scale-space derivatives

After applying the graph partition on the input video flow and optimizing the min-max objective function we propose to perform the shot boundary detection on the derivatives of the minimum vector $v(n)$.

In [Lefevre07], authors showed that the exploitation of the first order derivatives can increase the reliability of the temporal segmentation. However, in order to further increase the detection efficiency, we propose to perform the analysis within the scale space of derivatives.

More precisely, let $v^{'}(n)$ denote the first order derivative of vector $v(n)$, defined as the following finite difference:

$$v^{'}(n) = v(n) - v(n-1) \tag{2.71}$$

We construct the set of cumulative sums $\{v_k^{'}(n)\}_{k=1}^{N}$, over the difference signal $v(n)$ up to order *N*, by setting:

$$v_k^{'}(n) = \sum_{p=0}^{k-1} v^{'}\,(n-p) \tag{2.72}$$

The signals $v_k(n)$ can be interpreted as low-pass filtered versions of the derivative signal $v_k^{'}(n)$, with increasingly larger kernels, and constitute our scale space analysis.

If we process the above equation for all the frames inside a moving window we can determine the derivative within each point of the local minimum vector:

$$\begin{gathered} v'_{k-0}(n) = v(n) - v(n-1) \\ v'_{k-1}(n) = v(n-1) - v(n-2) \\ v'_{k-2}(n) = v(n-2) - v(n-3) \\ \dots \\ v'_{k-n+1}(n) = v(n-(k+1)) - v(n-(k+2)) \\ v'_{k-n}(n) = v(n-(k)) - v(n-(k+1)) \end{gathered} \tag{2.73}$$

After summing up all of the above equations, the cumulative sum $v'_k(n)$ can be simply expressed as:

$$\sum_{p=0}^{k-1} v'(n-p) = v(n) - v(n-k) \tag{2.74}$$

Figure 2.23 illustrates the set of derivatives signals obtained. We can observe that smoother and smoother signals are produced, which can be helpful for eliminating variations related to camera/large object motions.The peaks which are persistent through several scales correspond to large variations in terms of video content and can be exploited to detect the transitions [Ţapu10].

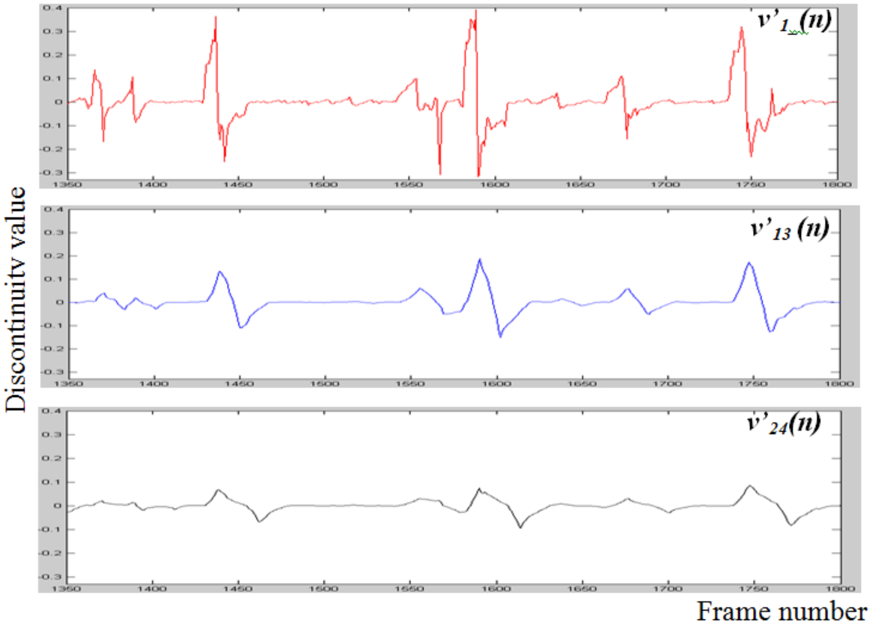


Figure 2.23. The set of scale space derivatives.

In order to detect such peaks, a non-linear filtering is first applied to the multi-scale representation. More precisely, the following filtered signal is constructed:

$$d(n) = \max_k \left\{ \left| v'_k(n) \right| \cdot h(k) \right\} = \max_k \left\{ \left| v(n) - v(n-k) \right| \cdot h(k) \right\} \tag{2.75}$$

where the weights *h(k)* are defined as:

$$h(k) = \begin{cases} e^{-k}, & k \in \left[0, \frac{N-1}{2}\right] \\ e^{N-1-k}, & k \in \left[\frac{N+1}{2}, N\right] \end{cases} \tag{2.76}$$

The shot detection process is applied on the $d(n)$ signal thus obtained. The weighting mechanism adopted privileges derivative signals located at the extremities of the scale space analysis. In this way, solely peaks that are persistent through all the considered scales are retained.

The proposed process also helps to eliminate variations related to large object/camera motions, while preserving the peaks corresponding to the true transitions [Ţapu10], as illustrated in Figures 2.24 and 2.25.

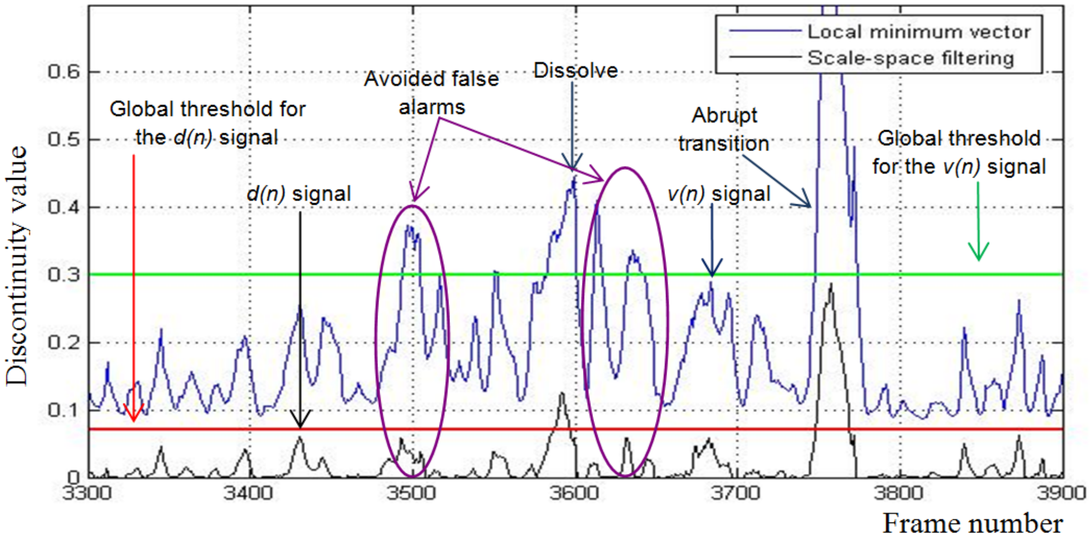


Figure 2.24. A false alarm due to camera motion is avoided when using the scale-space filtering approach.

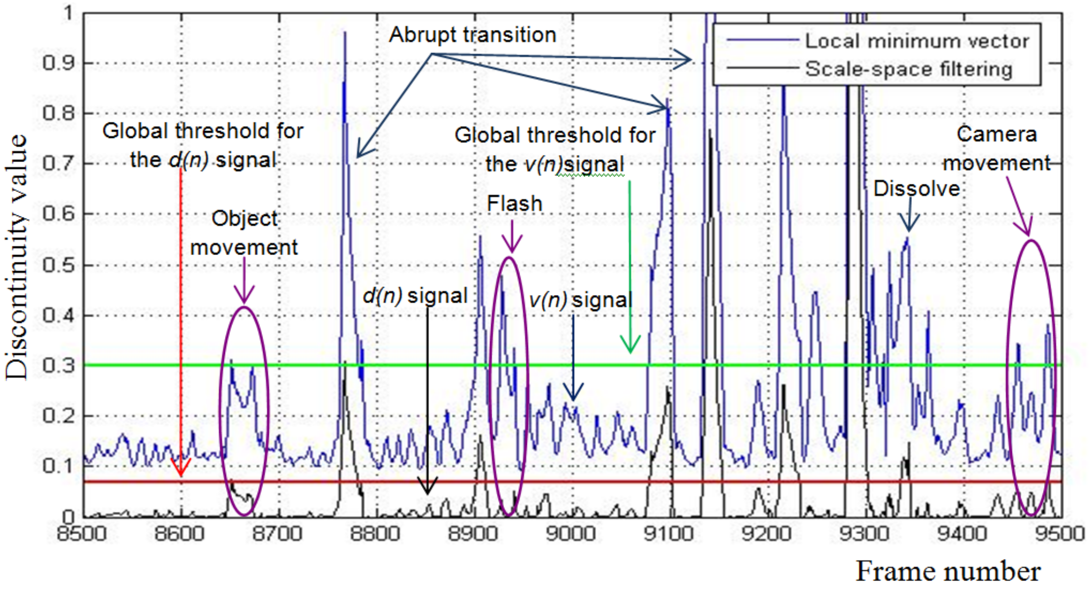


Figure 2.25. False alarms due to large object motion are avoided when using the scale-space filtering approach.

Let us note that a progressive transition can be considered as a transition whose effects have been spread on multiple frames. This behavior makes it possible to detect both types of transitions (abrupt and gradual) in a relatively similar way.

Here, false alarms appear when thresholding the initial local minimum vector $v(n)$. When considering the scale-space filtered signal *d(n)*, this false alarms are avoided since the related variations are considerably reduced.

In the second phase of our video structuring and segmentation framework, we focused on the computational complexity aspects involved.

### 2.3.3. Two-pass approach

The most time consuming stage in the shot detection process concerns the construction of the local minimum vector $v(n)$, which involves the computation of the optimal cut (*cf.* Equation (2.64)).

Determining the similarity matrix requires a large amount of resources (because with each step a number of *N-2* partitions are computed).

In order to reduce the computational complexity, we have considered a new technique based on a two-step analysis process. The principle consists of identifying so-called “certain” transitions in a first stage, while applying the graph partition method uniquely on uncertain time intervals (Figure 2.26).

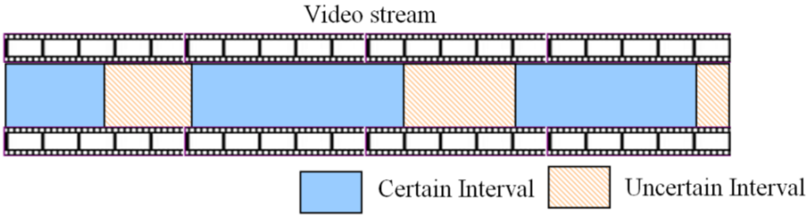


Figure 2.26. Classification of video in certain/uncertain segments.

*Step1*: In a first stage, the algorithm detects video segments that can be reliable considered as belonging to the same shot. Here, a simple (and fast) chi-square comparison of HSV color histograms associated to each pair of consecutive frames is performed, instead of applying the graph partition model. In the same time, abrupt transitions are here detected.

$$D(I_t, I_{t-1}) = \sum_k \frac{(H_k^t - H_k^{t-1})^2}{H_k^t + H_k^{t-1}} \tag{2.77}$$

Concerning the threshold used in the first stage we have considered a value of $T_{g_1} = 0.9$ for detecting a subset of transitions, considered as certain. The selected value is high enough to avoid the introduction of false positives. Also in this stage we have considered a second threshold, set to $T_{g_2} = 0.35$, in order to determine uncertain time intervals. If the dissimilarity values between a number of successive frames are above the second threshold $\left(D(I_t, I_{t-1}) > T_{g_1}\right)$, and also inferior to $T_{g_1}$a more detailed analysis is required and the method passes to the second step. All frames with chi-square distances between HSV histograms of successive images lower than $T_{g_2}$ are considered to belong to the same shots.

*Step 2*: In the second stage, we consider the scale space filtering method described in Sections 2.3.1 and 2.3.2, but applied uniquely to the remaining uncertain video segments. The complex shot boundary detection methods receives as input only specific fragments of the original movie for which the first detector cannot distinguish between camera/object motion, abrupt changes in the light intensity… This second step makes it possible to identify the transition type (whether is abrupt or gradual) and has almost constant detection rates not being influenced by the movie genre, production stile and age conditions, as demonstrated in the following section.

### 2.3.4. Experimental evaluation of the shot boundary detection system

In order to evaluate the proposed algorithm, we have considered a sub-set of videos from the "TRECVID 2001 and 2002 campaigns, which are freely available on Internet (www.archive.org and www.open-video.org). The video corpus includes 8 documents (NASA 25th Anniversary Show: Segment 05, Segment 07 and Segment 08 from TRECVID 2001 and Wrestling with Uncertainty, Exotic Terrane, Adelante Cubanos (Part I), Desert Venture (Part I), The Egg and US) totalizing 122.36 minutes and 1183 shots.

Some videos in the considered corpus are illustrated in Figure 2.27.

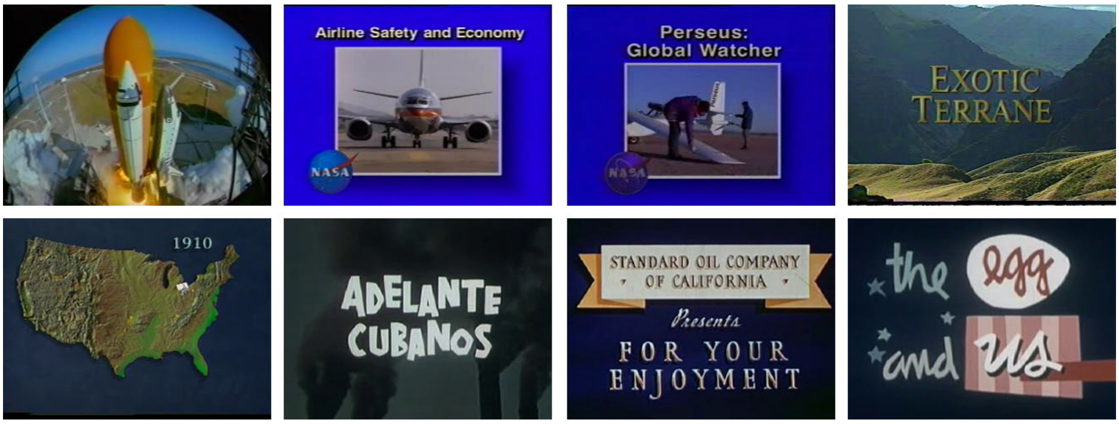


Figure 2.27. Video corpus.

The videos are mostly documentaries that vary in style and date of production, while including various types of both camera and object motion. For all the films in the database, a ground truth has been established by manually segmenting the video data. Their corresponding characteristics are summarized in Table 2.1.

As evaluation metrics, we have considered the traditional Recall (*R*), Precision (*P*) and F1 norm (*F1*) measures, defined in Section 2.2.4.

We have compared the proposed approach with the method of [Yuan07], which is considered as state of the art and which is also based on a graph partition approach. Let us underline that the Yuan's method also exploits a trained classifier that uses the support vector machine (SVM) to detect transitions. In our case, no learning process is achieved.

Table 2.1. Evaluation corpus features.

| Video title | Number of frames | Number of transition | Abrupt transition | Gradual transition | | | File name |
|---|---|---|---|---|---|---|---|
| | | | | Fade in / out | Dissolve | Other type | |
| NAD 55 | 26104 | 185 | 107 | 21 | 57 | - | NASA Anniversary |
| NAD 57 | 10006 | 73 | 45 | 6 | 22 | - | NASA Anniversary |
| NAD 58 | 13678 | 85 | 40 | 7 | 38 | - | NASA Anniversary |
| UGS09 | 23918 | 213 | 44 | 25 | 144 | - | Wrestling Uncertainty |
| UGS01 | 32072 | 180 | 86 | 6 | 88 | - | Exotic Terrane |
| 23585a | 14797 | 153 | 80 | 2 | 71 | - | Adelante Cubanos |
| 10558a | 19981 | 141 | 79 | 20 | 42 | 1 | Desert Venture |
| 06011 | 23918 | 153 | 81 | 26 | 46 | - | The Egg & US |
| **TOTAL** | **164474** | **1183** | **562** | **113** | **508** | **1** | |

Table 2.2 summarizes the results obtained for shot boundary detection when considering the Yuan *et al.* approach, while Table 2.3 presents the detection performances obtained for the proposed scale-space filtering approach.

The various parameters involved are the following $N$ = 25 frames and threshold $T_g$ = 0.7 for all videos.

*Note:* In the case of abrupt transitions detection we considered a tolerance of 10 frames (in both directions) from the actual position of a cut, while for gradual transitions the deviations from the actual location was set at 25 frames.

The results presented in Table 2.2 and 2.3 can be further processed in order to compute the precision, recall and *F1* norm rates. These measures allow us to make a complete evaluation of the proposed method against other techniques, existent in the technical literature. The obtained scores after applying both methods are given in Table 2.4 and 2.5.

Table 2.2. Yuan *et al*. algorithm’s performance evaluation.

| Video title | Number of transitions | Abrupt transitions | | | | Gradual Transitions | | | | Total FA |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Total | Detected | MD | FA | Total | Detected | MD | FA | |
| NAD 55 | 185 | 107 | 103 | 4 | 22 | 78 | 68 | 10 | 24 | 46 |
| NAD 57 | 72 | 45 | 39 | 6 | 6 | 28 | 22 | 6 | 5 | 11 |
| NAD 58 | 85 | 40 | 38 | 2 | 7 | 45 | 35 | 10 | 18 | 25 |
| UGS09 | 203 | 44 | 43 | 1 | 8 | 169 | 137 | 32 | 40 | 48 |
| UGS01 | 180 | 86 | 78 | 8 | 14 | 94 | 79 | 15 | 26 | 40 |
| 23585a | 153 | 80 | 60 | 20 | 5 | 73 | 58 | 15 | 2 | 7 |
| 10558a | 141 | 79 | 68 | 11 | 10 | 62 | 48 | 14 | 4 | 14 |
| 06011 | 153 | 81 | 74 | 7 | 8 | 72 | 60 | 12 | 23 | 31 |
| **TOTAL** | **1173** | **562** | **503** | **59** | **80** | **621** | **507** | **116** | **142** | **222** |

Table 2.3. The novel graph partition based on scale space derivative algorithm performance.

| **Video title** | **Number of transitions** | **Abrupt transitions** | | | | **Gradual Transitions** | | | | **Total FA** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Total** | **Detected** | **MD** | **FA** | **Total** | **Detected** | **MD** | **FA** | |
| NAD 55 | 185 | 107 | 107 | 0 | 9 | 78 | 73 | 5 | 18 | 27 |
| NAD 57 | 72 | 45 | 43 | 2 | 2 | 28 | 24 | 4 | 3 | 5 |
| NAD 58 | 85 | 40 | 38 | 2 | 4 | 45 | 43 | 2 | 10 | 14 |
| UGS09 | 203 | 44 | 44 | 0 | 4 | 169 | 159 | 10 | 14 | 18 |
| UGS01 | 180 | 86 | 84 | 2 | 11 | 94 | 93 | 1 | 15 | 26 |
| 23585a | 153 | 80 | 75 | 5 | 4 | 73 | 67 | 6 | 1 | 5 |
| 10558a | 141 | 79 | 76 | 3 | 9 | 62 | 60 | 2 | 10 | 19 |
| 06011 | 153 | 81 | 76 | 5 | 6 | 72 | 68 | 4 | 10 | 16 |
| **TOTAL** | **1173** | **562** | **543** | **19** | **49** | **621** | **587** | **34** | **81** | **130** |

Table 2.4. Recall, precision and F1 norm for Yuan *et al*. algorithm.

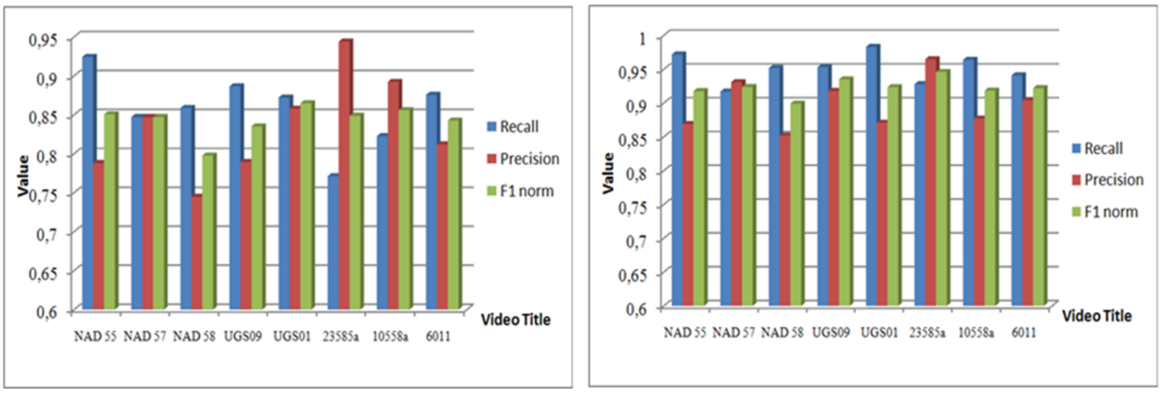
| **Video title** | **Abrupt transitions** | | | **Gradual Transitions** | | | **All transitions** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **R** | **P** | **F1** | **R** | **P** | **F1** | **R** | **P** | **F1** |
| NAD 55 | 0.9626 | 0.824 | 0.8879 | 0.8717 | 0.7391 | 0.7999 | 0.9243 | 0.788 | 0.8507 |
| NAD 57 | 0.8666 | 0.8666 | 0.8666 | 0.7857 | 0.8148 | 0.7999 | 0.8472 | 0.8472 | 0.8472 |
| NAD 58 | 0.95 | 0.8444 | 0.8940 | 0.7777 | 0.6603 | 0.7142 | 0.8588 | 0.7448 | 0.7977 |
| UGS09 | 0.9727 | 0.8431 | 0.9032 | 0.8106 | 0.7740 | 0.7918 | 0.8866 | 0.7894 | 0.8351 |
| UGS01 | 0.9069 | 0.8472 | 0.8760 | 0.8404 | 0.7523 | 0.7939 | 0.8722 | 0.8579 | 0.8649 |
| 23585a | 0.75 | 0.923 | 0.8275 | 0.7945 | 0.9666 | 0.8721 | 0.7712 | 0.944 | 0.8488 |
| 10558a | 0.8607 | 0.8717 | 0.8661 | 0.7741 | 0.923 | 0.8420 | 0.8226 | 0.8923 | 0.8560 |
| 06011 | 0.9135 | 0.9024 | 0.9079 | 0.8333 | 0.7228 | 0.7741 | 0.8756 | 0.8121 | 0.8426 |
| **TOTAL** | **0.8950** | **0.8627** | **0.8785** | **0.8164** | **0.7812** | **0.7984** | **0.8610** | **0.8198** | **0.8398** |

Table 2.5. Recall, precision and F1 norm for the proposed algorithm.

| **Video title** | **Abrupt transitions** | | | **Gradual Transitions** | | | **All transitions** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **R** | **P** | **F1** | **R** | **P** | **F1** | **R** | **P** | **F1** |
| NAD 55 | 1 | 0.922 | 0.9594 | 0.935 | 0.802 | 0.8634 | 0.972 | 0.869 | 0.9176 |
| NAD 57 | 0.955 | 0.955 | 0.955 | 0.857 | 0.888 | 0.8722 | 0.917 | 0.931 | 0.9239 |
| NAD 58 | 0.95 | 0.904 | 0.9264 | 0.955 | 0.811 | 0.8771 | 0.952 | 0.852 | 0.8992 |
| UGS09 | 1 | 0.916 | 0.9561 | 0.941 | 0.919 | 0.9298 | 0.953 | 0.918 | 0.9351 |
| UGS01 | 0.976 | 0.884 | 0.9277 | 0.989 | 0.861 | 0.9205 | 0.983 | 0.871 | 0.9236 |
| 23585a | 0.937 | 0.949 | 0.9429 | 0.917 | 0.985 | 0.9497 | 0.928 | 0.965 | 0.9461 |
| 10558a | 0.962 | 0.894 | 0.9267 | 0.967 | 0.857 | 0.9086 | 0.964 | 0.877 | 0.9184 |
| 06011 | 0.938 | 0.926 | 0.9319 | 0.944 | 0.871 | 0.9060 | 0.941 | 0.904 | 0.92212 |
| **TOTAL** | **0.9661** | **0.917** | **0.9409** | **0.945** | **0.877** | **0.9097** | **0.955** | **0.896** | **0.9245** |

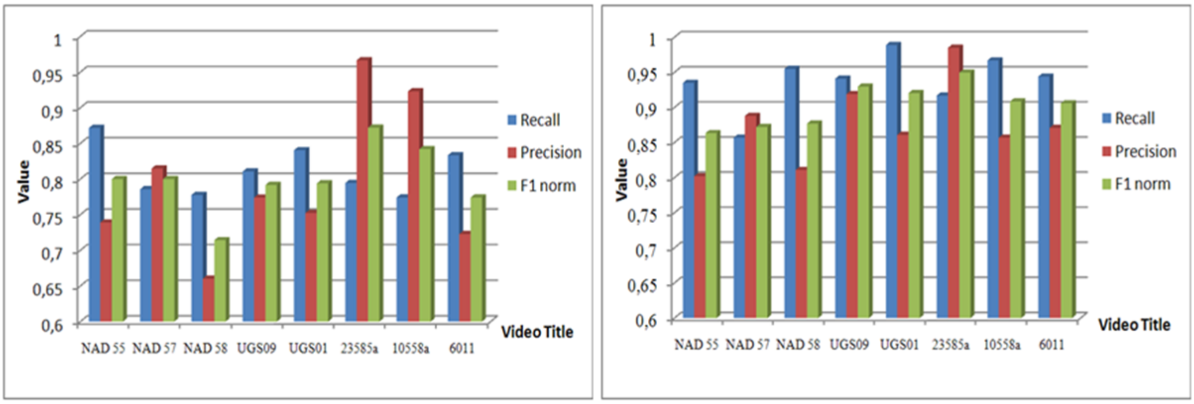
The results clearly demonstrate the superiority of our approach, for both abrupt and gradual transitions, with global gains in terms of recall and precision rates of 9.4% and 7.7%, respectively (Figure 2.28). These gains are even more important in the case of gradual transitions with recall and precision rates of 94,5% and 87,7%, respectively (with respect to *R* = 81.6% and *P* =  78,1% for the reference method [Yuan07]) (Figure 2.29).

Let us also note that such scores are approaching the performances obtained for the detection of abrupt transitions ($R = 96.6\%$ and $P = 91.7\%$), which is quite a remarkable result (Figure 2.30).



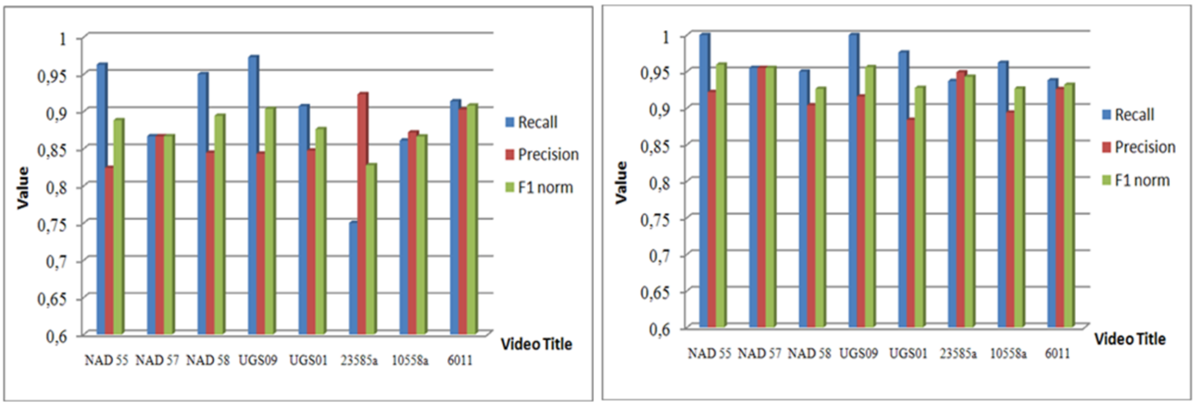
a. b.

Figure 2.28. Recall, Precision and F1 norm rates when detecting all types of transitions for: (a) Yuan *et al*. algorithm, (b) The proposed scale space derivative technique.



a. b.

Figure 2.29. Recall, Precision and F1 norm rates when detecting gradual transitions for: (a) Yuan *et al*. algorithm, (b) The proposed scale space derivative technique.



a. b.

Figure 2.30. Recall, Precision and F1 norm rates when detecting abrupt transitions for: (a) Yuan *et al*. algorithm, (b) The novel scale space derivative technique.

In order to determine the computational complexity and the improvement brought by the proposed two-pass approach, we have also evaluated the processing time required. Table 2.6

and Figure 2.31 present the obtained detection times[1] and its variation in both cases: when applying graph partition for the entire video stream and only for uncertain time intervals.
The results demonstrate the improvement due to our approach with respect to the state of the art algorithm, with savings greater than 25% in computational time. The detection performances are equivalent in both situations, so the two-pass approach does not influence the quality of the shot boundary detection system.

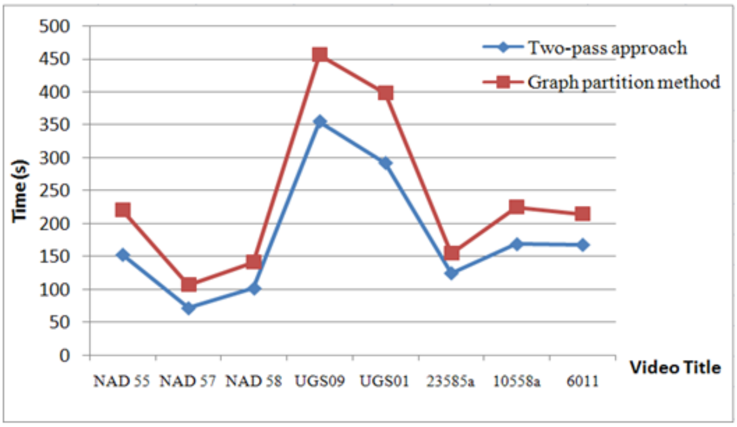


Figure 2.31. Shot boundary detection time when using the two-pass approach.

Table 2.6. Computation time and gain for scale space filtering graph partition method and two-pass approach.

| Video title | Video duration Time (s) | Two-pass approach Time (s) | Graph partition method Time (s) | Gain (%) |
|---|---|---|---|---|
| NAD55 | 871 | 153 | 221 | **30.7** |
| NAD57 | 417 | 72 | 107 | **32.7** |
| NAD58 | 455 | 102 | 141 | **27.5** |
| UGS09 | 1768 | 355 | 457 | **22.3** |
| UGS01 | 1337 | 292 | 399 | **26.8** |
| 23585a | 615 | 125 | 155 | **19.3** |
| 10558a | 833 | 169 | 225 | **25.3** |
| 06011 | 997 | 168 | 215 | **21.8** |
| **TOTAL** | **7293** | **1436** | **1920** | **25.2** |

## 2.4. CONCLUSIONS AND PERSPECTIVES

In this chapter, we have considered the issue of shot boundary detection. First, we have presented an overview of the scientific literature dedicated to the subject, and the main families of methods proposed over the last two decades, including intensity-based approaches, histogram representations, compressed domain techniques, edge/contour-based methods, motion-based approaches and graph-based techniques. The analysis of the state of the art highlighted the following conclusions:

[1] The algorithms were run on a Pentium IV machine with 3.4 GHz and 2 Go RAM, under a Windows XP SP3 platform.

1. Methods based on the absolute difference of pixels colors/intensities are highly sensitive to noise and camera/object motion.

2. Color histogram-based approaches offer an interesting and useful approach, due to histogram invariance to information spatial distribution and insensitivity to low motion.

3. Techniques based on edges/contours have slightly inferior performances compared to histogram-based methods. In addition, they suffer from a higher computational complexity. However, such edge features are highly useful for removing false alarms caused by abrupt illumination changes.

4. Motion-based approaches can take advantage of features available in the MPEG compressed domain (*e.g.*, motion vectors associated to MPEG macro-blocks). However, such motion information is mostly related to motion compensation purposes and thus remains unreliable for shot detection objectives.

Whatever the visual features involved, graph-based representations proved recently their superiority in terms of detection performances when compares to frame-to-frame comparison techniques.

By considering the general graph-based representation and detection framework proposed in [Yuan07], we have proposed a novel technique, based on a scale-space analysis of a dissimilarity vector computed with the help of HSV color histograms. The key stage of our algorithm concerns the non-linear scale space filtering of the derivatives of the similarity vector associated to the graph partition model. Notable, this mechanism makes it possible to enhance the robustness of the detector with respect to camera/large object motion. The filtering stage makes it possible to eliminate the signal variations cased by motions, while preserving the peaks corresponding to the real transitions.

Moreover, we have proposed a two-pass analysis approach, in order to reduce the associated computational complexity. In a first step, the algorithm detects time intervals which can be reliably considered as belonging to the same shots. Abrupt transitions considered as certain are also detected in this stage. In a second step, the analysis is further performed only for uncertain time intervals.

The experimental results, carried out on a data set of publicly available video sequences of various types and including both abrupt and gradual transitions demonstrate the superiority of the proposed approach with respect to the state of the art method of [Yuan07], with average gains in precision and recall rates of 8%, for 25% savings in computational time.

In our future work, we will consider the integration in our approach other visual representations, including edge and motion features. An adaptive thresholding technique can also be considered in this framework in order to further enhance the detection performances. Finally, the object detection/identification issues (*cf.* Chapter 5), can also be taken into account within a more general framework.

# 3. AUTOMATIC VIDEO ABSTRACTION

**Summary:** In this chapter, we consider the issue of video abstraction/summarization which aims at providing a concise representation of a video document, based on a set of representative key-frames. First, we analyze the state of the art, focusing on the main advantages and limitations of each technique. Then, we introduce a novel method to construct static storyboards, which extracts, for each detected shot, a variable number of keyframes depending on the visual content variation. In order to reduce the computational time, the analysis is performed only on a reduce number of frames by taking advantage of the shot boundary detection algorithm presented in Chapter 2, which computes graph partition within a moving window. The leap keyframe extraction method leads to a gain of 23.2% in overall extraction efficiency. An additional post-processing step, based on SIFT interest points is introduced, in order to remove irrelevant images from the selected set of keyframes.

**Keywords:** Video abstraction, static storyboard; visual content variation, keyframe extraction, blank or test card frames removal.

Video abstraction techniques aim at eliminating the inherent redundancy present in video documents, in order to provide a compact, comprehensive and illustrated summary of the considered video sequence. Such a summary consists of a shorter representation of the original sequence and should include the most relevant segments in order to enable a fast browsing and retrieval of the original content.

The keyframe-based representation is highly useful for video indexing and browsing objectives. Thus, the selected representative images offer to the end user guidance to locate specific video segments of interest. In the same time, the keyframes are the most suitable in representing the entire content of an image sequence, so the visual features encountered here can constitute the basis of any video indexing application.

Video abstraction techniques provide concise information, with still or moving images, about the video content while conserving the original message [Pfeiffer96]. Elaborating automatic or semi-automatic methods able to capture in a semantically pertinent manner the video content and requiring a minimal amount of human intervention is still today a challenging issue of research.

Two different types of summaries can be developed in order to characterize image sequences: static video summary and video skimming (Figure 3.1) [Li01].

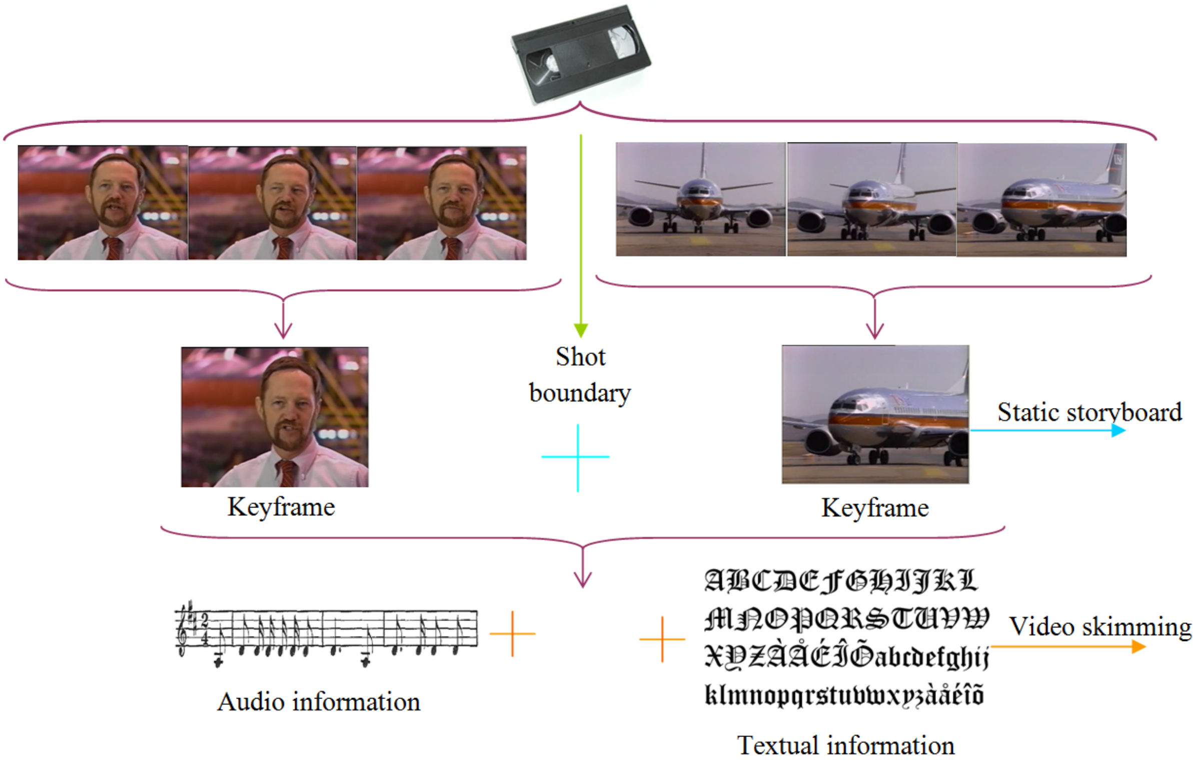


Figure 3.1. Video abstraction: types of summaries.

Static video summaries, also known as still abstracts or storyboards, are defined as a set of representative images (keyframes) selected from the original movie. On the other hand, video skimming, also called moving abstract, is a collection of short video sequences incorporating several audio-visual cues for presenting the user with a condensed and succinct representation of the video content. They can be further categorized into highlights and summary sequences. A movie highlight includes only the most interesting parts of a film while the summary sequence attempts to condense all the semantically meaningful parts in a shorter version with respect to the temporal order.

Whatever the type of video summary considered, two different parameters need to be taken into account: the selection of relevant keyframes/shots, and their corresponding layout. The first parameter is a global characteristic given by the representative power of the selected features, while the second is related to the data layout in both spatial and temporal domains.

In our work, we have considered solely the video summarization techniques, based on static story-boards. The various keyframe detection methods proposed in the state of the are described in the following section.

## 3.1. DEVELOPING VIDEO SUMMARIES

A video summary is defined as a set of salient images (keyframes) selected or reconstructed from an original video sequence. The selection of such salient images from all the frames of an original video is in this case essential for guaranteeing the representative power of the resulted video summary.

In a more formal manner, a static summary ( $St_{summary}$ ) of a video sequence $S$ is defined as:

$$St_{summary}(S) = \{frame_1, frame_2, ..., frame_N\} \tag{3.1}$$

where $frame_i$ represent the $i^{th}$ extracted keyframe, with $i = 1,..., N$ while $N$ gives the total number of representative images included in the storyboard. Parameter *N* directly influences the quality of the resulting storyboard.

An important issue to be solved concerns the value of parameter *N*. A first approach consists of imposing *a priori* the number of considered key-frames. However, it is difficult to imagine that a single value can be adapted and useful for all types of videos encountered in practice. So, ideally, video summarization techniques should be able to adaptively determine appropriate values of parameter *N*.

Video summarization techniques based on a keyframe extraction principle can be divided into four categories [Li01]: sampling-based, shot-based, and segment-based approaches. They are described in the following sub-sections.

### 3.1.1. Sampling – based key-frame extraction

The simplest method for keyframe generation consists of uniformly sampling the video sequences, [Taniguchi95], [Mills92]. Such an approach offers the advantage of simplicity and the associated computational complexity is almost negligible. However, their major drawback is related to the reduced representative power of such a representation which does not take into account the video stream structure and dynamics.

In this case, short yet important segments may not have any representative frames, while longer segments, with redundant informational content, can represented by multiple keyframes.

A second type of approach notably aims to capture the video structure and is related to the shot structure of the video document.

### 3.1.2. Shot-based keyframe extraction

One of the first attempts to automate the keyframe extraction process is to consider first a shot detection technique. Then, a keyframe is selected and defined as the first, middle, random or last frame appearing in each detected shot (Figure 3.2) [Shahraray95].

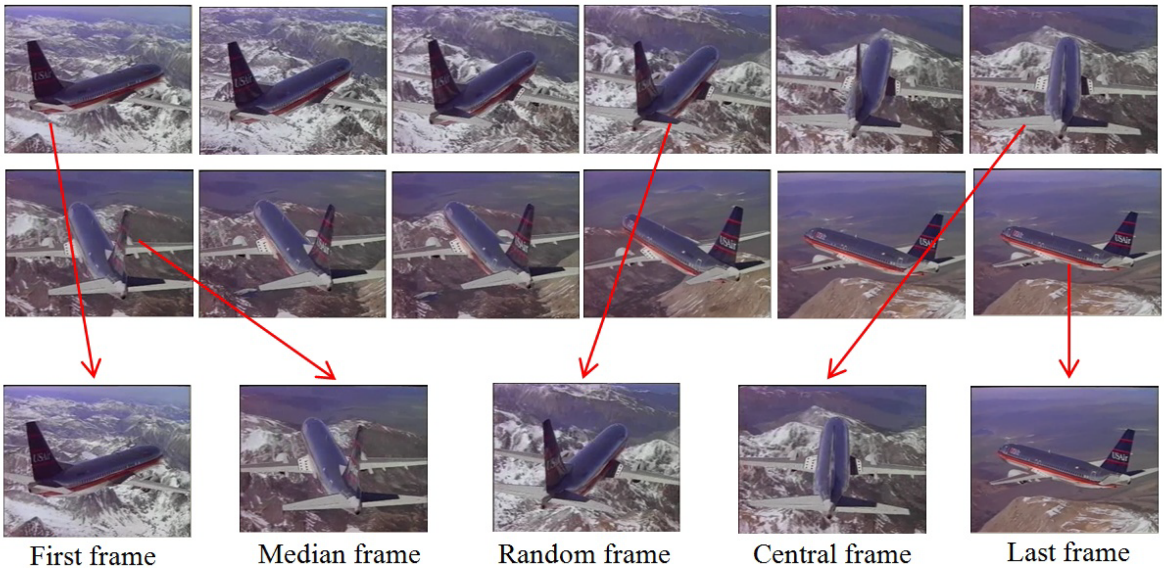


Figure 3.2. Example of a static summary for a shot with large object motion.

Such a strategy provides satisfactory results for stationary shots, where the video content variation is relatively low. However, a single frame does not provide an acceptable representation for highly dynamic shots. In such cases, it is necessary to elaborate methods that are able to extract significant information based on the visual content variation along the video flow [Hanjalic99].

In [Hanjalic99], different keyframe extraction strategies are compared. An interesting observation is here highlighted: the strategy of selecting keyframes without any information about the content provides satisfactory results for a quasi-constant shot, while for more complex content the frame situated in the middle of the considered shot is more pertinent to offer a significant preview of the global action. Such an observation is consistent with the definition of a shot (*cf*. Chapter 2). The direct implication is that in the case of highly dynamic shots, where the visual homogeneity principle is violated (in the sense of associated visual characteristics), more advanced keyframe extraction techniques should be considered.

This shows the necessity of adapting to the underlying semantic content, maintaining as much as possible the dynamic nature of the video content while removing all redundant information. In theory, semantic primitives of videos, such as interesting objects, actions and events should be used. However, because a general semantic analysis is a highly difficult task, notably when no information from soundtracks and/or close caption is available, the sole alternative is to rely on low-level image features, such as: color, motion, hybrid…. Such approaches are described in the following sections.

#### *3.1.2.1. Color based selection*

Methods based on color information select representative keyframes considering the statistic analysis of pixel intensity or the color histogram variation over a shot.

One of the first methods proposed in the literature is described in [Yeung95]. Here, each shot is represented by an image set that can effectively capture temporal variations caused by camera operations or object motions. The approach starts from the following hypothesis: for image sequences with little or no significant variation in time, one representative frame (*e.g.* the first image) is sufficient to capture all the informational content. For long shots or shots with important variations, a variable number of images is considered.

Selecting a keyframe set that captures all information from a shot is achieved by a nonlinear temporal sampling, which measures the dissimilarity (based on the luminance projections) between the last frame selected as important and all the other remaining frames. A new image is marked as representative if it presents a considerable visual variation.

A major limitation of the technique is related to the high probability that the selected frame belongs to a transitional effect. Clustering algorithms (Figure 3.3) offer a natural solution to solve such problems.

One of the first methods proposing an unsupervised clustering based approach to select representative frames is introduced in [Zhuang98]. The $N$ frames of a shot determined after a temporal segmentation algorithm are clustered into $M$ classes using an agglomerative clustering technique. For each couple of frames the similarity of their visual content is measured with the help of color histograms, computed in the HSV color space. A cluster density measure (and representing the minimum number of objects within a given radius) is also computed.

All clustering algorithms have a weakness related to the threshold parameter (denoted by τ) which controls the cluster density. For a higher value of parameter $\tau$, the number of elements included in a same class can become extremely large.

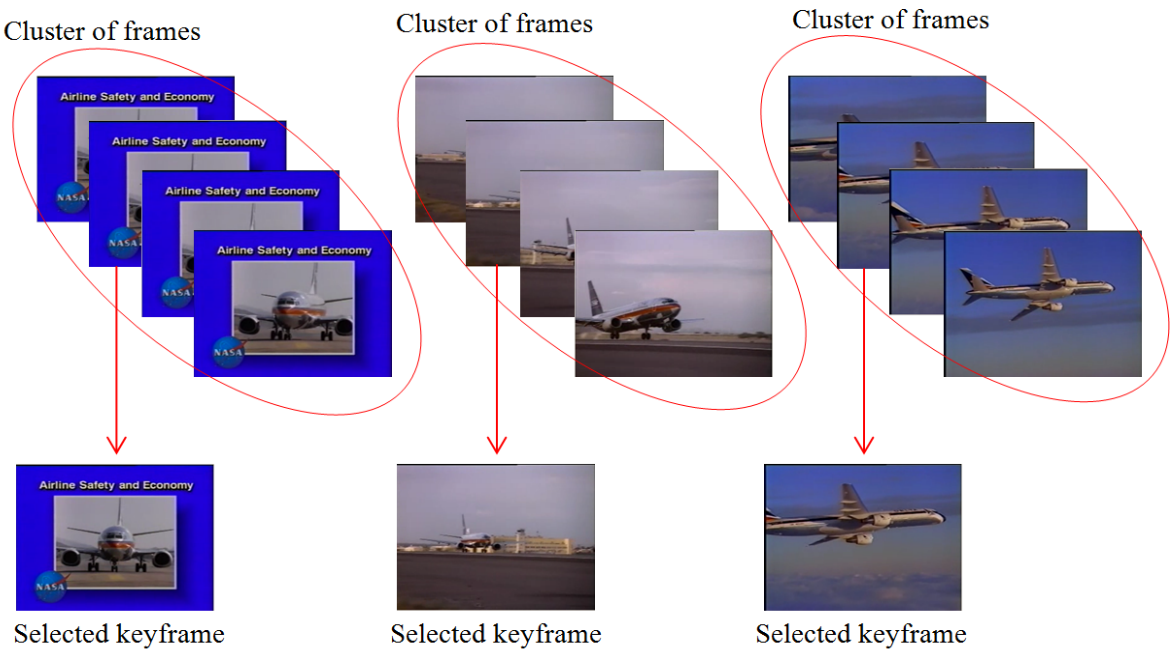


Figure 3.3. Keyframe selection based on clustering techniques.

The method presented in [Gunsel98] integrates the shot boundary detection and keyframe selection tasks in one single step by using a 2D feature vector, expressing the color variation in the YUV color space. The YUV color representation is here adopted due to its consistency and applicability in both compressed and uncompressed domain.

The first component of the feature vector is obtained with the help of a color histogram comparison and is determined for each pair of successive frames *t* and *t+1*, as described in the following equation:

$$f_i^1 = \sum_{j=0}^{N-1}\left(|H_i^Y(j)-H_{i-1}^Y(j)|+|H_i^U(j)-H_{i-1}^U(j)|+|H_i^V(j)-H_{i-1}^V(j)|\right) \qquad (3.2)$$

where $H_i^Y(j)$, $H_i^U(j)$, $H_i^U(j)$ represent the color histogram for Y, U and V components and $N$ is the number of histogram bins.

The second component of the feature vector is computed as the difference between the current frame histogram and the mean histogram of all previous ones in the current shot:

$$f_i^2 = \sum_{j=0}^{N-1}\left(|H_i^Y(j)-H_{mean}^Y(j)|+|H_i^U(j)-H_{mean}^U(j)|+|H_i^V(j)-H_{mean}^V(j)|\right) \qquad (3.3)$$

Finally, two thresholds are used, a first one in order to determine the shot boundaries, based on Equation (3.2), and the second one dedicated to the keyframe selection process (Equation (3.3)). The multi-threshold technique makes it possible to overcome the problems related to gradual shot transitions.

An enhanced method of selecting representative frames with an improved computational efficiency is described in [Girgensohn99]. The approach is based on the observation that clustering methods are computational expensive. In order to reduce the complexity, the algorithm starts by considering only keyframes that are very dissimilar to each other (with respect to a visual criterion). To reduce the influence of noise caused by any type of video motion a chi-square ( $\chi^2$ ) histogram difference is considered.

The histogram representation is performed in the YUV color space with 8 bins for luminance and 4 bins for each chrominance, for a total of 128 bins. The problem of selecting an optimum threshold to establish the cluster density is here avoided by adopting a complete link method, which consists of setting the maximum difference between two frames in different clusters as threshold of the hierarchical agglomerative clustering technique considered.

Recent trends in keyframe extraction methods consider spatio-temporal color distributions in order to determine representative images. The approach described in [Sun08] starts by constructing a so-called temporally maximum occurrence frame (TMOF) by observing the distribution of the pixel value in the same position through frame pairs within a shot. First, the method computes the color histogram $H_{i,j}(b)$ with pixel values from the same position through the frames in the shot. Second, a Gaussian filtering operation is applied on the histogram. Finally, for each filtered signal authors search for a peak value that will be considered as reference.

From the distribution of the pixel value in the same position throughout the frames, a content descriptive frame is constructed, considered as the reference frame for the shot. In the TMOF frame each pixel position stores the representative color information that lasts for the longest duration. By comparing the distance between each frame in the shot with TMOF, they obtain the distribution of the features that describes the shot content. The keyframes correspond to the peaks in the distance sequence.

Existing algorithms exploit the property of histogram invariance to relatively low camera/object motion. However, in the case of large amplitude camera/object motion, this property does not hold anymore. In addition, the various thresholding parameters involved need to be manually adjusted.

In order to overcome such limitations, a second family of methods exploits motion analysis of the considered video.

#### *3.1.2.2. Motion-based approaches*

One of the first methods encountered in the technical literature is presented in [Wolf96]. For each frame, the global motion amplitude is associated to, by summing the amplitude of all motion vectors determined with the help of an optical flow algorithm.

The local minima of the global motion amplitude signal over time are selected as keyframes. This process is illustrated in Figure 3.4.

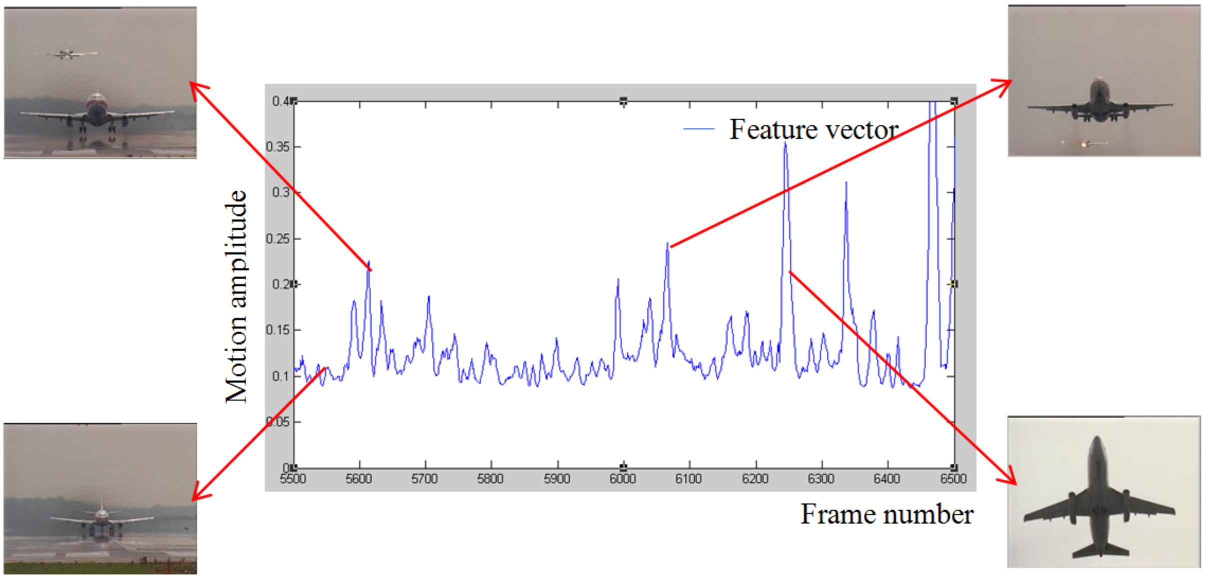


Figure 3.4. Motion amplitude variation over time.

A similar principle is adopted in [Dirfaux00], where shot boundary detection and keyframe extraction are simultaneously performed. Here again, the selected keyframes correspond to frames with low motion activity, in order to avoid blurring or excessive coding artifacts.

An algorithm that attributes semantic meaningful representations of shots is described in [Luo09]. The number of representative images selected from a video depends on various factors including camera motion, scene content, image quality and interaction between objects. A psycho-visual experiment is proposed, in order to determine common criteria used by humans when selecting keyframes.

In [Liu03], motion patterns are used to characterize the content variation in videos. For each rapid modification of the informational content, a specific motion pattern is created. The motion characterization is performed with the help of a so-called perceived motion energy (PME), which takes into account the percentage of dominant motion direction and the motion magnitude. Such a parameter can be determined with the help of the motion vectors included in MPEG streams. By considering an acceleration/deceleration model, The PME makes it possible to segment shots into sub-shots. For each sub-shot, a keyframe, corresponding to the maximum/minimum instance in the acceleration / deceleration process is associated to.

The approach described in [Guironnet07] introduces a keyframe extraction method dedicated to video shots presenting continuous and homogeneous camera motion. Authors suggest that for dynamic shots the selection of the first, last and median images, is sufficient to completely describe the analyzed segments. For relatively static shots, they claim that the first and the last frame are sufficient to completely capture all the informational content. The camera motion magnitude is also considered in this framework.

In [Fauvet04], a set of geometric criteria are applied for selecting representative keyframes. The analysis starts with a temporal segmentation of the input video flow. Next, all frames within a shot are projected onto the same coordinate system by using the 2D affine motion parameters. This makes it possible to determine the amount of content corresponding to entering, shared, and lost areas (Figure 3.5).

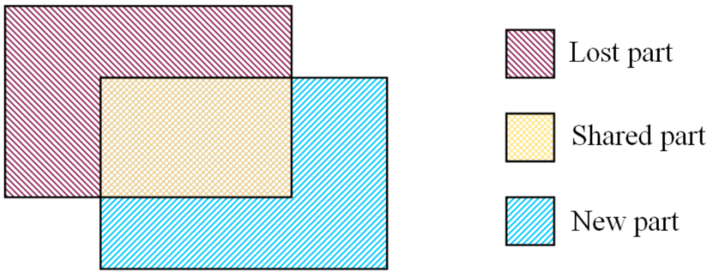


Figure 3.5. Three scene information parts.

The number of selected keyframes is here set manually by the user.

The motion-based approaches are useful to extract representative images for shot with important variation of the visual content. However, such techniques suffer from various limitations, in particular in the case of gradual transitions or large object motions.

#### *3.1.2.3. Hybrid approaches*

Hybrid techniques combine various sources of information such as color, texture and motion, in order to obtain a reliable keyframe selector.

Within this context, let us first mention the non-sequential video content representation algorithm proposed in [Doulamis00]. Here, a preliminary temporal segmentation is performed, in the compressed domain. The informational content of each frame is described with the help of global color and motion histograms. The keyframes are determined with the help of a multi-resolution approach based on Recursive Shortest Spanning Tree (RSST). Afterwards, a motion activity descriptor is considered, in order to characterize both camera and object movement. The motion activity is determined within each detected shot, based on a macro-block analysis. For keyframe extraction, a cluster-based method exploiting both color and motion features and relying on agglomerative clustering technique is proposed.

In [Zhang99], a robust keyframe extraction technique is proposed. The approach is based on the information accumulated during a detection process. The algorithm starts by choosing the first frames located near a shot boundary as a representative one. Successive frames are then selected according to the following criteria:

- Color based criteria - after the first keyframe is selected, the following frames in the shot are sequentially compared against the last detected keyframe, based on their similarities defined by color histogram or moments. If a significant content change occurs between the current frame and the last detected keyframe, the current frame is selected as a new keyframe.

- Motion based criteria – in order to select keyframes representing camera panning and zooming motion, the authors introduced two additional criteria. For a zooming sequence, at least two frames will be selected: the first and last frame since one will represent a global, while the other will represent a more focused view. For a panning-like sequence, the number of frames to be selected will depends on the scale of panning.

Authors point out the importance of considering the motion information in the keyframe extraction process. Thus, in the case of panning sequences, each keyframe should capture different parts of activity, in order to reveal the spatial context of the whole action with little overlap.

A combined histogram and interest point-based keyframe extraction method is proposed in [Fu09]. A shot boundary detection method, based on color histogram comparison, is first considered. Histograms are here constructed in the RGB color space, with histogram intersection as dissimilarity measure. The interest points are extracted using the Harris point detector. The first frame of each video shot is selected as a benchmark, reference frame.

In the second step, all the shot frames are compared with the reference keyframe one by one. If the dissimilarity between the first frame and the current frame *k*, given by a histogram intersection and the total number of matched keypoints, is greater than a pre-established threshold, the current frame is selected as keyframe and included in the representative image set. For all the following frames, the dissimilarity is computed based on the comparison of the current image with the most recent benchmark frame. The algorithm ensures a good representation of the video content with a variable number of keyframes for each analyzed sequence. However, the choice of the threshold is made on an empirical basis.

Moreover, when selecting the first frame encountered after a shot boundary as a keyframe, as presented in [Zhang99] and [Fu09] a possible limitation is related to the possibility that such a frame belongs to a transitional effect, which makes it poorly representative. So, selecting the first frame is not an optimal solution.

A universal content representation scheme for video information management is presented in [Ferman98]. The technique develops a quantitative description of each detected shot by using statistical features such as mean color and color histograms.

The summarization scheme proposed in [Stefanidis00] selects keyframes based on object trajectory tracing and identification. In order to select representative images the object trajectory line is segmented with brake points called nodes. The nodes are distributed dynamically to capture the information content of regions in the 3D spatio-temporal (*x*,*y*,*t*). More nodes are assigned where trajectory presents breakpoints (*e.g.*, moving at fixed velocity) and fewer nodes are assigned to segments where the spatiotemporal behavior of an object is smooth. The number of nodes is dynamically distributed along the trajectory, using elements from self organizing maps (SOM) or clustering approaches (K-means), in order to capture the variation of the informational content.

The keyframe extraction method described in [Kelm09] is focused on selecting representative images from amateur movies stored on video websites. Unprofessional films are quite hard to deal with because of their unstructured character and low quality. The shot boundaries are first determined with the help of HSV color histogram comparisons. Each shot is further divided into sub-shots in order to detect the content variation introduced by camera or object motion. A motion intensity vector is here defined, with the help of the motion vectors included in the MPEG compressed stream. The local minima of the motion intensity signals are identified and the corresponding frames are considered as keyframes. In addition, a face (using Haar classifier) and a text detector are used in order to increase the representative power of the keyframe representation.

A complex algorithm to extract keyframes from already segmented videos is proposed in [Ciocca05]. Here, multiple visual descriptors (color histogram, edge histogram and wavelet statistics) are exploited in order to capture the changes and complexity of the visual content. The keyframe selection algorithm is applied on a curve of cumulative frame differences computed based on feature variation over the entire shot. Sharp slopes indicate complex shots with consistent motion or dynamic events. The selected keyframes correspond to the mid points between each pair of consecutive curvature peaks.

One limitation of such approaches is related to the number of representative frames obtained, which may be considerably high in the case of long video documents (*e.g.*, more than an hour). In order to deal with such an issue, various methods exploit a higher structural level (called segment) for the analysis. The segment can be identified as a scene, but also the entire input video stream can be considered for abstracting.

### 3.1.3. Keyframe extraction on segment level

One of the first approaches in this area of is introduced in [Fauvet04]. The video is first divided into video segments of fixed length $L$. The resulting segments are classified into two classes, so-called small-change and the large-change clusters, corresponding to the motion activity present in the considered segment. For the small change clusters the first and the last frame are selected as representative frames, while for the large-change clusters all the frames are retained. If the number of the resulted frames is not the desired one, the procedure dynamically regroups (using a K-means clustering technique) the set of retained images, and the procedure is re-iterated. The algorithm converges when the desired number of keyframes is obtained.

In [Sun00], authors aim at generating a static storyboard that optimally covers all the video content, with the help of a Singular Value Decomposition (SVD) analysis. The input video is first sampled at a rate of 3 frames per seconds (fps). For each selected frame, a color histogram in the RGB color space is computed. In order to incorporate spatial information each frame is further dived in 3 x 3 non-overlapping blocks to which a color histogram is associated with. The extracted features are used in order to construct a similarity matrix *A* to which the SVD is applied.

The SVD makes it possible to determine both static and dynamic video segments. The considered frames are grouped using a clustering algorithm based on similar features, while the cluster centroid is selected as keyframe.

The method assures a minimum informational redundancy for the summary, as well as the possibility for the user to specify the number of frames contained in the still abstract. One potential problem is given by the lack of consistency regarding the temporal order of frames, which is lost in the grouping process.

A multi-resolution keyframe extraction method is proposed in [Yu04]. Here, the KPCA (Kernel-based Principal Component Analysis) method is used in order to identify representative images. The KPCA is performed on a color histogram extracted from the DC image constructed in the compressed domain using the MPEG video stream. Finally, all frames are clustered with the help of a fuzzy c-means clustering technique that selects a predetermined number of classes. The cluster centroid represents the selected keyframe.

Methods that extract keyframes from segments or even from the entire video sequence attempt to solve the problems encountered by shot based approaches by eliminating the temporal segmentation step and by selecting a reduced number of frames to characterize the visual content.

## 3.2. THE PROPOSED STATIC STORYBOARD TECHNIQUE

In this section, a novel technique to develop static storyboards, with reduced computational complexity, and based on keyframe representation is proposed.

### 3.2.1. Leap extraction algorithm

For each detected shot, a variable number of keyframes is determined. The number depends on the visual content variation. The first keyframe extracted is always situated near a shot boundary. We use here the shot boundary detection method introduced in Section 2.3. By definition, this frame is located at *N* (*i.e.*, the window size used for the shot boundary detection - *cf*. Section 2.3.1) frames away after a detected transition, in order to make sure that the selected image does not belong to a gradual effect.

However, for dynamic shots with relatively important amount of motion, only one frame is not sufficient to adequately represent the content of a video shot. In this case, multiple images need to be selected, based on the visual variation, for a finer shot characterization (Figure 3.6).

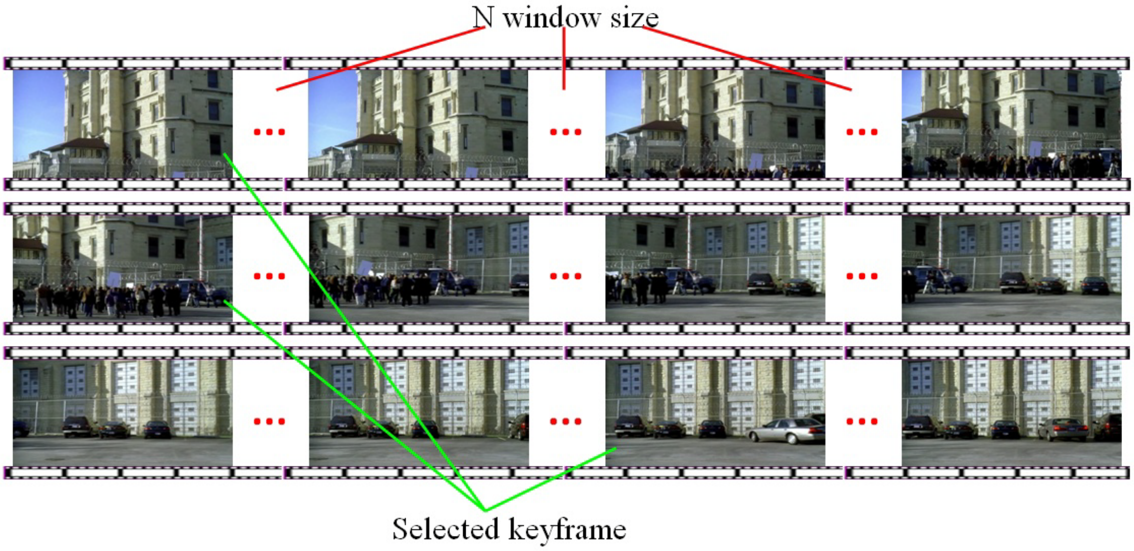


Figure 3.6. Keyframe selection in a dynamic shot with camera movement (pan left).

In order to characterize dynamic shots that present various types of camera motion or large object displacement, we introduce a leap-extraction method that consider for analysis only the images located at integer multipliers of window size (Figure 3.7).

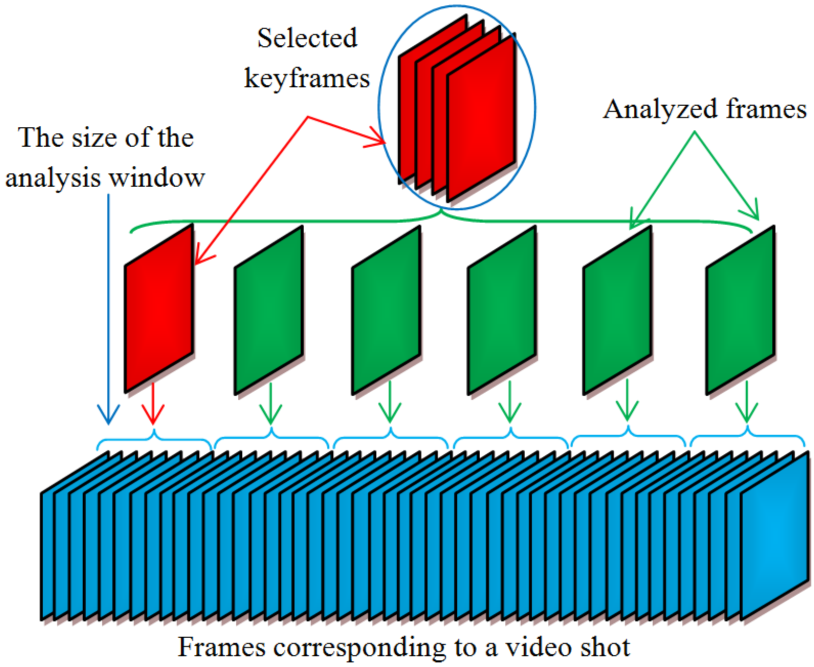


Figure 3.7. Keyframe extraction based on the proposed technique.

After selecting the first keyframe the algorithm starts computing the dissimilarity between the image being analyzed and the selected keyframe.

In our case, we define the dissimilarity as a normalized cross correlation distance between color histograms represented in the HSV color space. The frames are normalized in order to avoid the variation of lighting and exposure conditions.

$$D(I_t, I_{t-1}) = \frac{\sum_{k=1}^{N} H_k^{'i} \cdot H_k^{'j}}{\sqrt{\sum_{k=1}^{N} H_k^{'i\,2} \cdot H_k^{'j\,2}}} \tag{3.4}$$

where $H'^{i}_{k}$ and $H'^{j}_{k}$ are the normalized color histograms for the *i* and *j* frames, computed as:

$$H_k^{'i} = H_k^i - \left(\frac{1}{N}\right) \cdot \left(\sum_{j=1}^{N} H_k^j\right) \tag{3.5}$$

and *N* is the total number of bins in the histogram.

If the distance $D(I_t, I_{t-1})$ is superior to a given threshold $T_g$ , the analyzed image is selected as keyframe and included in the shot storyboard. The process is repeated recursively with the observation that all the following frames are further compared with the entire set of keyframes already extracted. Based on the amount of visual content variation (expressed as normalized cross correlation distance in the HSV color space) a new frame is selected as representative image if its visual content differ significant (above a fixed threshold) from all other frames previously extracted.

Let us note that the analysis is performed only upon a reduced number of frames, by taking advantage of the shot boundary detection algorithm. By computing the graph partition within a sliding window, the method ensures that all the relevant information will be taken into account. Let us also note that the number of detected keyframes set per shot is not fixed *a priori*, but automatically adapted to the content of each shot.

### 3.2.2. Post-processing: useless frame detection and removal

We consider an additional post-processing step, firstly described in [Chasanis08] that eliminates all the monochrome and color bar images from the selected set of keyframes assuring that the story board captures all informational content of the original movie without any irrelevant images, which influence directly the representative power of the summary.

The static summary obtained may contain in its structure some useless frames such as monochrome images or color bars (Figure 3.8.a and 3.8.b).

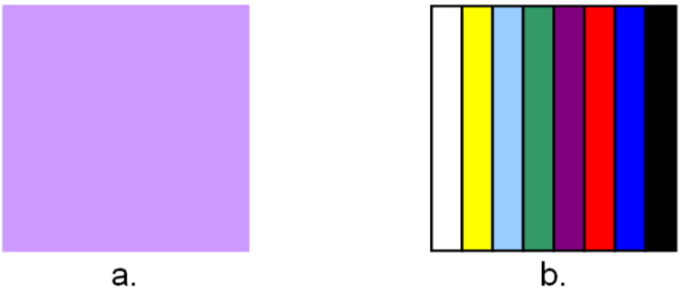


Figure 3.8. Useless keyframes: (a) monochrome; (b) color bars.

The method starts computing an edge histogram descriptor. The keyframe is divided in 4 x 4 rectangular sub-images and the detected edges are classified into five categories: vertical, horizontal, $45^0$ diagonal, $135^0$ diagonal and non-directional edges. A total number of $5 \times 16 = 80$ histograms need to be computed in order to make correct classification of an edge orientation in a whole frame.

In Figure 3.9 we present the edge histogram for a color bar (a) and for a normal frame (b). As it can be observed, a color bar frame has some distinctive features: the histogram bins only for horizontal and vertical direction have non-zero values and all the other bins corresponding to diagonal and non-directional edges are null.

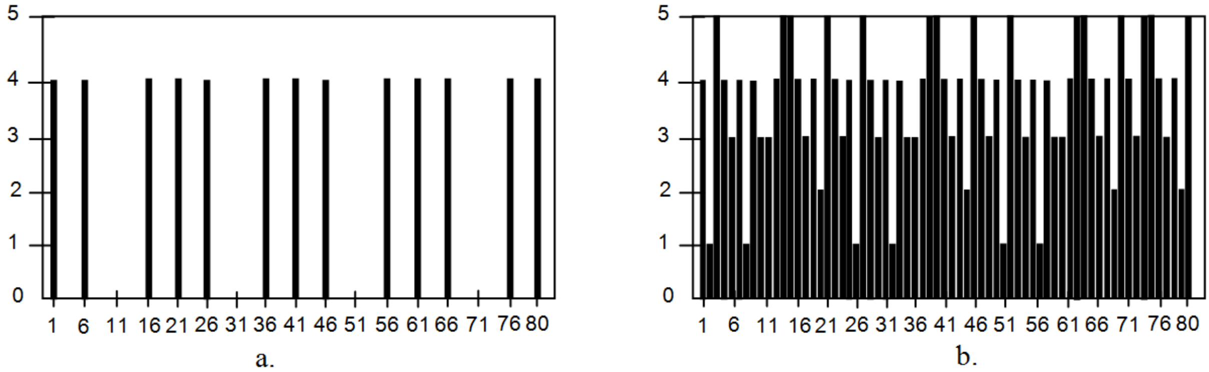


Figure 3.9. Edge direction for: (a) color bar; (b) normal frame.

In this case it is straightforward to distinguish between an important keyframe and a useless image because a color bar or a monochrome frame return small differences (lower than a threshold) between the sum of all bins in the edge histogram descriptor and the sum corresponding only to the vertical and horizontal bins.

## 3.3. EXPERIMENTAL EVALUATION

In our experiments, we have considered for evaluation a corpus of 12 videos (both sitcoms and Hollywood movies) commonly used in the technical literature [Rasheed05], [Zhang99]. The following extra videos in the database have been considered: Seinfeld (SF), Two and a half men (TM), Prison Break (PB), Ally McBeal (AM), Sex and the city (SC), Friends (FR) - sitcoms , The 5th element (5E), Ace-Ventura – The Pet Detective (AV), Lethal Weapon 4 (LW), Terminator 2 – The Judgment Day (T2), The Mask (MK) and Home Alone 2 (HA2) – Hollywood movies.

Table 3.1 presents the computational time necessary to extract representative frames for each detected shot, in both cases: when applying our leap-extraction strategy for selecting keyframes and the classical method of selection [Rasheed05], [Zhang99] based on direct comparison of all adjacent frames inside a shot. The detected representative frames for each shot are highly similar in both cases.

Table 3.1. Computation time and gain for classic and leap keyframe extraction strategy.

| | Video title | Video duration Time (s) | Leap-Extraction Method Time (s) | Classical Extraction Time (s) | Gain (%) |
|---|---|---|---|---|---|
| Sitcoms | SF | 1313 | 297 | 434 | **31.5** |
| | TM | 1200 | 344 | 509 | **33.4** |
| | PB | 2558 | 990 | 1260 | **21.5** |
| | AM | 2607 | 1209 | 1642 | **26.4** |
| | SC | 1779 | 824 | 1067 | **22.7** |
| | FR | 1506 | 309 | 371 | **17.1** |
| Hollywood movies | 5E | 7277 | 3548 | 4581 | **22.5** |
| | AV | 5353 | 1982 | 2774 | **28.5** |
| | LW | 7333 | 3015 | 3985 | **24.3** |
| | T2 | 8812 | 3589 | 4154 | **13.6** |
| | MK | 5820 | 2173 | 2987 | **27.2** |
| | HA2 | 7200 | 3212 | 4215 | **23.8** |
| **TOTAL** | | **52758** | **21492** | **27979** | **23.2** |

The results presented in above table demonstrate that the proposed approach makes it possible to significantly reduce the computational complexity for static storyboard development, for equivalent performances regarding the selected informational content. The leap keyframe extraction method leads to a gain of 23.2% in extraction efficiency (Figure 3.10).

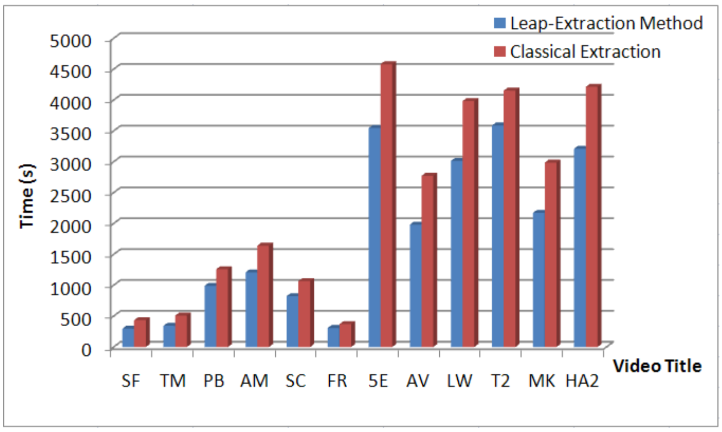


Figure 3.10. Static storyboard construction time.

For each detected shot identified based on the technique presented in Section 2.3 we applied the leap extraction method. In Figure 3.11 and 3.12 we illustrated the set of representative images chosen for complex shots with important visual content variation (large object displacement and camera movement).

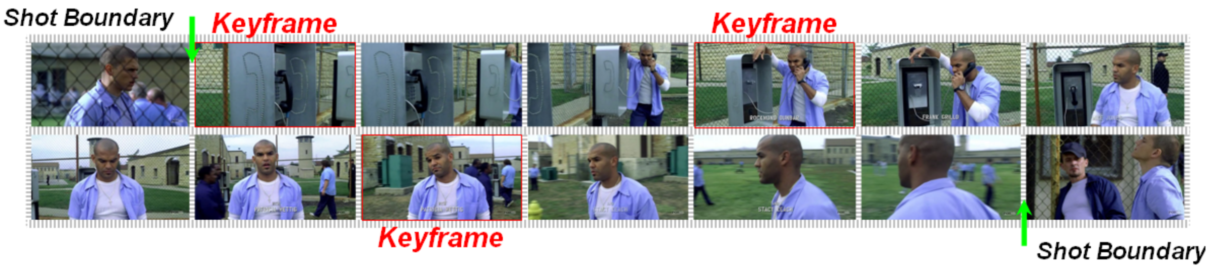


Figure 3.11. Experimental results – shot boundary detection and keyframe.

In this case a large number of keyframes is required in order to capture all informational content. As it can be observed, the first segment is completely described by 3 images while the second one needs 6 keyframe.

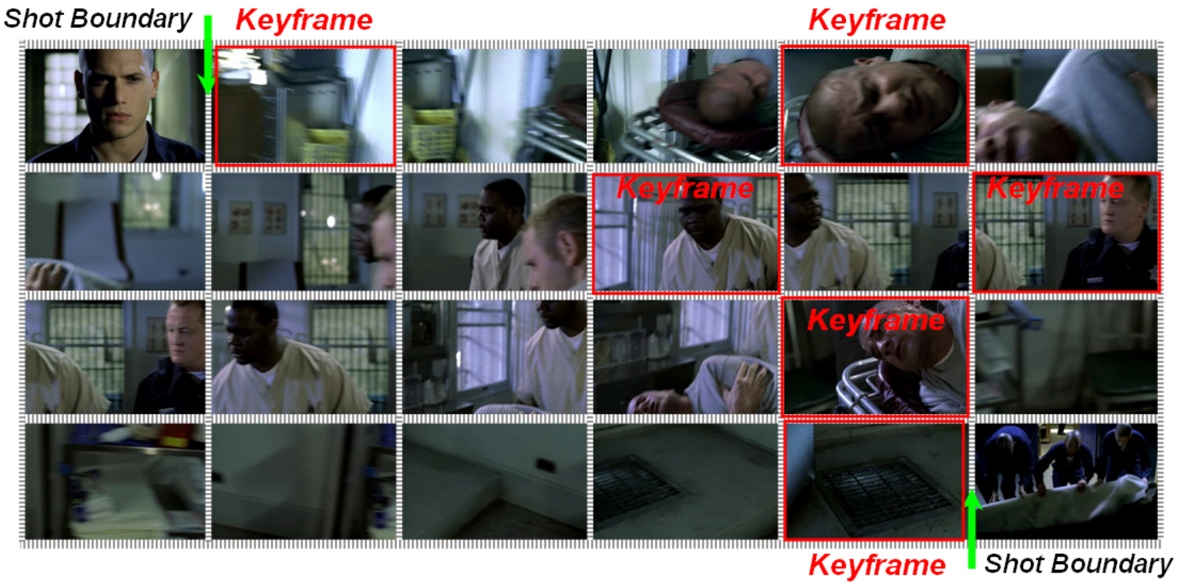


Figure 3.12. Experimental results – shot boundary detection and keyframe selection for complex shots characterized by important visual content variation and abrupt changes in the light intensity.

## 3.4. CONCLUSIONS AND PERSPECTIVES

In this chapter, we have considered the issue of video summarization with static storyboards. The objective is to determine a set of salient images, so-called keyframes that are considered representative with respect to the informational content of a given video sequence.

A review of state of the art summarization techniques has been first proposed. The rich literature dedicated to this research issue illustrates a great variety of approaches involving various visual features, including color, motion and hybrid techniques.

Inspired from the method introduced in [Rasheed05], we proposed a fast static storyboard technique based on leap keyframe extraction algorithm. The method exploits the shot boundary technique proposed in Chapter 2 and is able to adaptively identify the required number of keyframes, based on a measure of content variation. In order to reduce the computation time, the analysis is performed only on a reduced number of frames while assuring that all the relevant information is kept. The leap keyframe extraction method leads to a gain of 23.2% in overall extraction efficiency.

# 4. HIGH LEVEL TEMPORAL VIDEO SEGMENTATION

**Summary:** In this chapter we tackle the issue of video scene/DVD chapter segmentation. First, we propose a review and analysis of the most salient methods existent in the technical literature. A scene can be interpreted as a group of video shots that are correlated according to the semantic interpretation of the content and needs to respect three continuity rules related to space, time and action. However, in some circumstances such constraints may not hold from a purely visual point of view, as in the case of scenes with large camera/object motion.

In the second part of the chapter, we introduce a novel methodological framework for high level video temporal structuring and segmentation that extracts scenes/DVD chapters based on temporal constraints clustering, adaptive temporal lengths, neutralized shots and adaptive thresholding mechanism. The output of our method provides a structured video and facilitates the user access to different parts of the image sequence. In order to validate the proposed technique we have considered two types of low level visual features (*i.e.* the HSV color histogram and the interest points extracted using the SIFT descriptor). The experimental evaluation validates the proposed approach returning an average score for the *F1* norm of 86%.

**Keywords:** Scene segmentation, DVD-chapter extraction, temporal clustering, neutralized shots, adaptive thresholds.

A scene is defined in the Webster's dictionary as follows: "a subdivision of an act or a play in which the setting is fixed, the time is continuous and the action is developed in one place". This traditional definition of a scene can be further interpreted as a group of video shots that satisfy a certain homogeneity with respect to a *semantic* criterion. By definition, a scene needs to respect three continuity rules, corresponding to unitary space, time and action [Lowe04].

However, from a purely visual analysis, in some circumstances such constraints may not hold, as in the case of scenes with large camera/object motion. Elaborating methods for pertinent and automatic scene identification is still today an open issue of research.

When creating video summaries, the necessity of this phase is put into evidence by the increased number of shots that might appear in commercial movies (*e.g.* in "Terminator 2 – The Judgment Day" in 15 minutes of video there were identified 314 shots). In such cases, a static summary becomes extensively large and almost meaningless to any user. Moreover, a human watches a film by its semantic scenes and not by its structural elements as shots with the corresponding keyframes. In order to cover meaningful semantic parts of a movie the extracted shots need to be organized into scenes. So, in the area of video indexing applications, the process of scene identification/construction becomes a fundamental step.

In real-life videos there are various circumstances when multiple events are simultaneously developed and sequentially revealed to the user. As a typical example, let us mention the case of a dialogue scene where two people are talking to each other. Even though both persons are involved in the same discussion, the video shots alternate, and the camera switches back and forth between them (Figure 4.1).

In this case, the various shots involved do not have any continuity in space, which translates into different visual appearances. However, from a semantic point of view, the constituent shots are related and should be grouped into the same scene.

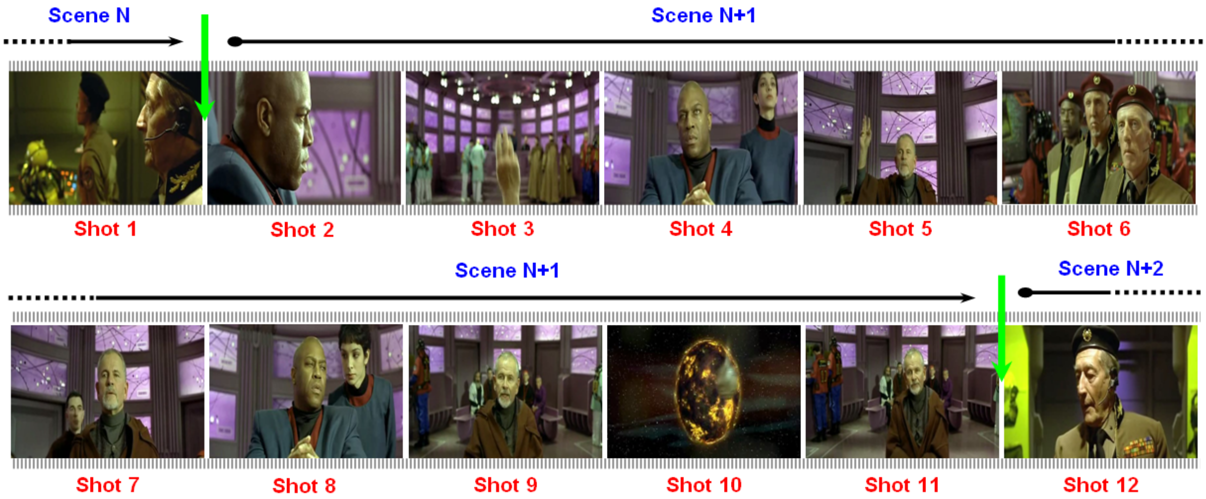


Figure 4.1. Switch back and forth between successive shots belonging to same scene.

Let us first review the main families of methods proposed in the literature.

## 4.1. RELATED WORK

As representative of scene-based identification techniques, let us first mention the graph-based segmentation approaches.

### 4.1.1. Graph-based segmentation methods

In a general manner, such methods use, as fundamental unit of a given video sequence, the shot. Scenes are then obtained by grouping together sets of shots, under both visual and temporal constraints. The graph representation can offer a compact structure of a video based on the temporal relation between their constituent shots.

Within this context, let us first mention the approach proposed in [Yeung98]. Here, the dissimilarity between two different shots in the video is defined as:

$$d(S_i, S_j) = \min_{b_i \leq l \leq e_i,\, b_j \leq l \leq e_j} D(f_l, f_k) \tag{4.1}$$

where $D(.,.)$ is the measure of dissimilarity between two images (frames), $S_i$ and $S_j$ are two video shots determined as their corresponding sets of frames: $S_i = \{f_k\}_{k=b_i}^{e_i}$ and $S_j = \{f_k\}_{k=b_j}^{e_j}$. Equation (4.1) states that the two shots are considered as similar if they contain at least one pair of similar frames $f_m, f_n$ with $b_i \leq m \leq e_i;\ b_j \leq m \leq e_j$.

Let us note that a temporal sub-sampling operation is also performed, in order to reduce the total number of frames associated with each shot.

A scene transition graph (STG) is a oriented graph with $G = (V, E, F)$, where $V = \{v_i\}$ is the set of nodes, $E$ is the set of edges and $F$ is a mapping that partitions the set of shots $\{S_i\}$ into $v_1, v_2, \dots, \in V$ such that the nodes are sufficient similar according to some dissimilarity metric (Equation (4.1)). In this context, each shot is considered as a node in the graph structure. A directed edge is created between two nodes ($U$ and $W$) if there is a shot represented by node $U$ that immediately precedes any shot represented by $W$. The relation between nodes is governed by the temporal ordering of the shots and the dissimilarity between the associated keyframes.

Based on the graph concept a scene is defined as set of related shots in a particular setting. In the graph structure, each interaction is shown by the presence of an edge connecting the nodes containing the respective shots.

In order to prevent grouping far apart, but also similar shots, a temporal constraint condition is imposed. Thus, within a scene transitional graph, each shot can be described as a node and the temporal relationship ($d_t(S_i, S_j)$) between two shots can be represented by the edges weights. The temporal distance between $S_i$ and $S_j$ is determined based on Equation (4.2):

$$d_t(S_i, S_j) = \begin{cases} \min(|b_j - e_i|, |b_i - e_j|), & i \neq j \\ 0, & i = j \end{cases} \tag{4.2}$$

The graph structure can be extracted automatically using the visual content and some temporal information, without any specific knowledge of the video content and structure. The nodes are clusters of visually similar shots and the edges indicate the temporal flow of the story. The scene segmentation is performed also by taking into consideration the temporal distance between shots. This means that for any two shots that are far apart in time, even if they share similar visual contents, they potentially represent different contents or occur at different scenes. Time-constrained clustering further imposes a time-window parameter $T$ that prevents two shots that are far apart in time but similar to be clustered together.

In order to detect the scene boundaries, the resulted graph is split into sub-graphs using the complete link method of hierarchical clustering [Jain88]. In this case, a sub-graph can be considered as a scene if the intersection between color histograms is above a fixed threshold.

In [Rasheed05], the detection problem is formulated as an optimal partition of a graph. The nodes of the graph represent the various shots of the video. The weight of an edge ($W(i,j)$) is defined as proportional with the product between shot similarity and temporal proximity:

$$W(i,j) = w(i,j) \times ShotSim(i,j) \tag{4.3}$$

where:

- $w(i,j) = w(|i-j|)$ is a decreasing function of the temporal distance between shots, defined as:

$$W(i,j) = \exp\left(-\frac{1}{d}\left|\frac{m_i - m_j}{\sigma}\right|^2\right) \tag{4.4}$$

where $m_i$ and $m_j$ are the middle frames from shots $i$ and $j$ respectively, $\sigma$ is the standard deviation of a shot duration (with respect to the considered video) and $d$ is a factor controlling the temporal decrease;

- $ShotSim(i,j)$ represents a global similarity between shots defined as:

$$ShotSim(i,j) = \alpha \cdot VisSim(i,j) + \beta \cdot MotSim(i,j) \tag{4.5}$$

where $\alpha$ and $\beta$ are positive, real-valued weights respectively associated with the so-called visual similarity ($VisSim$) and motion similarity ($MotSim$).

The visual similarity is defined between any pair of arbitrary shots $i$ and $j$ as the maximum $ColSim(x,y)$ of all possible pairs of their key-frames.

$$VisSim(i,j) = \max_{p \in K_i, q \in K_j} \left(ColSim(p,q)\right) \tag{4.6}$$

where $K_i$ and $K_j$ denote the set of keyframes belonging to shot $i$ and $j$ respectively, while $ColSim$ is defined as the color similarity between two images:

$$ColSim(x,y) = \sum_{h \in bins} \left(H_x(h), H_y(h)\right) \tag{4.7}$$

where $H_x$ and $H_y$ are the HSV color histogram of frames $x$ and $y$ respectively and $ColSim(x,y) \in [0,1]$.

To each shot *i*, authors propose to associate a measure of the motion content ($Mot_i$).

Based on the motion content the motion similarity is defined as described in Equation (4.8).

$$MotSim(i;j) = \frac{2 \times \min(Mot_i, Mot_j)}{Mot_i + Mot_j} \tag{4.8}$$

If two shots have similar motion content their *MotSim* will have a high value.

Finally, the weighted unidirectional graph is partitioned with the help of the normalized graph cut measure (*cf.* Section 2.3) [Yuan07].

In [Rasheed05], Rasheed *et al.* simply choose the decrease control factor *d* as a constant value.

In [Zhao07], authors affirm that *d* is related to the number of shots *N*. When the shot number *N* is large/small, *d* should correspondently increase/decrease to avoid over-segmentation/under-segmentation. They claim that the square root of parameter *d* should be proportional to the number of shots *N*. The temporal distance becomes then:

$$w(i,j) = \exp\left(-\frac{\left(m_i - m_j\right)^2}{\sqrt{N} \cdot \sigma^2}\right) \tag{4.9}$$

In addition, an extra measure called shot goodness (*G(i)*) that depends on the visual content variation within the shots included in the current scene is defined as:

$$G(i) = \left(\sum\nolimits_{j \in Scene} ShotSim(i,j)\right)^2 \cdot Length(i) \tag{4.10}$$

where $ShotSim(i,j)$is defined as in Equation (4.5).

Here, two different types of scenes, so-called parallel and serial are introduced. A parallel scene includes at least one interacting event (PI) (*e.g.,* dialog between two persons) or two or more serial events developed simultaneously (PS) (*e.g.*, a men is going home on the road while the child is fighting with the thieves at home). A serial scene (SS) include neither interacting events nor serial events happening simultaneously (*e.g.,* a man is driving a car from one city to a mountain).

Figure 4.2 presents the temporal layout patterns of shots from different scenes. Each circle represents a shot and the same letter in different shots indicates similar shots. For parallel scenes with interacting events there are often two fixed cameras capturing two persons and the view points are switched alternative. For parallel scenes with simultaneous serial events the action switches between two actions developed simultaneously, with an impact one over the other. For serial scenes the camera setting keeps changing and the visual similarity for consecutive shots returns high values.

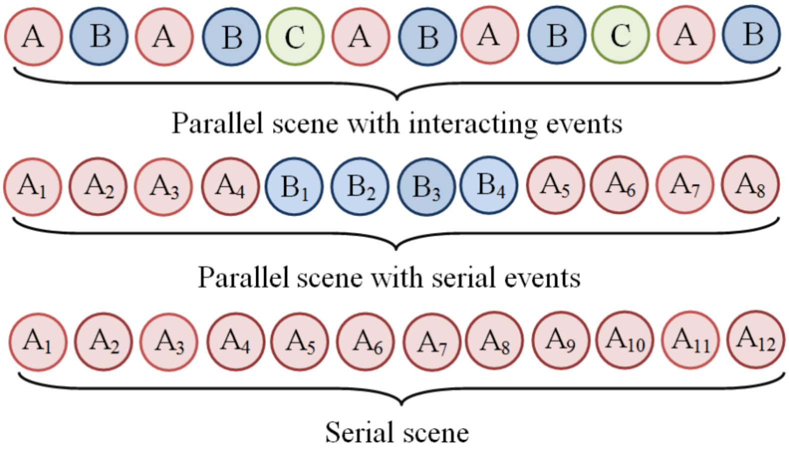

A B A B C A B A B C A B
Parallel scene with interacting events
A1 A2 A3 A4 B1 B2 B3 B4 A5 A6 A7 A8
Parallel scene with serial events
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12
Serial scene

Figure 4.2. Semantic description of a video scene.

A mosaic-based scene detection method is introduced in [Aner02]. Two principles are here used: shot clustering and repetitive event detection.

The proposed analysis is based on identifying the shots with an important camera motion or large object displacement that can offer a large amount of informational content about the segments. Once these shots are determined their corresponding mosaics are chosen as representative mosaics (R-mosaics). Two shots are identified as developed in the same location if their corresponding R-mosaics are highly similar. The visual similarity between two mosaics is determined based on the rubber sheet matching method [Gozalez93] which takes into account the topological distortions among mosaics.

A scene segmentation algorithm based on shot clustering is introduced in [Bouthemy99]. The structure is constructed in order to represent the similarity between frames with the help of a clustering procedure. In this case, the collection of shots is partitioned into nodes and directed edges are drawn between any two adjacent shots. The representation allows the analysis of a video through the graph properties. One important feature is a cut edge defined as an edge of an unidirectional graph that has the property that when it is removed, two disconnected graphs are formed. The graph definition also guarantees that there always exists a path from any given node to any other node. In this way, each connected sub-graph after the removal of the cut edges will represent a story unit.

In order to reduce the computational time a temporal constraint is introduced. So, to compute the *N(N-1)* distances describing the similarity between shots, authors propose to compute the pairwise distances between keyframes located within a given temporal window. Distances are accessed via a priority stack for efficiency reasons. At each iteration, the smaller distance among the *N(N-1)* distances is selected and the two corresponding clusters are merged. All distances are updated through Lance-Williams formula [Ziberna04]. The algorithm stops when a pre-established value of the maximum similarity is reached. The oriented graph is afterwards segmented into strongly connected components, which are assumed to correspond to scene boundaries.

The technique suffers from various limitations, because of the too restrictive definition of the semantic story unit considered. For example, a shot story element that is included in a broadcast transmission will be included in a larger scene due to the recurrent anchor elements existent in the keyframes selected from anterior and predecessor shots.

A second family of scene segmentation approaches exploits a modeling stage based on a so-called logical story unit.

### 4.1.2. Logical story units approaches

The approach introduced in [Truong03] defines a scene as a story unit that has start and end instants corresponding to the arrival and departure of the characters involved. A HLS color model is here employed for representing visual similarities.

In the case of relatively static low-motion shots, a single frame is sufficient to completely describe the informational content of the shot. Such a frame is selected as the first frame encountered after a transition. In the case of dynamic (high-motion) shots, an increased number of frames is required.

Defined as presented above the color changes can be regarded as edges. The authors propose to identify the scenes boundaries by using an edge detector that is based on Deriche's recursive filtering algorithm [Adams00]. In order to increase the robustness to different lighting conditions and shading characteristics (determined by various camera shooting angles and motion), a new metric is proposed that gradually eliminates from comparison the regions with highest dissimilarity value.

The scene segmentation algorithm introduced in [Hanjalic99] is based on the concept of logical story unit (LSU) that relies on the global temporal consistency of the visual content. An LSU includes repetitive visual elements (*e.g.*, settings, characters…) that appear at various instants in the video document. A LSU include a sequence of temporally continuous shots, which are characterized by overlapping links that connect similar visual elements.

The scene segmentation algorithm proposed in [Hanjalic99] is based on the investigation of the visual information and the temporal variations, as well as on the assumption that the visual content within a movie episode is temporally consistent. In general, for videos with strong actions, the technique can provide satisfactory segmentation results, although the LSU boundaries only approximate the actual scene boundaries.

The method proposed in [Rasheed03] considers a two-pass algorithm. In the first stage, a set of *uncertain* scene boundaries are identified based on the so-called backward shot coherence.

A similarity measure between the current shot $i$ and a set of previously analyzed shots (included in a window of $N$ frames) is first defined as:

$$\mathrm{SC}_{\boldsymbol{i}}^{\boldsymbol{j}} = \max_{f^x \in K_i f^y \in K_j,} (D(f^x, f^y)) \tag{4.11}$$

where $\mathrm{SC}_{\boldsymbol{i}}^{\boldsymbol{j}}$ expresses the shot coherence of shot *i* with shot *j*, where shot *i* and *j* respectively include $n$ and $m$ keyframes, $f^x$is the *x*$^{th}$ frame of the video shot *i*, $f^y$the *y*$^{th}$ frame of video shot *j* and $D(f^x, f^y)$ represents the intersection of histograms of frames *x* and *y*, in the HSV color space.

The backward coherence for a shot *i* is computed as the maximum shot coherence within a temporal window of length *N*:

$$BSC_i = \max_{1 \leq k \leq N} (SC_i^{i-k}) \tag{4.12}$$

A scene is defined as a collection of contiguous shots in time taken at the same location and presenting similar visual content. At the beginning of a new logical story unit, the initial shots are not similar with the shots from the previous LSU. Therefore, the BSCs for these shots are very small. As the action progresses, similar shots are developed. Consequently, the BSCs measures computed for these shots attain higher values. This process continues until the start of a new LSU. The beginning of a new scene can be detected by locating a minimum in the plot of BSCs.

The method is able to determine effectively scenes presenting a repetitive structure as well as dialogue scenes. For semantic units with weaker structure the proposed technique generally leads to an over-segmentation. In order to overcome such a limitation, authors propose to incorporate, in addition to the color information, the shot length and the motion content.

Most of the above approaches determine the shot similarity based on their corresponding visual similarity. Furthermore, they consider the temporal distance of shots as an extra feature that is taken into account when computing the similarity between two shots for shot clustering into scenes. Due to the absence of prior knowledge concerning the video content or the average shot duration, the methods fail to correctly determine the scene boundaries since, in this case, the similarity between shots is computed within a sliding window of fixed dimension. Such techniques return in most of the cases an over-segmented video.

A third family of approaches concerns the methods based on statistical analysis/modeling, described in the following section.

### 4.1.3. Statistical models

In [Chaisorn02], authors exploit Hidden Markov Models (HMM) in order to model and detect the scene transitions. A Markov chain includes a finite set of states that are characterized by their associated probability distribution. A scene transition is detected by using the transition probabilities that are determined based on the shot scene/location change, tagged category represented by a tag id ranging from 1 to 11 (*e.g.*, intro, anchor, gathering, still image, live-reporting, speech, sport, text scene, special, weather and finance) and the speaker change.

Figure 4.3 illustrates an ergodic HMM system with 4 hidden states.

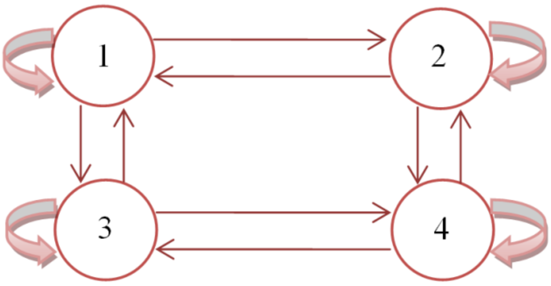


Figure 4.3. Ergodic HMM system.

Let us mention that a HMM is also utilized in [Xie04] in order to distinguish between the two types of moments involved in a soccer game, play and brake.

In order to characterize the video production syntax and the video content, authors propose to exploit the dominant color ratio and the motion intensity. The motion intensity m is computed as the average magnitude of the motion vectors associated to each frame. This measure of motion intensity gives an estimate of the overall motion at the considered instant, including both camera and object motions.

Using the above-cited features, the algorithm can be summarized into two phases. First, the data likelihood of fixed length, relatively short video segments, is evaluated against a pre-trained HMM. In the second step, a correlation measure is developed in order to smooth labels associated to segments and to generate the final segmentation.

In all cases, the HMM is trained with manually labeled, ground truth data with the help of the well-known Expect-Maximization (EM) algorithm. The observations (*i.e.*, the topologies for play and for brake) are modeled as mixtures of Gaussians. Each feature involved is smoothed over time with a low pass filter and normalized with respect to its mean and variance.

The method introduced in [Zhai06] is also based on Markov Monte Carlo Chains (MMCC) in order to identify the central concept within a scene. The central concept refers to shots correlated not only in terms of associated physical environment, but also sharing the same story topic, or the same sub-theme of the story line. Based on such elements, authors propose to use a statistical solution and model the number of scenes and their corresponding boundary locations. A scene boundary is here considered as a change point in the central concept. The estimation of the scene boundaries is determined with the help of MMCC. A hierarchical Bayesian model is here developed, and the optimization problem is solved with the help of the Metropolis-Hasting-Green algorithm [Green95].

In a general manner, such techniques are usually sufficient to cluster together shots characterized by pronounced visual similarity However, the above methods fail to return high precision and recall values in the case of more complex scenes (under 50%).

So far, the approaches presented above are exclusively based on visual features for describing the content of the scenes/shots/frames. A promising axis of research concerns the multi-modal approaches, which combine both audio and visual feature for scene detection purposes.

### 4.1.4. Multi-modal approaches

In [Huang98], three different types of feature are jointly used for video segmentation: audio, color, and motion. For each considered feature, a segmentation process is performed, which leads to three types of distinct breaks.

The audio breaks are detected by sampling the audio signal and dividing it into non-overlapping clips with a fixed length. For each segment, the following set of features is extracted: non-silence ratio, volume standard deviation, dynamic volume range, modulation energy, pitch period deviation, frequency centroid, bandwidth, and energy ratio of three frequential sub-bands. An audio brake is identified based on an audio dissimilarity metric which measures the similarities of the features vector within a window of *N* frames centered on the current frame.

The color brake detection is based on the color frame dissimilarity index, computed based on color histograms in the RGB color space.

Motion information can be determined by computing the motion histograms difference associated to successive frames. However, the accuracy of motion strongly depends on the motion estimation algorithms employed. In order to overcome this limitation, authors propose to use a phase correlation function (PCF) that avoids the explicit compotation of motion vector.

Finally, the scene boundaries are determined based on the common information existent in all the similarity indexes. Using color histogram alone can yield false detection under lighting condition. This false detection can be avoided by examining the PCF, which is invariant to lighting changes.

A scene detection method combining audio and visual information is presented in [Lienhart99]. The technique is based on the assumption that a scene transition will determine a profound change not only on the visual structure but also in the audio track. Two types of audio signals are here defined: background and foreground. The background segments present in their structure a general feeling of the atmosphere encountered in the considered scene but do not carry any relevant information for the action development and understanding. The first step consists of detecting such background audio segments.

In a second phase, the foreground segments are analyzed in order to determine their spectral content. The sound similarity is determined based on a spectral analysis, performed within a Hamming sliding window. An important modification within the audio signal is registered as *audio cut*. An audio shot is then defined as a segment between two successive background - foreground transitions.

Solely for the foreground segments, a set of visual features is then extracted. A color coherence vector (CCV) [Pass96] representation is here adopted. In addition, a global image structure orientation, based on a gradient analysis, is also determined.

The distance between two shots with respect to their color or orientation content is measured based on the disaggregated set representation of the shots, using the minimum distance between the most common feature values [Lienhart98]. The scene extraction algorithm works as follows: a shot cluster includes all shots between two shots which are no further apart than a fixed distance with similarities, in term of color coherence and image orientation, below a fixed threshold. Also, overlapping shot clusters are grouped into the same scene.

The technique also permits to classify shots into various categories, including audio sequences, settings and dialogs.

The method performance depends mainly on the selected set of features and much less on the employed clustering algorithm. Authors also suggest some improvements that could potentially make it possible to distinguish between acts, scene and story units.

Ngo *et al*. [Ngo02] propose to integrate both motion/activity and color cues in order to construct a global criterion for scene detection. The authors propose describing a video as an image volume with *(x,y)* image dimension and *t* temporal dimension. The horizontal slices with dimension *(x,t)* are denoted with **H**, while the vertical slices with a dimension *(y,t)* with **V.** The slices are analyzed for both horizontal and vertical direction in order to discover motion patterns, because they provide rich visual cues along a larger temporal scale.

In order to reduce the computational time and to efficiently use the storage capabilities the authors propose processing and analyzing the slices in the MPEG compressed video domain. The local orientations of slices are estimated based on a so-called structure tensor. The distribution of the local orientation across time inherently reflects the motion trajectory in an image volume.

A concrete way of describing video content for scene change detection is to represent each shot with background images. The authors propose a video representation strategy that consists of two major parts: keyframe selection and background reconstruction. In this case, the shot is represented with the help of both motion and color features (*i.e.*, HSV color histograms).

The proposed technique is able to identify different types of motion (*e.g.*, pan, tilt, zoom…) existent in a video shot and to extract a variable number of keyframes that completely describe its content.

The method introduced in [Ngo02] clusters shots based on the color histogram intersection of keyframes. They use motion information only to exclude moving foreground objects from camera motion and not as a cue for determining shot similarities. However, when sequences of shots form a scene, it is often because the shots are correlated by the same environment rather than by the visual similarity existent in the selected keyframes.

A fifth family of approaches attempts to exploit the semantics associated with the visual content for scene detection purposes.

### 4.1.5. Semantic representation methods

In [Tavanapong04] the authors identify the following three semantic categories of scenes:

- *Travelling scenes* – that consist of one or more characters that travel together to various locations, and spending a brief period in each visited place;
- *Serial event scenes* –characterized by a succession of shots that share the same location;
- *Parallel event scenes* – consists of two or more serial events that are developed simultaneously and reveled to the end user sequentially.

For each detected shot, two keyframes are selected: the first and the last frame of the shot. For each keyframe, a feature vector is associated with. For MPEG videos the features are defined as the average value of all the DC coefficients of the Y color component in the region. For uncompressed videos, the color feature is computed using the average pixels values instead of DC coefficients. A L1 distance between feature vectors is adopted as similarity measure. The scenes boundaries are detected using clustering algorithms constructed in order to take into consideration the semantic scenes categories.

However, the approach can be effectively used only for narrative films. A major limitation of the proposed algorithm is given by its inapplicability to any type of video clip (*e.g.*, news, sport, advertisement) due to the imposed scene categories.

The scene segmentation algorithm proposed in [Li03] is based on temporally overlapping skims that are clustered with the help of the K-means algorithm. The method is based on the assumption that semantically inter-related shots correspond to the same event since different events shall have distinct topics. The technique integrates speech and face information.

Three different categories of scenes are here considered: two-speaker dialogs, multi-speaker dialogs and hybrid scenes. Figure 4.4 illustrates two movie dialogs model. Each node in the figure represents a shot that contains the indicated speaker, the arrows are used to indicated the switches between the two shots.

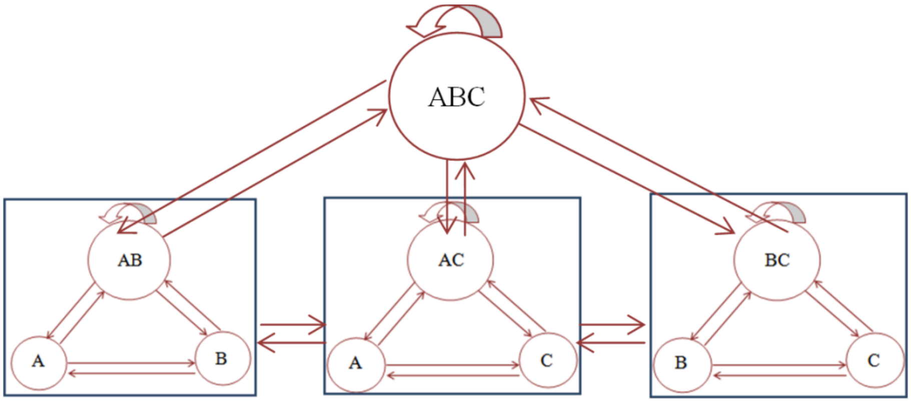


Figure 4.4. Traditional movie dialogs for: 2-speaker dialog (inside the first square speaker A and B), multi-speaker dialog (speaker A, B and C).

Since a scene is generally characterized by a repetitive visual structure, the technique extracts all video segments that possess the same features. A scene is determined in four steps:

*Step 1*: Shot sinks generation using the window-based sweep algorithm. A shot sink is defined as a pool of shots which are temporally close and visually similar. To compare the visual similarity of two shots a set of keyframes is selected for each shot using the technique presented in [Lienhart97]. The similarity between two shots is determined based on the Euclidian distance between color histograms in the RGB color space.

*Step 2*: Sink clustering and characterization into one of the three predefined classes: periodic, partially-periodic and non-periodic based on the shot repetition degree.

*Step 3*: Scene extraction and classification. At this stage, the scenes are extracted by grouping temporally overlapped sinks into one event. In this way, the shots which are semantically inter-related with each other will belong to the same scene, since different events should have different thematic topics, so different scenes. If an event contains two periodic shot sinks, a 2-speaker dialog is detected. If the event contains several partly periodic sinks a multi-speaker dialog is identified. All the remaining events are labeled as hybrid scenes.

*Step 4*: Post-processing phase that integrates speech and face cues in order to remove false alarms presented in events identification.

In a general manner, the approaches based on semantic representations of the scene content suffer mainly from the too restrictive definition of the scenes and events considered, which cannot respond effectively to the variety of concepts encountered in real-life videos.

A last category of methods exploits a set of temporal constraints. Such approaches are described in the following section.

### 4.1.6. Temporally constrained techniques

The underlying principle consists of imposing a temporal distance constraint with the help of a sliding window which selects the maximum total number of shots that can be possibly included into the same scene. In other words, only the shots situated in the temporal window and satisfying certain similarity constraints can be grouped into the same video scene. The proposed criterion is used in order to avoid grouping into the same scene shots that exhibit a high similarity of the visual content but located at a distance greater than the specified parameter. This principle is illustrated in Figure 4.5.

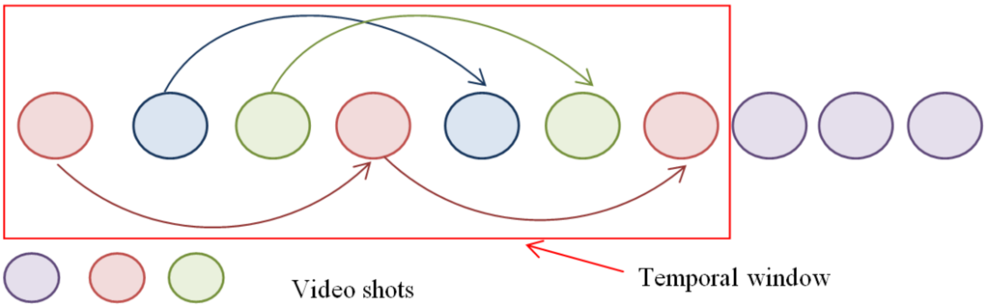


Figure 4.5. Shot grouping based on temporal constrained window.

Naturally; the window length has a strong impact on the segmentation results. On one hand, if the selected size is too large, shots from two or more scenes will be grouped together in a same scene. In this case, the number of missed detected scenes will be very large causing an under-segmentation of the original video flow. On the other hand, a smaller value of the sliding window, will determine shots from the same scene to be attached to different scenes. In this case, the number of false alarms is increasing and the input movie will be over-segmented.

As representative of this category of approaches, let us first mention the method introduced in [Zhu09], which exploits a spatio-temporal shot clustering technique. The technique starts by performing shot detection and keyframe extraction using the gray-level variance histogram and wavelet texture variance histogram.

The similarity between shots should reflect the correlation between visual content elements: such as locations, persons and events. In order to match the information between shots the authors propose using the pair of keyframes which returns the highest value of the reciprocal Bhattacharya distance between normalized color histograms in the RGB color space.
The temporal locality constraint is characterized with the help of a parameter that sets the time-window length, denoted by $T_{window}$ , and representing the maximum number of shots that a scene can include.

Moreover, the content of each segment is further classified into one of the following general categories:

1. *Conversational scenes*: include in their structure shots with faces or objects with similar spatial position and size; they are characterized by a succession of shots that show low visual content variation and reduced camera/object motion.
2. *Suspense scenes:* characterized by a succession of shots that show average motion in time, low audio energy and visual content variation but that are followed by a sudden change either in the sound track or in the activity intensity, or in most of the cases in both parameters.

3. *Action scenes:* composed of a succession of shots with reduced temporal duration, but with intensive motion activity or audio energy.

The technique introduced in [Sakarya10] is also based on a temporal interval that is set to 9 shots. For each position of the sliding window, a similarity matrix between sets of keyframes associated to each shot is determined. The similarity matrix is used in order to optimize the objective function that partition a graph into two sets based on the min-max algorithm. A scene boundary is associated with the resulting partition. In order to reduce the total number of false scene boundaries, a filtering process is introduced. Finally, a clustering operation is performed by using two distinct grouping strategies: k-means clustering and dominant set framework.

The technique introduced in [Chasanis09] is based on histograms of visual words and temporal distances. The author claims that the color histogram fail to describe the connectivity between shots in the case of a rapid change of the visual content So, in order to avoid this situation they propose using two other descriptors invariant to transformation as rotation and scale: SIFT and CCH (color context histogram). For each shot, a different number of descriptors is computed that describe certain objects or interest points within the given shot. Then, a visual word is defined as a set of descriptors, selected from each keyframes, that are concatenated to represent the whole shot.

To extract the visual word, the set of descriptors associated to *N* videos shots is cluster intro *k* groups (where *k* denotes the total number of visual words considered). Thus, given a shot and its associated descriptors a visual word histogram is determined (Figure 4.6). Next, the similarity between shots is established in order to detect the scene and chapter boundaries.

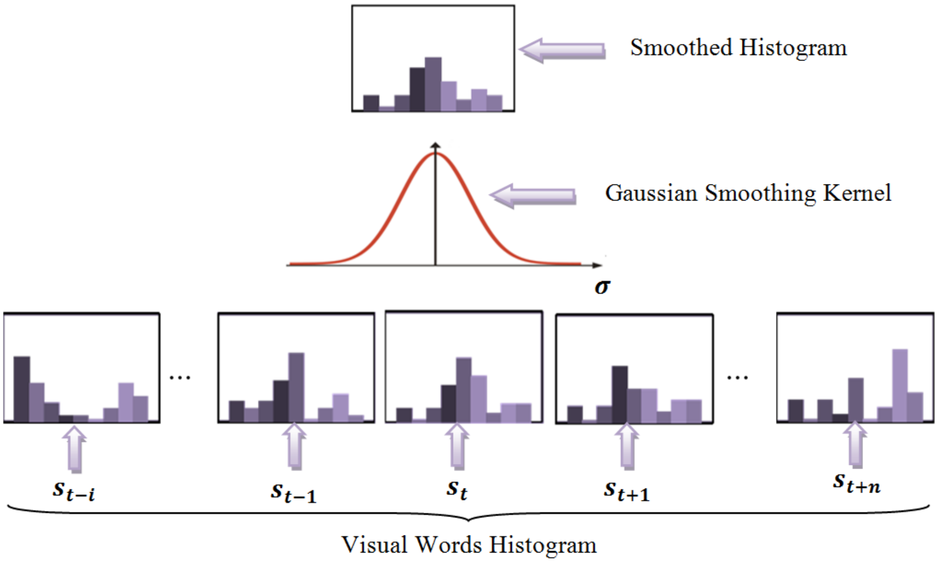


Figure 4.6. Temporal smoothing of visual words histograms.

Authors propose to consider the video shots as the words of a document that compose a paragraph (scene) that further compose a book chapter (DVD chapter), describing a specific theme. A Gaussian smoothing kernel is used to smooth temporally the visual words histogram of a shot with respect to the histograms of neighboring shots.

The temporal constraints impose a maximum number of shots to be included in a shot/DVD chapter, 8 and 16 respectively. The scene boundaries are detected by using the Euclidean distance between successive smoothed histograms.

Some of methods presented above [Tavanapong04], [Yeung98], [Rasheed05] are based on the assumption that the scene background remains unchanged and use the location of an anchor persons/object inside the scene. In this case, the shots of anchor persons/objects are shown in certain time interval which helps locating the boundaries. However, such methods were dedicated by construction to the specific characteristics of the news video. Other techniques [Zhu09] are focused on classifying the shots into particular common categories. Such categories are not available for any type of videos, such as home videos or feature films. On the other hand, other methods [Lienhart97] [Huang98], do not fully taken into account the characteristic of film editing such as the linking of shots and scenes.

A novel scene detection algorithm based on hierarchical clustering is introduced next. The technique adopts a shot grouping method that exploits temporal constraints, adaptive thresholds, neutralized shots and a new similarity measure between two shots.

## 4.2. PROPOSED APPROACH

The keyframes selected based on the leap-extraction method, introduced in Section 3.2.1 are exploited to form scenes defined as a collection of shots that present the same theme and share similar coherence in space and time [Truong07].

We have considered two different types of low level features to characterize the visual structure of a keyframe. In the first case we used the traditional HSV color histogram extracted at image level. In the second case we applied the scale invariant features transform (SIFT) [Lowe04] to each frame belonging to the static summary.

### 4.2.1. Scene / DVD chapter extraction

The proposed algorithm can distinguish between semantic continuous actions and actual scenes brakes by using a temporally distance-constrained analysis. The technique is based on a novel clustering procedure which consists of iteratively merging shots falling into a temporal analysis window and satisfying certain grouping criteria.

The temporal constraint selection has a major impact on the final segmentation result. A larger value of this parameter will determine clustering shots from different scenes together (under-segmentation), while a smaller value cause shots from the same scene to be labeled as distinct (over-segmentation). The first major contribution brought by the proposed technique is given by the adaptive modality of selecting the temporal distance parameter (*dist),* which is set proportional to the ratio between the total number of frames of the video sequence and the number of shot boundaries detected in the movie (*i.e.*, the average number of frames per shot).

$$dist = \alpha \cdot \frac{Total\, number\ of\ frames}{Total\, number\ of\ shots} \tag{4.13}$$

where $\alpha$ denotes a user-defined parameter.

We consider further that a scene $S_l$ is completely described by its constituent shots:

$$S_l : s(S_l) = \{s_{l,p}\}_{p=1}^{N_l} \longrightarrow \left\{\{f_{l,p,i}\}_{i=1}^{n_{l,p}}\right\}_{p=1}^{N_l} \tag{4.14}$$

where $S_l$ denotes the $l$th video scene, $N_l$ the number of shots included in scene $S_l$, $s_{l,p}$ the $p$th shot in scene $S_l$, and $f_{l,p,i}$ the $i$th keyframe of shot $s_{l,p.}$ containing $n_{l,p}$ keyframes.

The proposed scene change detection algorithm based on shot clustering consists of the following steps:

***Step 1****: Initialization* – The first shot of a film is automatically assigned to the first scene $S_1$. Scene counter $l$ is set to 1.

***Step 2****: Shot to scene comparison* – Consider as current shot $s_{crt}$ the first shot which is not yet assigned to any of the already detected scenes. Detect the sub-set Ω of scenes anterior to $s_{crt}$ and located at a temporal distance inferior to parameter *dist*. Compute the visual similarity between the current shot and each scene $S_k$ in the sub-set Ω, as described in the following equation:

$$\forall\ S_k \in \Omega, SceneShotSim(s_{crt}, S_k) = \frac{n_{matc\ hed}}{n_{k,p} \cdot N_k \cdot n_{crt}} \tag{4.15}$$

where $n_{crt}$ is the number of keyframes of the considered shot and $n_{matched}$ represents the number of matched keyframes of the scene $S_k$. A keyframe from scene $S_k$ is considered to be matched with a keyframe from shot $s_{crt}$ if a given *visual similarity measure* between the two keyframes is superior to a threshold $T_{group}$. Let us note that a keyframe from the scene $S_k$ can be matched with multiple frames from the current shot.

Concerning the visual similarity measure involved in the above-described process, we have considered two different approaches, based on:

(1) chi-square distance between HSV color histograms;

(2) the number of matched interest points determined based on SIFT descriptors with a KD-tree matching technique [Lee07].

Finally, the current shot $s_{crt}$ is identified to be similar to the scene $S_k$ if:

$$SceneShotSim(s_{crt}, S_k) \geq 0.5 \tag{4.16}$$

In this case, the current shot $s_{crt}$ will be clustered in the scene $S_{k.}$ In the same time, all the shots between the current shot and the scene $S_k$ will also be attached to scene $S_k$ and marked as *neutralized*. Let us note that the scenes to which initially belonged such neutralized shots disappear (in the sense that they are merged to the scene $S_k$). The list of detected scenes is then updated.

The neutralization process allows us to identify the most representative shots for a current scene (Figure 4.7.a and 4.7.b), which are the remaining non-neutralized shots. In this way, the influence of outlier shots which might correspond to some punctual digressions from the main action in the considered scene is minimized.

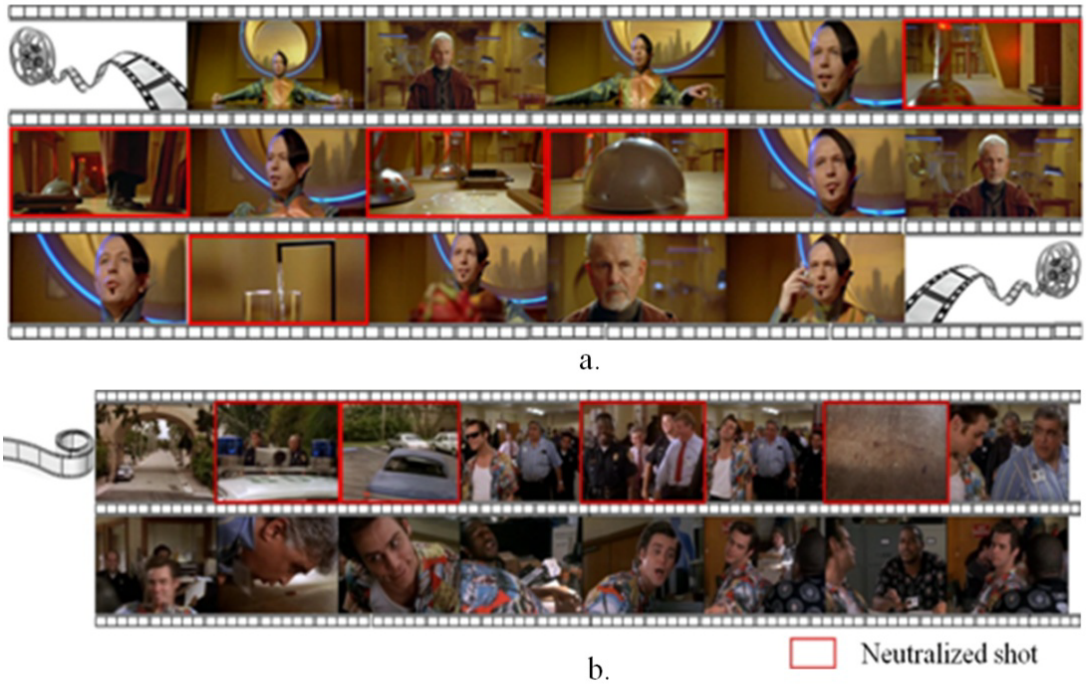


Figure 4.7. Neutralizing shots (marked with red) based on visual similarity:
(a) The 5th element; (b) Ace Ventura the pet detective.

If the condition described in Equation (4.16) is not satisfied, go to step 3.

***Step 3****: Shot by shot comparison* – If the current shot ($s_{crt}$) is *highly* similar (*i.e.*, with a similarity at least two times bigger than the grouping threshold $T_{group}$) with a shot of any scene in the sub-set Ω determined at step 2, then $s_{crt}$ is merged in the corresponding scene together with all the intermediate shots. If $s_{crt}$ is found highly similar to multiple other shots, than the scene which is the most far away from the considered shot (in the sense of the temporal distance) is retained.

Both the current shot and all its highly similar matches are unmarked and for the following clustering process will contribute as normal, non-neutralized shots (Figure 4.8). This step ensures that shots highly similar with other shots in the previous scene to be grouped into this scene and aims at reducing the number of false alarms.

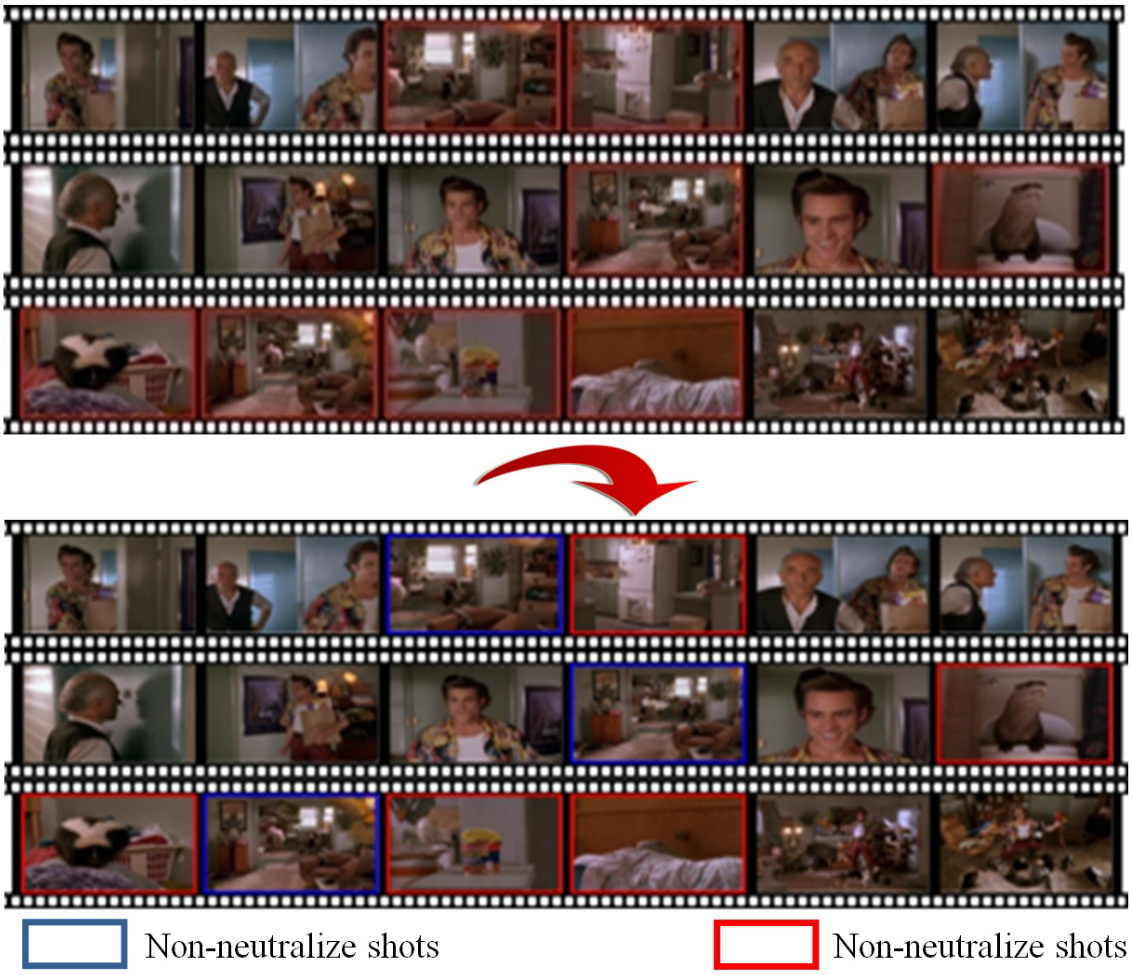


Figure 4.8. Shot non-neutralization based on high similar value.

***Step 4***: *Creation of a new scene* – If the current shot $s_{crt}$ does not satisfy any of the similarity criteria in steps 2 and 3, a new scene, including $s_{crt}$, is created.

***Step 5***: *Refinement* - At the end, scenes including only one shot are attached to the adjacent scenes depending on the maximum similarity value. In the case of the first scene, this is grouped with the following one by default.

The grouping threshold $T_{group}$ is adaptively established depending on the input video stream visual content variation as the average chi-square distance / number of interest points between the current keyframe and all anterior keyframes located at a temporal distance smaller then parameter *dist*.

### 4.2.2. Experimental evaluation

The evaluation of our scene change detector algorithm is done on TRECVID 2001 and 2002 video corpus (Figure 2.27) and on the set of 6 sitcoms and 6 Hollywood movies introduced in the automatic static summary development stage (Figure 4.9). The selected videos are also used for evaluation purposes in the state of the art algorithms presented in [Rasheed05], [Chasanis09] and [Zhu09].

Figure 4.9. Extended video database

At the beginning, for all videos, a manual segmentation has been performed in order to constitue a ground truth : here, 10 human observers have manually detected the scene boundaries. Let us note that the process can be influenced by subjectivity and depends on each individual perception. In order to establish a reliable benchmark we selected as correct only the scenes identified by all the observers.

Figure 4.10 and Figure 4.11 illustrate some examples of scene boundary detection, obtained with both the SIFT and HSV -based approaches.

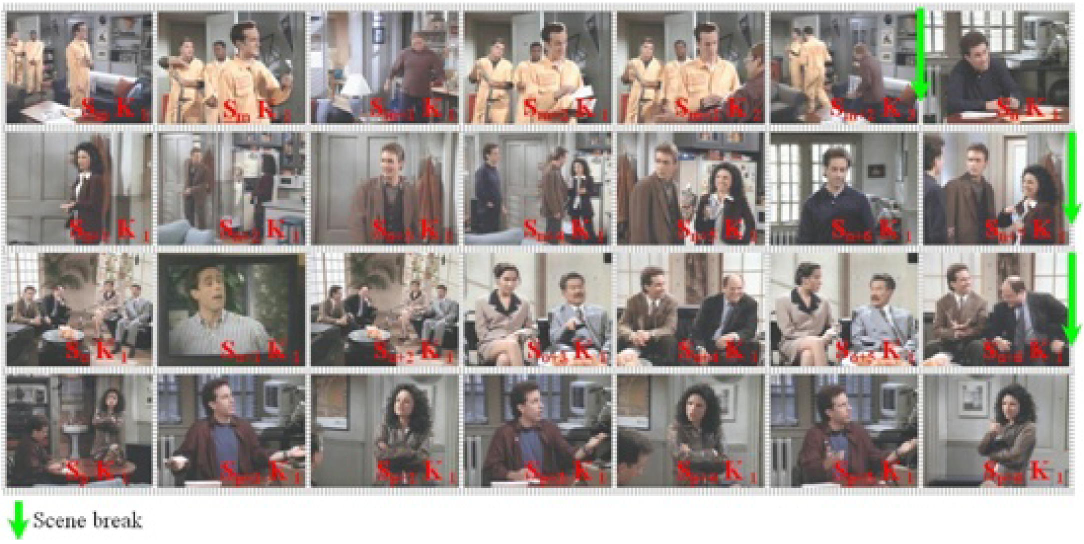


Figure 4.10. Detected scenes when using the interest points extracted based on SIFT descriptor.

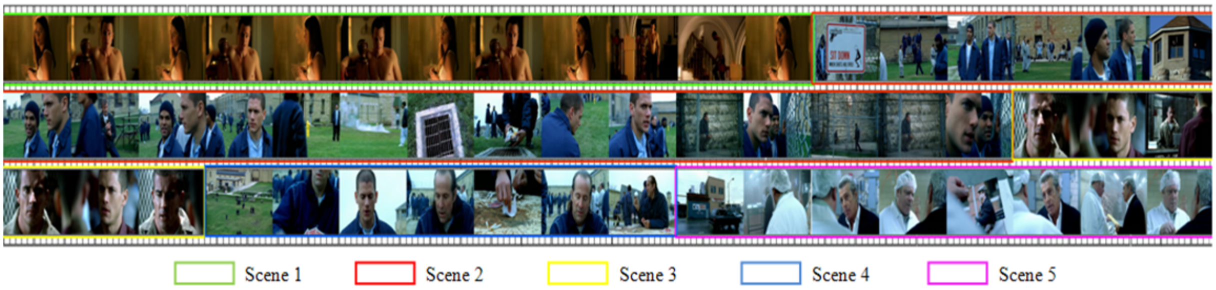


Figure 4.11. Detected scenes when using the HSV color histogram features.

We can observe that in all cases the scenes have been correctly identified. Table 4.1 and 4.2 summarize the scene detection results obtained when using to determine the visual similarity between keyframe the SIFT descriptors and HSV color histogram comparisons, respectively. As it can be observed, the detection efficiency is comparable in both cases. The α parameter (equation 4.13) was set here to a value of 7.

Table 4.1. Scene detection based on SIFT features.

| **Current number** | **Campaign** | **Video name** | **Ground truth (Human observers)** | **Detected** | **False Alarms** | **Missed Detected** | **Precision (%)** | **Recall (%)** | **F1 (%)** |
|---|---|---|---|---|---|---|---|---|---|
| 1. | TrecVid 2001 and 2002 | NAD57 | 4 | 3 | 0 | 1 | 100 | 75.00 | 85.71 |
| 2. | | NAD55 | 14 | 10 | 0 | 4 | 100 | 71.42 | 83.33 |
| 3. | | NAD58 | 5 | 4 | 3 | 1 | 57.14 | 80.00 | 66.66 |
| 4. | | UGS01 | 5 | 4 | 3 | 1 | 57.14 | 80.00 | 66.66 |
| 5. | | UGS09 | 14 | 13 | 2 | 1 | 86.66 | 92.85 | 89.65 |
| 6. | Sitcoms | SF | 24 | 19 | 1 | 5 | 95.00 | 79.16 | 86.36 |
| 7. | | TM | 22 | 18 | 0 | 4 | 100 | 81.81 | 90.00 |
| 8. | | PB | 39 | 31 | 3 | 8 | 91.17 | 79.48 | 84.93 |
| 9. | | AM | 32 | 28 | 11 | 4 | 71.79 | 87.50 | 78.87 |
| 10. | | SC | 20 | 17 | 0 | 3 | 100 | 85.00 | 91.89 |
| 11. | | FR | 17 | 17 | 7 | 0 | 70.83 | 100 | 82.92 |
| 12. | Hollywood movies | 5E | 63 | 55 | 24 | 8 | 69.62 | 87.30 | 77.46 |
| 13. | | AV | 36 | 34 | 11 | 2 | 75.55 | 94.44 | 83.95 |
| 14. | | LW | 67 | 63 | 39 | 4 | 61.76 | 94.02 | 74.55 |
| 15. | | T2 | 66 | 61 | 11 | 5 | 84.72 | 92.42 | 88.41 |
| 16. | | MK | 44 | 40 | 5 | 4 | 88.88 | 90.91 | 89.88 |
| 17. | | HA2 | 68 | 56 | 6 | 12 | 90.32 | 82.35 | 86.15 |
| **TOTAL** | | | **540** | **473** | **126** | **67** | **78.96** | **87.59** | **83.05** |

*Note:* In the case of scene change detection we considered a tolerance of 10% from the actual size of the ground truth scene when identifying the position of a scene brake (for both HSV color histogram and SIFT interest points).

The analysis of the experimental results presented in Figure 4.12, 4.13 and 4.14 leads to the following conclusions.

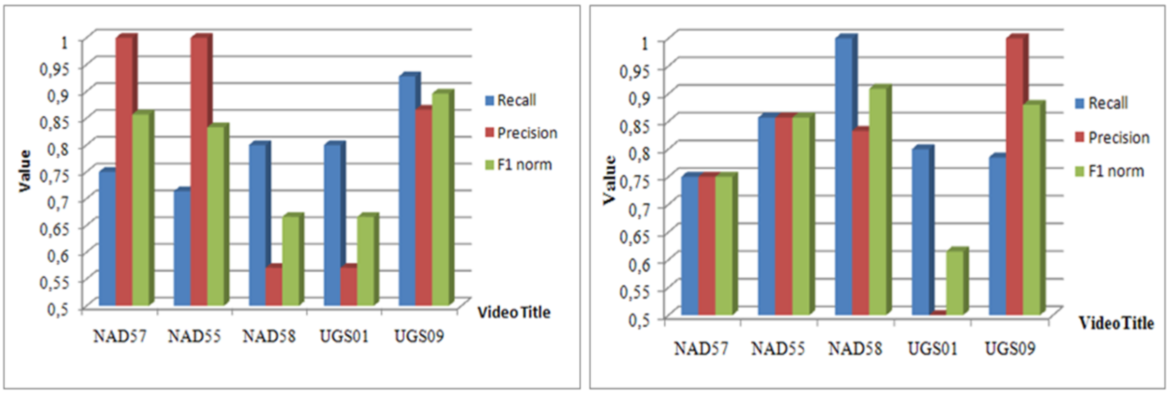
1. The keyframe similarity based on HSV color histogram is much faster than the SIFT extraction process (the average time for the histogram extraction is 0.5 seconds/image while for the interest points the extraction time is 2 seconds/image) and can be used when feature detection and matching becomes difficult due to the complete change of

the background, important variation of the viewing point, or the action development (Figure 4.15).

2. The matching technique based on interest points is better suited for scenes that have undergone some great changes but where some persistent, perennial features (such as objects of interest) are available for extraction and matching. In this case, the technique is robust to abrupt changes in the intensity values introduced by noise or changes in the illumination condition (Figure 4.16).

Table 4.2. Scene detection based on HSV color histogram.

| **Current number** | **Campaign** | **Video name** | **Ground truth (Human observers)** | **Detected** | **False Alarms** | **Missed Detected** | **Precision (%)** | **Recall (%)** | **F1 (%)** |
|---|---|---|---|---|---|---|---|---|---|
| 1. | TrecVid 2001 and 2002 | NAD57 | 4 | 3 | 1 | 1 | 75.00 | 75.00 | 75.00 |
| 2. | | NAD55 | 14 | 12 | 2 | 2 | 85.71 | 85.71 | 85.71 |
| 3. | | NAD58 | 5 | 5 | 1 | 0 | 83.33 | 100 | 90.90 |
| 4. | | UGS01 | 5 | 4 | 4 | 1 | 50.00 | 80.00 | 61.53 |
| 5. | | UGS09 | 14 | 11 | 0 | 3 | 100 | 78.57 | 88.00 |
| 6. | Sitcoms | SF | 24 | 20 | 0 | 4 | 100 | 83.33 | 90.90 |
| 7. | | TM | 22 | 17 | 2 | 5 | 89.47 | 77.27 | 82.92 |
| 8. | | PB | 39 | 33 | 0 | 6 | 100 | 84.61 | 91.66 |
| 9. | | AM | 32 | 21 | 4 | 8 | 84.00 | 72.41 | 77.77 |
| 10. | | SC | 20 | 15 | 1 | 5 | 93.75 | 75.00 | 83.33 |
| 11. | | FR | 17 | 17 | 7 | 0 | 70.83 | 100 | 82.92 |
| 12. | Hollywood movies | 5E | 63 | 49 | 10 | 9 | 83.05 | 84.48 | 83.76 |
| 13. | | AV | 36 | 26 | 2 | 7 | 92.85 | 78.78 | 85.24 |
| 14. | | LW | 67 | 64 | 25 | 3 | 71.97 | 95.52 | 82.05 |
| 15. | | T2 | 66 | 60 | 7 | 6 | 89.55 | 90.90 | 90.22 |
| 16. | | MK | 44 | 38 | 7 | 6 | 84.44 | 86.36 | 85.39 |
| 17. | | HA2 | 68 | 50 | 5 | 11 | 90.90 | 81.96 | 86.20 |
| **TOTAL** | | | **540** | **445** | **78** | **77** | **85.08** | **85.25** | **85.17** |



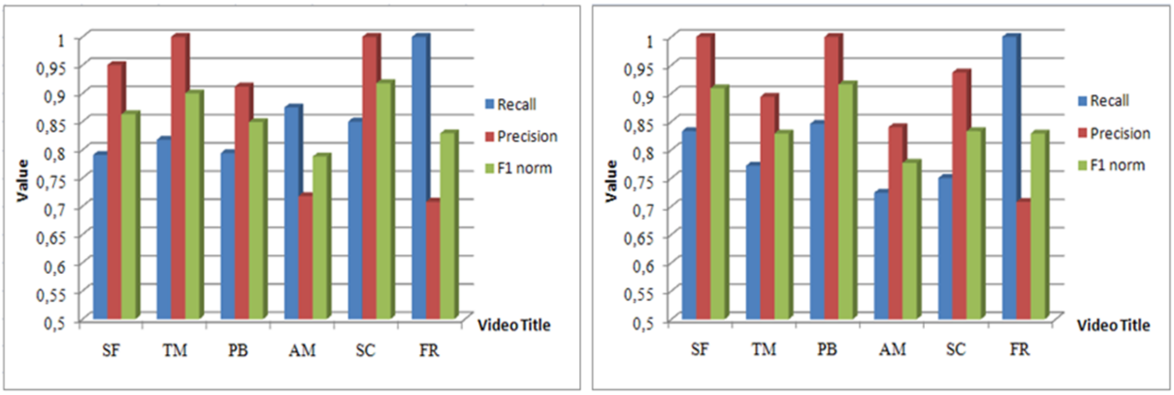
a. b.

Figure 4.12. Recall, Precision and F1 norm rates for the proposed technique when tested on TRECVid 2001 and 2002 databases: (a) interest points; (b) HSV color histogram.

The average precision and recall rates are the following:

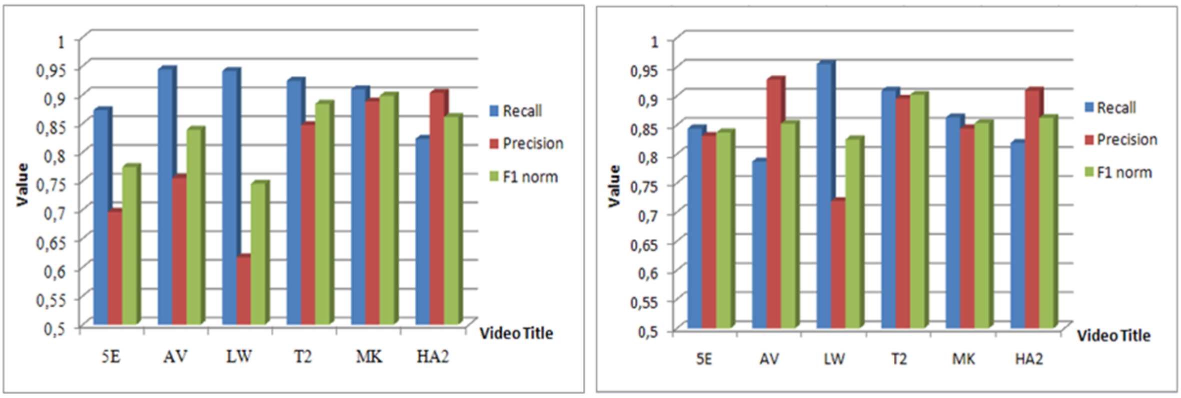
- *R*=88% and *P*=78%, for the SIFT-based approach, and
- *R*=85% and *P*=85%, for the HSV histogram approach.

These results demonstrate the superiority of the proposed scene detection method with respect to existing state of the art techniques [Rasheed05], [Chasanis09] and [Zhu09], which provide precision/recall rates between 82% and 77%.



a. b.

Figure 4.13. Recall, Precision and F1 norm rates for the proposed technique when tested on sitcoms: (a) interest points; (b) HSV color histogram.



a. b.

Figure 4.14. Recall, Precision and F1 norm rates for the proposed technique when tested on Hollywood movies: (a) interest points; (b) HSV color histogram.

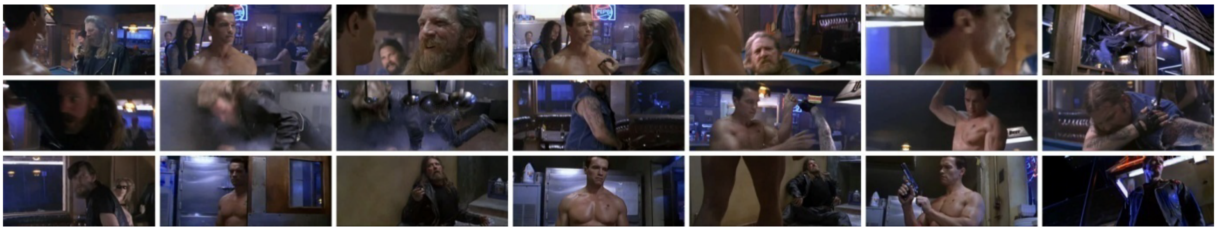


Figure 4.15. Video scene correctly identified using the HSV color histogram.

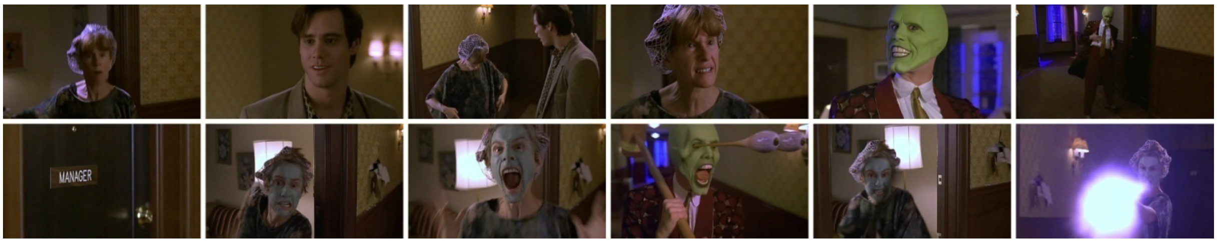
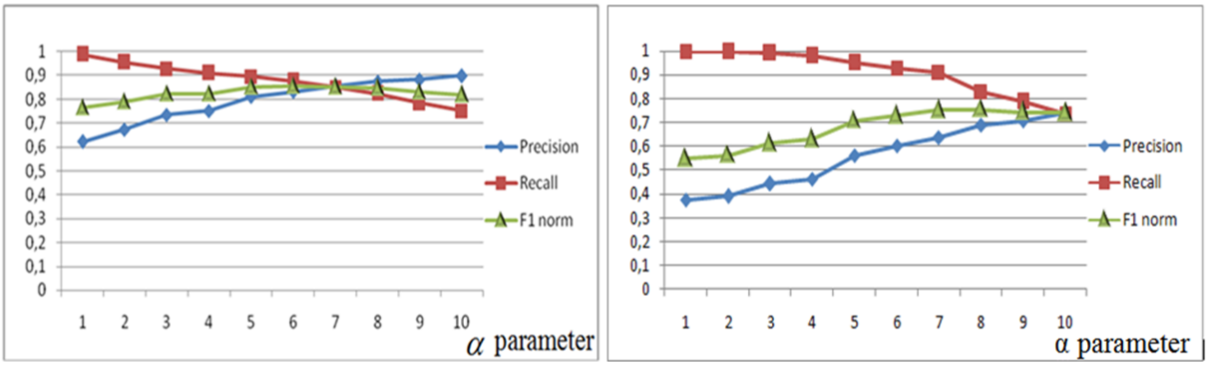


Figure 4.16. Video scene correctly identified using the interest points.

We also analyzed the impact of the different temporal constraints lengths (Equation (4.13)) on the proposed scene detection method. Figure 4.17 presents the precision, recall and *F1* scores obtained for various values of the α parameter when using the both type of descriptors.



a. b.

Figure 4.17. Precision, recall and F1 score variation for different α values.

As it can be noticed, a value between 5 and 10 returns quite similar results in terms of the overall efficiency.

We can observe that increasing the α parameter lead to lower recall rates. That means that for higher values of the α parameter, different scenes are grouped within a same one. In the same time, the number of false alarms (*i.e.*, false scene breaks) is reduced.

This observation led us to investigate the utility of our approach for a slightly different application, related to DVD chapter detection. For the considered Hollywood videos, the DVD chapters were identified by movie producers and correspond to access points in the considered video. The DVD chapters are highly semantic video units with low level of detail containing a scene ore multiple scenes that are correlated based on a purely semantic meaning.

Table 4.3 and 4.4 summarizes the DVD chapter results obtained when using to determine the visual similarity between keyframe the interest points extracted based on SIFT descriptors and HSV color histogram comparisons, respectively.

Table 4.3. DVD chapter detection based on SIFT features for Hollywood movies temporally segmented by producers.

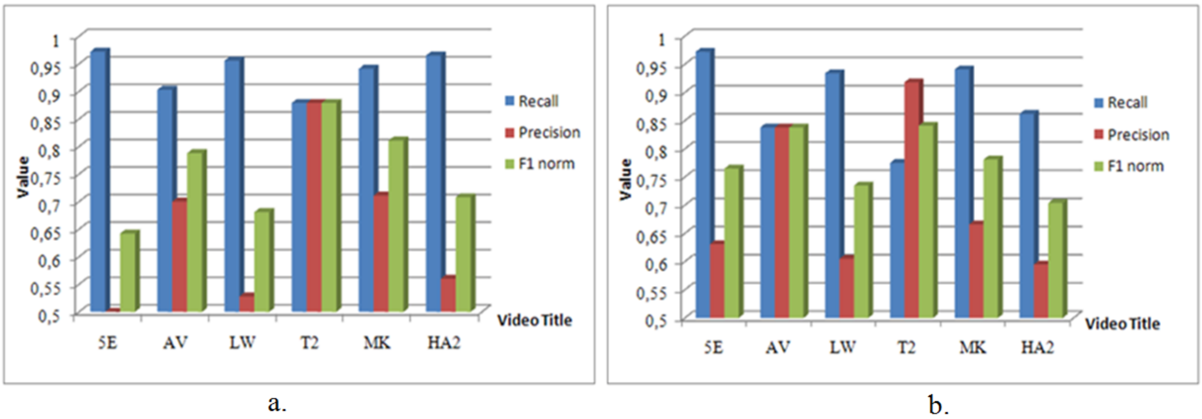
| **Current number** | **Campaign** | **Video name** | **DVD chapters (Movie producer)** | **Detected** | **False Alarms** | **Missed Detected** | **Precision (%)** | **Recall (%)** | **F1 (%)** |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Hollywood movies | 5E | 37 | 36 | 39 | 1 | 48.00 | 97.29 | 64.28 |
| 2. | | AV | 31 | 28 | 12 | 3 | 70.00 | 90.32 | 78.87 |
| 3. | | LW | 46 | 44 | 41 | 2 | 52.87 | 95.65 | 68.09 |
| 4. | | T2 | 58 | 51 | 7 | 7 | 87.93 | 87.93 | 87.93 |
| 5. | | MK | 34 | 32 | 12 | 2 | 71.11 | 94.11 | 81.01 |
| 6. | | HA2 | 29 | 28 | 22 | 1 | 56.00 | 96.55 | 70.88 |
| **TOTAL** | | | **235** | **219** | **133** | **16** | **62.42** | **93.19** | **74.76** |

Table 4.4. DVD chapter detection based on HSV features for Hollywood movies temporally segmented by producers.

| Current number | Campaign | Video name | DVD chapters (Movie producer) | Detected | False Alarms | Missed Detected | Precision (%) | Recall (%) | F1 (%) |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Hollywood movies | 5E | 37 | 36 | 21 | 1 | 63.15 | 97.29 | 76.58 |
| 2. | | AV | 31 | 26 | 5 | 5 | 83.87 | 83.87 | 83.87 |
| 3. | | LW | 46 | 43 | 28 | 3 | 60.56 | 93.47 | 73.50 |
| 4. | | T2 | 58 | 45 | 4 | 13 | 91.83 | 77.58 | 84.11 |
| 5. | | MK | 34 | 32 | 16 | 2 | 66.66 | 94.11 | 78.04 |
| 6. | | HA2 | 29 | 25 | 17 | 4 | 59.52 | 86.20 | 70.41 |
| **TOTAL** | | | **235** | **207** | **91** | **28** | **69.46** | **88.08** | **77.66** |

The value of the α parameter has been here set to 10. The average recall (*R*) and precision (*P*) rates obtained in this case are (Figure 4.18):

- *R*=93% and P=62%, for the SIFT-based approach, and
- *R*=68% and *P*=87%, for the HSV histogram approach.



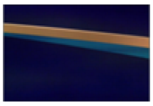
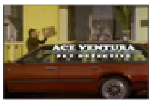
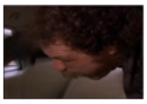
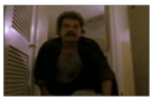
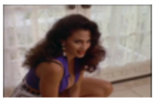
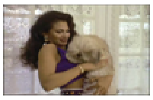
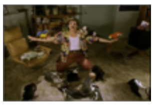
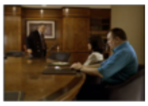
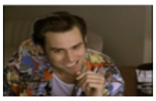
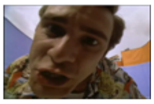
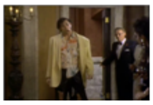
a. b.

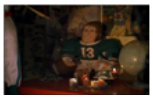
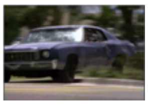
Figure 4.18. Recall, Precision and F1 norm rates for DVD extraction techniques tested on Hollywood movies: (a) interest points; (b) HSV color histogram.

Such a result is quite competitive with the state of the art techniques introduced in [Rasheed05], [Chasanis09] and [Zhu09] which yield precision/recall rates varying between 65% and 72%. As it can be observed, when extracting interest points the recall rates are superior to those obtained for the HSV color histogram but, in this case, the number of false alarms becomes extensively high, reducing dramatically the method precision. Another aspect that needs to be taken into consideration is the computational time for the SIFT extraction procedure (*e.g.* for 1500 keyframes – 86400 s) which is important for real-time applications.

In Table 4.5 we presented the experimental results obtained on the “Ace Ventura – The pet detective” Hollywood movie. In the present table we give also the DVD chapter keyframe automatically selected by the proposed temporal segmentation framework and the one manually chosen by movie producers. In our case, the most representative image for a DVD chapter is selected as cluster centroid. The presented results are obtained when extracted as visual feature the HSV color histograms.

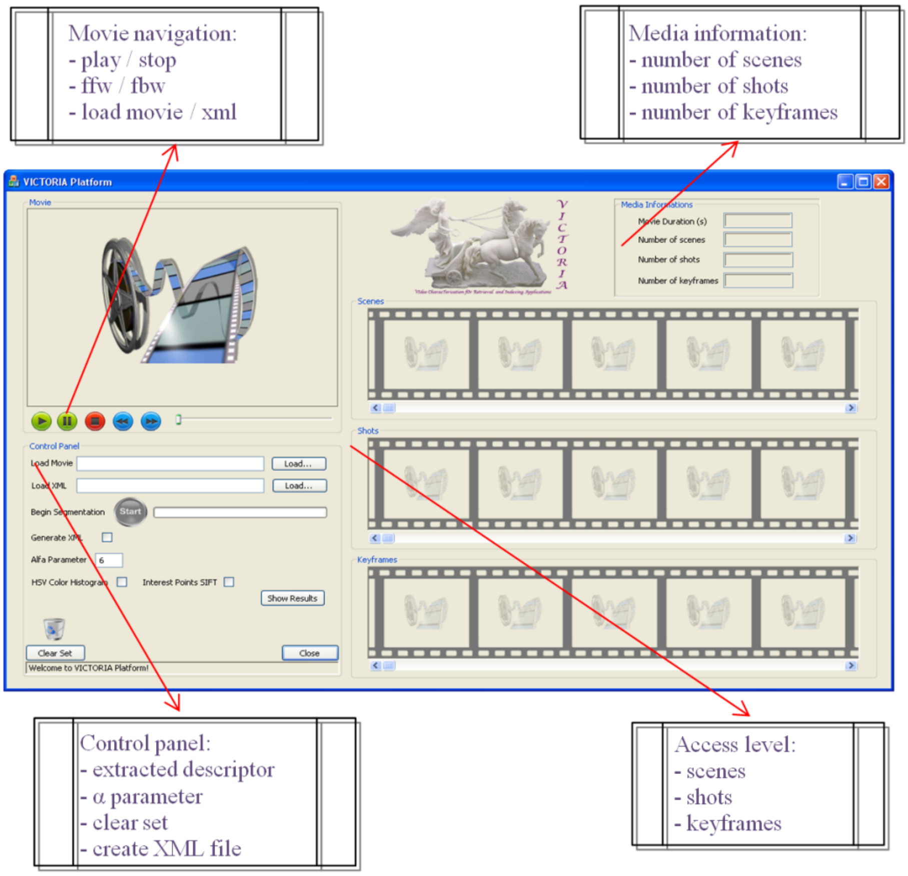
Table 4.5. DVD chapter detection for “Ace Ventura – The Pet Detective” movie.

| **Crt. Nr.** | 1 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 2 | **Selected Keyframe** | **DVD Keyframe** |
|---|---|---|---|---|---|---|---|
| **DVD Chapter** | Begin Titles |  | | **DVD Chapter** | Special Delivery | |  |
| **TL** | 0:15 | | | **TL** | 3:38 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 3 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 4 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Dissatisfied |  |  | **DVD Chapter** | Satisfied Client |  |  |
| **TL** | 2:08 | | | **TL** | 0:55 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 5 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 6 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Kidnapped! | | | **DVD Chapter** | Ace's Kingdom | |  |
| **TL** | 1:05 | | | **TL** | 3:12 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 7 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 8 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Pet Detection | |  | **DVD Chapter** | Ace on Case | |  |
| **TL** | 2:15 | | | **TL** | 1:45 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 9 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 10 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Tanked Up | |  | **DVD Chapter** | New Trainer | | |
| **TL** | 2:22 | | | **TL** | 1:50 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 11 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 12 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Lt. Einhorn | | | **DVD Chapter** | Camp's Party | | |
| **TL** | 4:42 | | | **TL** | 2:37 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 13 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 14 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Face of the Enemy | | | **DVD Chapter** | "Do Not Go In!" | |  |
| **TL** | 2:46 | | | **TL** | 4:56 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | +2 | | |
| **Crt. Nr.** | 15 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 16 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Ring Wearers | | | **DVD Chapter** | She's All Right | | |
| **TL** | 3:27 | | | **TL** | 1:50 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | +1 | | | **FA** | - | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Crt. Nr.** | 17 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 18 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | At the Crime Scene | | | **DVD Chapter** | The Missing Player | | |
| **TL** | 4:22 | | | **TL** | 3:32 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 19 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 20 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | No Sleep Tonight | | | **DVD Chapter** | Finkle's House |  | |
| **TL** | 1:18 | | | **TL** | 4:37 | | |
| **DC** | X | | | **DC** | √ | | |
| **FA** | - | | | **FA** | +1 | | |
| **Crt. Nr.** | 21 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 22 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | They've Got Marino |  | | **DVD Chapter** | Einhorn Commotion | | |
| **TL** | 2:10 | | | **TL** | 3:39 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | +1 | | | **FA** | - | | |
| | | | | | | | |
| **Crt. Nr.** | 23 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 24 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Master at Work | | | **DVD Chapter** | Finkle's Grabbag | | |
| **TL** | 2:22 | | | **TL** | 3:16 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 25 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 26 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | The Truth Dawns | | | **DVD Chapter** | Tailing Einhorn | | |
| **TL** | 3:45 | | | **TL** | 2:57 | | |
| **DC** | √ | | | **DC** | √ | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 27 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 28 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Nap Time | | | **DVD Chapter** | Penalty on Ace | | |
| **TL** | 1:41 | | | **TL** | 1:48 | | |
| **DC** | X | | | **DC** | X | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 29 | **Selected Keyframe** | **DVD Keyframe** | **Crt. Nr.** | 30 | **Selected Keyframe** | **DVD Keyframe** |
| **DVD Chapter** | Trouble W/the Lady | | | **DVD Chapter** | Stripped Pretense | | |
| **TL** | 1:31 | | | **TL** | 3:52 | | |
| **DC** | √ | | | **DC** | X | | |
| **FA** | - | | | **FA** | - | | |
| **Crt. Nr.** | 31 | **Selected Keyframe** | **DVD Keyframe** | **TL – Temporal Length**<br>**DC – Detected Chapter**<br>**FA – False Alarms** | | | |
| **DVD Chapter** | Lover of Animals | | | | | | |
| **TL** | 2:25 | | | | | | |
| **DC** | √ | | | | | | |
| **FA** | - | | | | | | |

### 4.2.3. The VICTORIA Platform

When considering the problem of video semantic segmentation it is highly useful to develop appropriate user interfaces that can help to evaluate the detection performances. The proposed video structuring platform, so-called VICTORIA (*VIdeo CharacTerization Of Retrieval and Indexing Applications*) illustrated in Figure 4.19 was developed in Visual Studio C++/MFC.



Movie navigation:
- play / stop
- ffw / fbw
- load movie / xml

Media information:
- number of scenes
- number of shots
- number of keyframes

Control panel:
- extracted descriptor
- α parameter
- clear set
- create XML file

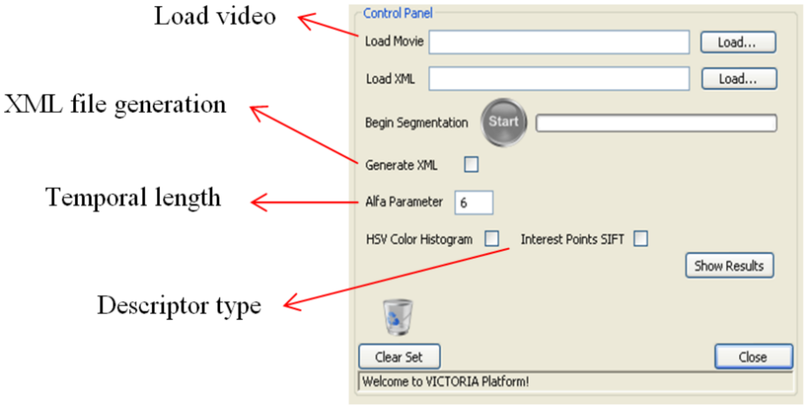
Access level:
- scenes
- shots
- keyframes

Alfa Parameter: 6

Welcome to VICTORIA Platform!

Figure 4.19. High level video temporal segmentation graphical interface.

The user can select an arbitrary input video and launch the segmentation/structuring process (Figure 4.20).



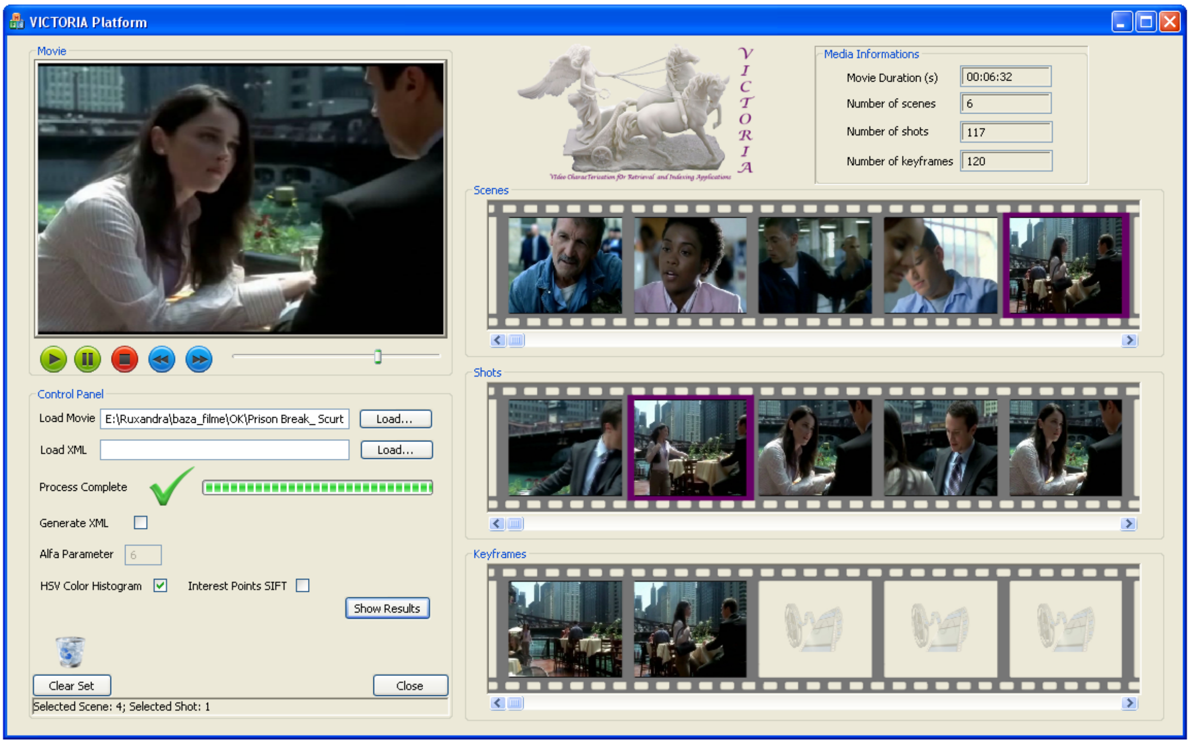
Alfa Parameter: 6

Welcome to VICTORIA Platform!

Figure 4.20. The control panel of VICTORIA platform.

After selecting a movie the user can also specify if he desires an XML file, the value for the α parameter that control the temporal length and also the descriptor type used for scene/DVD chapter extraction.

The outcome of the analysis is an XML file which specifies the shot and scene boundaries, as well as the temporal positions of the obtained keyframes. The adopted syntax is compliant with the MPEG-7 segment-based representation [ISO/IEC 15938-5]. Finally, the system allows the user to hierarchically browse the movie at each structural level (Figure 4.21).



Movie Duration (s) 00:06:32
Number of scenes 6
Number of shots 117
Number of keyframes 120

Load Movie E:\Ruxandra\baza_filme\OK\Prison Break_ Scurt
Load XML
Process Complete ✓
Generate XML □
Alfa Parameter 6
HSV Color Histogram ☑ Interest Points SIFT □

Selected Scene: 4; Selected Shot: 1

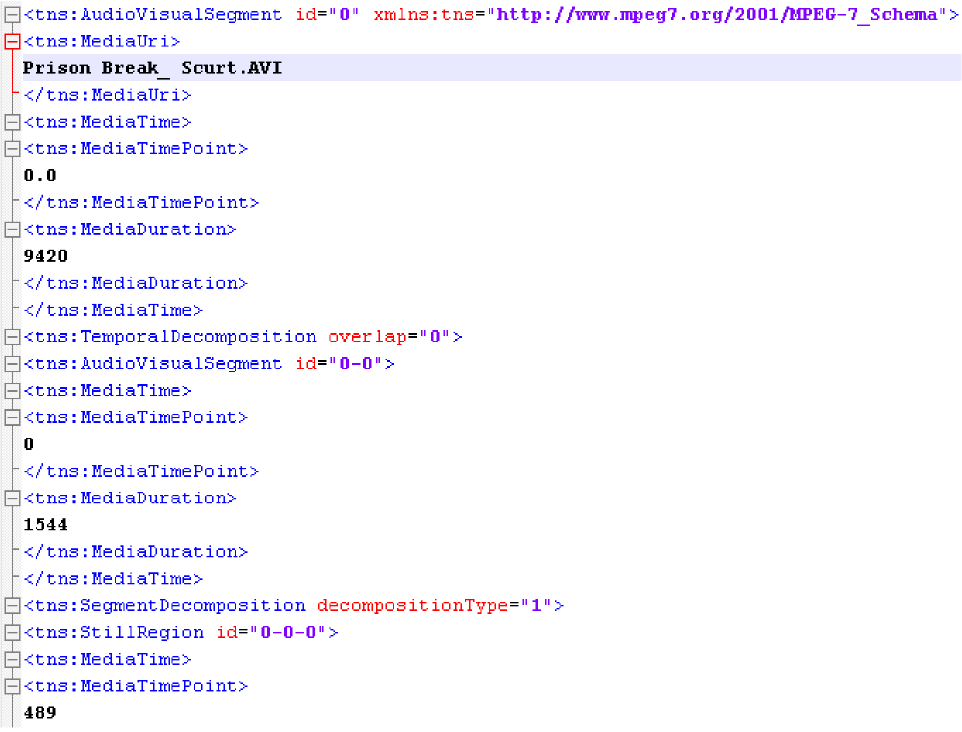
Figure 4.21. Video segmentation with the proposed system for an input video file loaded on the platform.

The XML file generated at the end of the program allow annotating the input video stream with MPEG-7 metadata. The annotated descriptors are associated with each element of the video structure starting from the scene level to shot and keyframes and stored in a XML file.

In MPEG-7 a Time Series structure describes a temporal set of descriptors extracted from a video segment and provides image to video-frame and video frames to video frames matching functionalities. Two types of Time Series are defined: Regular and Irregular. In the Regular Time Series, the descriptors are located at constant intervals within a given time span. Alternatively, for Irregular Time Series the descriptors can be located at random positions.

Based on this representation the temporal descriptor is characterized by the following set of attributes: ID, name, Access mode, descriptor length, temporal interval, the original offset and the bit stuffing sequence.

For the above video the resulted file is presented in Figure 4.22.



```
<tns:AudioVisualSegment id="0" xmlns:tns="http://www.mpeg7.org/2001/MPEG-7_Schema">
<tns:MediaUri>
Prison Break_ Scurt.AVI
</tns:MediaUri>
<tns:MediaTime>
<tns:MediaTimePoint>
0.0
</tns:MediaTimePoint>
<tns:MediaDuration>
9420
</tns:MediaDuration>
</tns:MediaTime>
<tns:TemporalDecomposition overlap="0">
<tns:AudioVisualSegment id="0-0">
<tns:MediaTime>
<tns:MediaTimePoint>
0
</tns:MediaTimePoint>
<tns:MediaDuration>
1544
</tns:MediaDuration>
</tns:MediaTime>
<tns:SegmentDecomposition decompositionType="1">
<tns:StillRegion id="0-0-0">
<tns:MediaTime>
<tns:MediaTimePoint>
489
```

Figure 4.22. Video segmentation using the MPEG-7 structuring format.

In order to ensure a large interoperability, we integrated our technique into the OVIDIUS (*On-line VIDeo Indexing Universal System*) [Bursuc10] indexing system, which is web platform developed with HTML, PHP and JavaScript technologies. The following components are included: selector of the segment type, selector of the hierarchical level of each segment, iconic representation of segments, navigation/browsing buttons, summary and keyword visualization and selection (Figure 4.23).

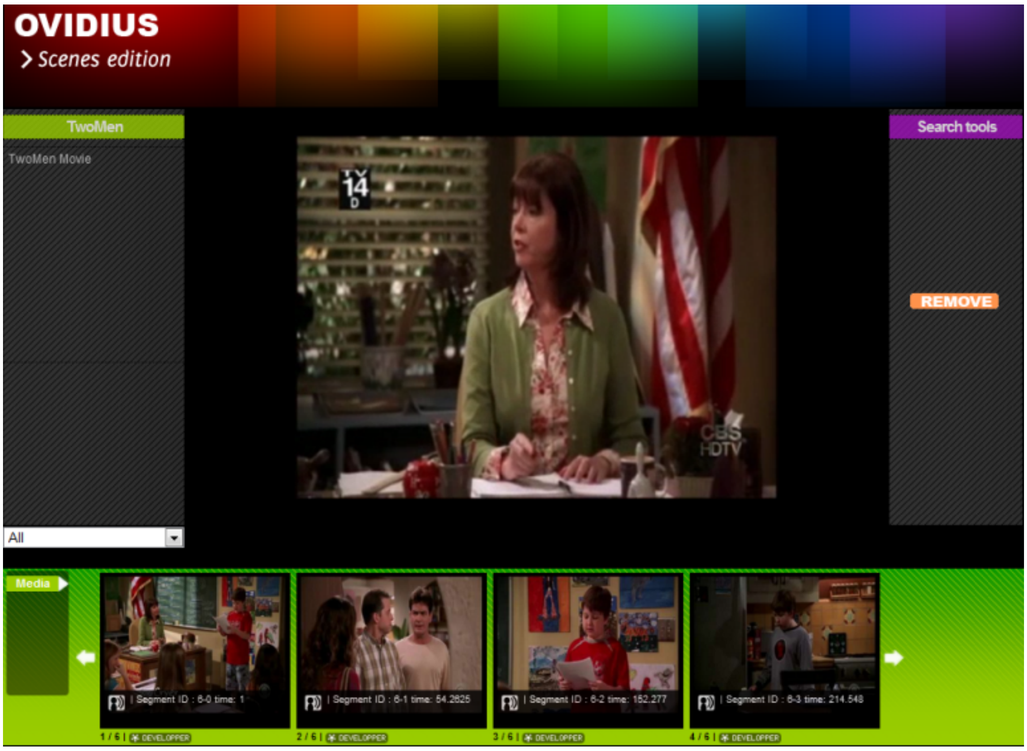
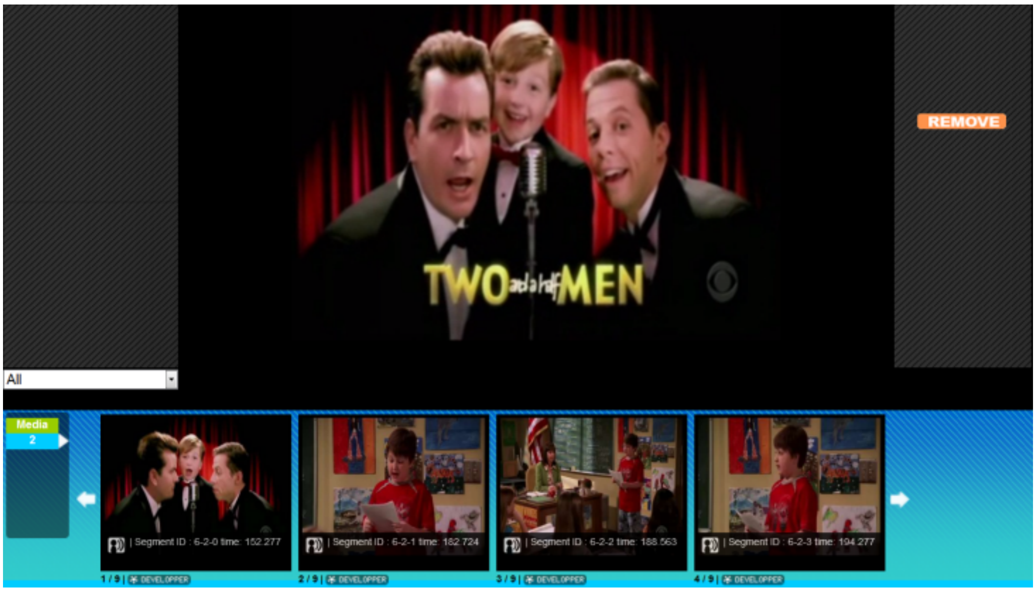


Figure 4.23. The web version of the VICTORIA platform.

One column of iconic preview representations is dedicated to each level of hierarchy of the video. Scenes, shots (Figure 4.24), and still regions (Figure 4.25) can be browsed and accessed in a hierarchical manner, as MPEG-7 segments. Additionally a color code for each level of decomposition is exploited, in order to help the user to locate him during the navigation process (*e.g.*, green for scenes, light blue for shots, dark blue for still regions).



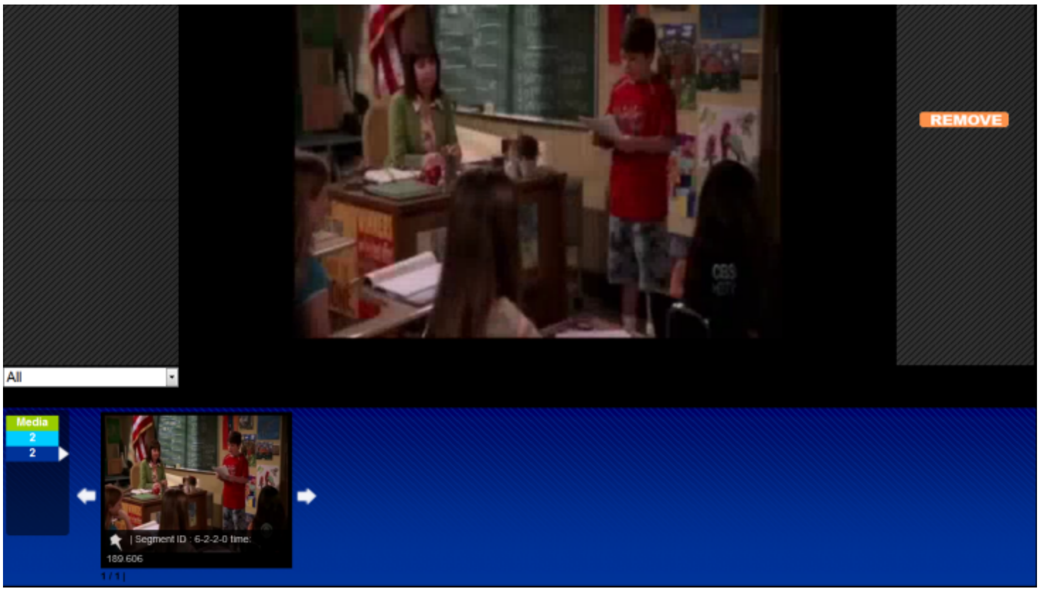
| Segment ID : 6-2-0 time: 152.277

| Segment ID : 6-2-1 time: 182.724

| Segment ID : 6-2-2 time: 188.563

| Segment ID : 6-2-3 time: 194.277

Figure 4.24. Video access at shot level.



| Segment ID : 6-2-2-0 time: 189.606

Figure 4.25. Video access at keyframe level.

## 4.3. CONCLUSIONS AND PERSPECTIVES

In the first part of this chapter we have presented the state of the art in the field of scene detection/grouping methods. We focused our attention on the most recent techniques introduced in this area and emphasized that the most promising methods, in terms of related performances, are based on a graph partition approach. In addition, we underlined the importance of combining temporal constraints and visual similarity for achieving improved performances.

Then, we have introduced a novel methodological framework for high level video temporal structuring and segmentation that extracts scenes/DVD chapters based on temporal constraints clustering, adaptive temporal lengths, neutralized shots and an adaptive thresholding mechanism.

Concerning the shot clustering into scenes, we have validated our techniques by using two types of features: the HSV color histogram and interest points extracted based on SIFT descriptors. The experimental evaluation validates the proposed approach, when using either interest points or HSV color histogram. The *F1* measure in both cases is around 86%.

We continued our analysis and proven that for a larger temporal length the proposed algorithm can be used to detect DVD chapters. The experimental results demonstrate our algorithm robustness, regardless to the movie type or gender, returning an average score of 76% for the *F1* norm.

Our perspectives of future work will concern the integration of our method within a more general framework of video indexing and retrieval applications, including object detection and recognition methodologies. Finally, we intend to integrate within our approach motion cues that can be useful for both reliable shot/scene/keyframe detection and event identification.

# 5. SALIENT OBJECT DETECTION AND TRACKING

**Summary:** In this chapter, we consider the issue of salient object detection. We start by presenting and analyzing the related work, and discuss for both spatial and temporal attention models proposed in the literature. Two categories of techniques are identified: bottom-up and top-down approaches. The last category is task driven, using prior knowledge of the video flow or its content and involves pattern, shape and other cognitive processing related features. In contrast, bottom-up algorithms exploit low level features (*e.g.* luminance, color, motion contrast…), in order to build a saliency map able to emphasize relevant regions.

In the second part, we propose a novel bottom-up approach for modeling the spatiotemporal attention in videos. The spatial model is developed starting from a region-based contrast measure associated to individual keyframes. The temporal model relies on interest points correspondence, geometric transforms (*i.e.* homographic motion model), motion classes estimation (using agglomerative clustering) and regions temporal consistency. Finally, the interest object is extracted with the help of GrabCut segmentation which takes as input to the saliency map previously determined.

**Keywords:** Saliency map, RANSAC algorithm, temporal attention model, homography transform, agglomerative clustering, *k-NN* algorithm, object segmentation, GrabCut, segmentation.

The human brain and visual system actively seek for regions of interest by paying more attention to some specific parts of the image/ video. The concept of visual saliency [Kim11] can be defined as the perceptual quality that allows an object, person or pixel to stand out from his neighbors by capturing our attention. Humans can easily understand a scene based on the selective visual attention which makes it possible to detect regions of interest in images or interesting actions in videos sequences [Achanta09].

In the field of computer vision, the objective is to simulate the human visual system behavior by automatically producing saliency maps of the target image or video sequence [Zhai06]. Salient object detection assume extracting the visual uniqueness, rarity, unpredictability or interesting events described as variations in image/video attributes such as color, texture, shape, edges and motion vectors. In this case, abnormal regions are quickly highlighted and can be further analyzed. However, some fundamental questions need to be solved: what part of the scene can be considered as salient? How can we define in a rigorous manner the concept of saliency?

One basic principle when considering the issue of modeling of the human visual system is to suppress the response to frequently occurring input patterns, while, in the same time, preserving sensitivity to novel features. Image/video information in this case can be regarded as a redundancy term plus a sparse contribution. Inspired by this insight, also from the perspective of cognitive science, the informational content, denoted by *Info(image/video*), can be decomposed into two components [Chandrasekaran09]:

$$Info(image|video) = \ Info(Redundancy) + Info(Saliency) \qquad (5.1)$$

where $Info(Redundancy)$ denotes the information with high regularities of the visual inputs and $Info(Saliency)$ represent the novel part. This principle is illustrated in Figure 5.1.

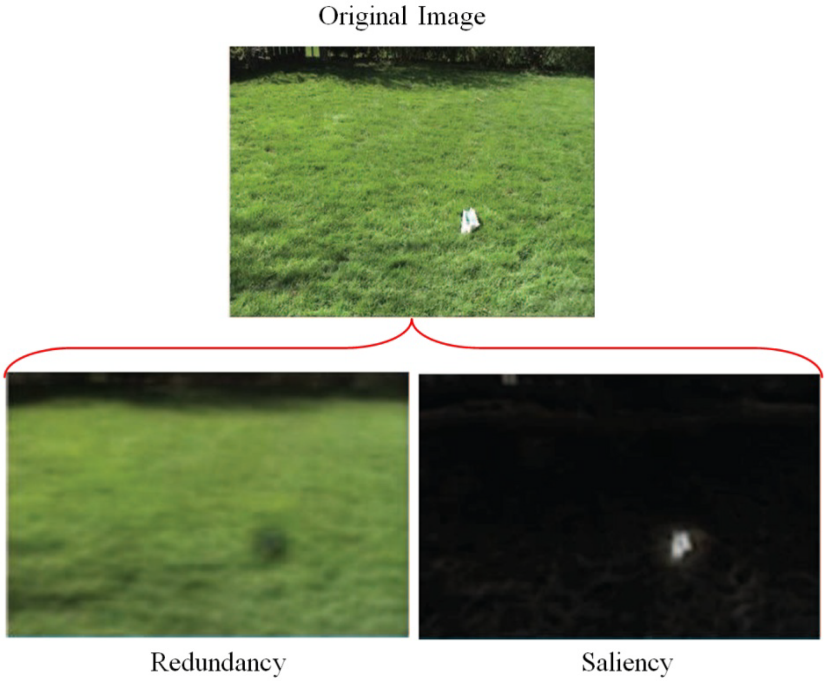


Figure 5.1. Decomposition of image information.

As it can be observed, the redundant information corresponds to statistical invariant features, while the salient characteristics always build on the properties assigned to a small number of objects, which correspond to a sparse component recovery problem.

Let us now review the various solutions proposed in the literature.

## 5.1. RELATED WORK

Recently, visual saliency has drawn great research interest in the fields of computer vision and of multimedia modeling. The image attention models can be divided into two families: bottom-up and top-down approaches. The last category is task driven, using prior knowledge of the video flow or its content and involves pattern, shape and other cognitive processing related features. Its major drawback is the lack of generality, since the same context is not available in every video document.

To solve this problem a various set of bottom-up approaches have been introduced [Li08], [Chen08], [Liu06] usually referred to as saliency detection or stimuli-driven techniques. Most of them model the human reaction to external stimuli, and exploit low level features (*e.g.* luminance, color, motion, contrast…), in order to build a saliency map which emphasizes relevant regions.

### 5.1.1. Spatial saliency detection

One of the first techniques proposed in the literature [Itti98] uses several feature attributes such as color, intensity and orientation.

The luminance image $I$ is used to create a Gaussian pyramid $I(\sigma)$, where $\sigma \in [0 \dots 8]$ is the scale. The center surround is defined as the difference between fine and coarse scales. The center is a pixel at scale $c \in \{2,3,4\}$ and the surround is the corresponding pixel at scale $s = c + \delta, \delta \in \{3,4\}$. The across-scale difference between two maps (denoted with $\Theta$) is obtained by interpolation to the finer scale and point by point subtraction.

The $r$, $g$ and $b$ channels are normalized by the luminance value $I$ in order to decouple hue from intensity. Four color channels are created: $R = r - \left(g + \frac{b}{2}\right)$, $B = b - \frac{r+g}{2}$, $G = g - \frac{r+b}{2}$ and $Y = \frac{r+g}{2} - \left|\frac{r-g}{2}\right| + b$. The first set of feature maps are concerned with intensity contrast computed in a set of six maps $\mathrm{M}(c,s)$:

$$M(c,s) = |I(c)\Theta I(s)| \qquad (5.2)$$

The second set of maps is similarly constructed for the color channels as follows:

$$RG(c,s) = \left|\big(R(c) - G(c)\big)\Theta\big(G(s) - R(s)\big)\right| \qquad (5.3)$$

$$BY(c,s) = \left|\big(B(c) - Y(c)\big)\Theta\big(Y(s) - B(s)\big)\right| \qquad (5.4)$$

Local orientation information is obtained from $I$ using the oriented Gabor pyramids $O(\sigma, \theta)$, where $\sigma \in [0 \dots 8]$represents the scale and $\theta \in \{0^{\circ}, 45^{\circ}, 90, 135^{\circ}\}$ is the preferred orientation. Orientation feature maps $\Phi(c, s, \theta)$, encode, as a group, local orientation contrast the between center and the surround scales:

$$\Phi(c, s, \theta) = |O(c, \theta) \Theta O(s, \theta)| \tag{5.5}$$

In total, 42 feature maps are computed: six for intensity, 12 for color and 24 for orientation.

The image saliency is defined based on the surrounded differences obtained across multi-scale image features. Authors propose a map normalization operator that can differentiate between scales in which only a small number of conspicuous locations are presented, while suppressing other maps with numerous comparable peaks responses. Only the local maxima of the various feature maps considered are here taken into account. The focus of attention is created as a simple disk, centered on each local maximum detected, with a fixed radius.

The proposed model has been shown to be successful in predicting human fixation and can be further enhanced to completely detect objects of interest. However, the approach requires the manual setting of numerous parameters.

A modified version of the technique introduced in [Itti98] is presented in [Harel07]. A graph-based technique is here considered in order to highlight the conspicuous parts of an image and to allow a combination with other importance maps.

Frintrop *et al*. introduced in [Frintrop07] a method inspired from Itti's [Itti98]. In contrast with the reference method, center surround differences obtained using a Gaussian image pyramid with 5 levels which are here computed in order to determine two types of features: (1) on-center differences, *i.e.* image parts responding strongly to bright regions on a dark background and (2) off-center differences, *i.e.* image parts responding strongly to dark regions on a bright background. The center surrounding mechanisms are computationally expensive. In order to decrease the processing time, authors propose to use integral images.

Another extension of the technique introduced in [Itti98] is proposed in [Hu04]. The visual saliency is here estimated by applying heuristic measures on the initial saliency maps obtained after thresholding the histogram of the features maps.

Based on the Itti's model where three feature maps are generated corresponding to color, intensity and orientation, the authors in [Hu04] propose a novel measure called Composite Saliency Indicator (CSI) to determine the contribution of each feature map to the salient region. Measures of spatial compactness and saliency density are here proposed in order to describe the neighborhood of each candidate salient point detected and to retain a reduced and consistent sub-set.

A technique based on the spectral residual is introduced in [Hou07]. For an input image the log Fourier spectrum $\mathcal{L}(I) = \log(A(I))$ is computed starting from a down-sampled image with height and width equals with 64 pixels (Figure 5.2).

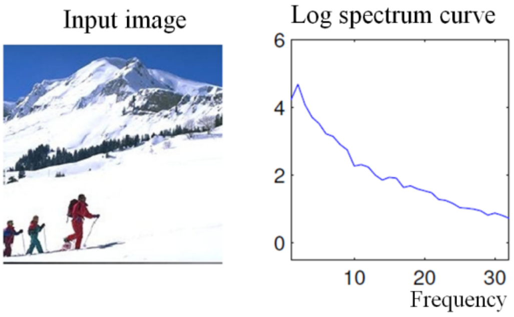


Figure 5.2. Orientation of average curves of log spectra.

The so-called spectral residual *R(I)* is computed as:

$$R(I) = \mathcal{L}(I) - A(I) \qquad . \tag{5.6}$$

Authors claim that the spectral residual *R(I)* makes it possible to detect the singularities of the considered image. However, the role of spectral residual in identifying salient regions is not clearly defined. Moreover the authors in [Guo10] suggest that the phase spectrum, instead of the amplitude spectrum of the image Fourier transform, can be more appropriate for determining the location of the salient areas.

A related approach is introduced in [Wang08], which exploits in a first stage the same spectral residual model presented in [Hou07] to quickly locate the visual pop-outs from the entire image. So, in this stage only coarse “unusual” regions are identified. In the second phase, a set of Gestalt features is exploited to propagate the results from the first step, based on a local coherence measure, to capture the object details.

In [Achanta08], authors propose to estimate the saliency using the center surrounding feature distance at various scales. This is evaluated as the distance between the average feature vectors of pixels within a sub-image with respect to the average feature vector of the pixels of its neighborhood sub-region. For each image, the saliency is determined at three different scales and the final map is computed as the sum of saliencies obtained at each resolution.

An extension to this technique is proposed in [Achanta09]. Here, authors propose a frequency-tuned method that directly defines pixel saliency using a pixel’s color difference from the average image color. In this case, the saliency map *S* of an image *I*, of width *W* and height *H* is defined as:

$$S(x,y) = \left\| I_{\mu} - I_{\omega_{hc}}(x,y) \right\| \tag{5.7}$$

where $I_{\mu}$ is the average image value, $I_{\omega_{hc}}(x,y)$ is the corresponding image value of the Gaussian blurred version of the original image, $\|\cdot\|$ is the $L_2$ norm, and $\omega_{hc}$ is the high frequency cut-off value. The proposed approach solely considers the first order statistics, which is not sufficient to analyze complex object variations encountered in natural images [Cheng11].

In bottom-up approaches the saliency of each location is a function of how distinct the considered location is from the surrounding background. This definition is satisfied by the ubiquity of “center-surround” mechanisms in the early stages of biological vision. Based on this observation, in [Gao07] authors propose to maximize the mutual information between the feature distributions of so-called center and surrounding image regions.

A different technique, called graph-based saliency that uses biological models, is introduced in [Harel07]. Here, the feature map is developed using the Itti’s method but the normalization is performed by employing a weighted graph-based approach, based on a Markovian modeling.

In [Guo10], the principles from human visual attention models are taken into consideration in order to construct the saliency model. First, low level cues are developed based on the supposition that a pixel is salient if its appearance is unique. However, in this case the isolated pixels need to be discarded and the analysis is performed on surrounding patches (obtained after dividing the image into blocks of 7x7 pixels). In the second step, a multi-scale analysis is performed in order to decrease the saliency of the background pixels. The background pixels are assumed to have similar patches at all scales while foreground pixels should differ significantly from one scale to another (Figure 5.3). In addition, the saliency map can be enhanced with a face detection [Viola01] algorithm.

By exploiting local contrast measures, the method tends to yield higher saliency values near edges instead of uniformly highlighting salient objects.

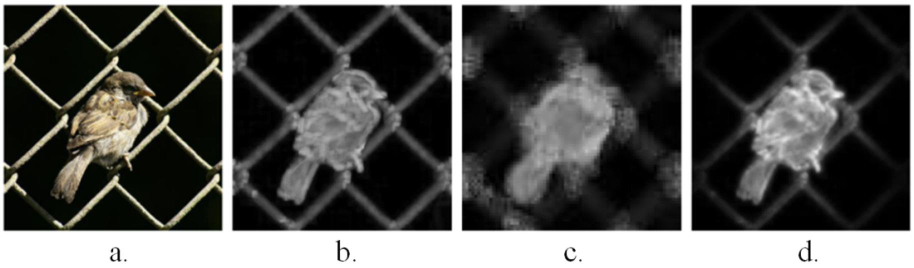


Figure 5.3. Saliency detection based on the technique proposed in [Guo10]: (a) Original image; (b) Saliency map at scale 1; (c) Saliency map at scale 4; (d) Final result.

Different studies [Liu06], [Gao08] attempt to model the visual saliency based on the relations existing between neighboring pixels.

Thus, Liu and Gleicher [Liu06] proposed a region-enhanced saliency detection strategy. The technique differentiates from other methods due to its scale invariance property and can be summarized into the following steps: First, the image is converted into L*u*v color space. Second the image is segmented and an adaptive Gaussian filtering pyramid is constructed.

Next, at each scale, a contrast pyramid is developed. The contrast value $c_{i,j,l}$ at scale $l$ is computed as:

$$c_{i,j,l} = \sum_{q \in \Theta} w_{i,j,l}\, d\left(p_{i,j,l}, p_q\right) \tag{5.8}$$

where Θ is the neighborhood of pixel *(i,j)* at scale *l*, $p_{i,j,l}$ the corresponding color value $p_q$ is the average color in the neighborhood, $d$ the $L_2$ distance, and $w_{i,j,l}$ weights designed in order to privilege as salient candidates the pixels located in the center of the image. Finally, the saliency map is constructed from the contrast pyramid by summing up all scales (Figure 5.4).

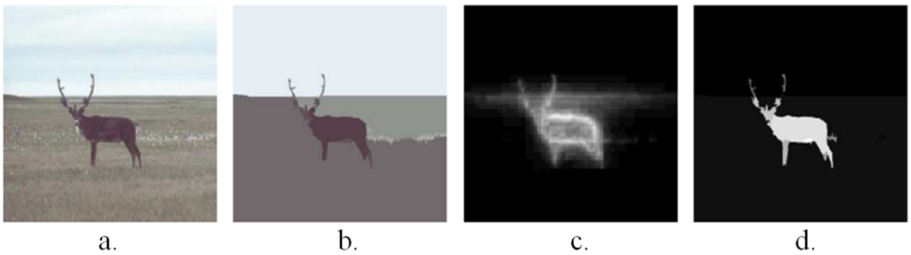


Figure 5.4. Region-enhanced saliency detection: (a) Original image; (b) Segmented image; (c) Scale invariant saliency, (d) Region enhanced saliency.

The technique introduced in [Muratov11] is based on image segmentation. The authors affirm that the major problem of previously developed saliency detectors is that only a small part of the interest object (*e.g*. edges or high contrast points) is detected. So, they propose to apply the saliency map on each segmented region by considering the relation between segments rather than pixels. Authors consider a set of global features including color information, luminance contrast and center surround histogram map. Based on the assumption that the object of interest is always located in the center of the image they computed two parameters corresponding to the segment location and size.

A three step method based on image patches is also introduced in [Duan11]. First, the image is segmented into non-overlapping patches for which only the color information is taken into consideration. Second, the patches are described in a low dimensional space which is obtained based on a method equivalent to PCA. In the final step, spatially weighted dissimilarities are evaluated based on a weighting mechanism that indicates the central bias.

The method presented in [Fang11] is also based on the principle of dividing the input image into small patches. For each patch the intensity Y and the color components R, G, B are considered in order to form two new color difference channels: $RG = R - G$ and $BY = B - Y$. By using Quaternion Fourier Transform (QFT) applied to each patch, the amplitude spectrum is determined as follows:

$$QR = A \cdot e^{i\varphi}, \text{ with } QR = QFT(Y, RG, BY) \quad . \tag{5.9}$$

The final saliency value of a patch is computed as the Euclidian distance between the amplitude spectrum of the image patch ($i$) and all its neighboring patches computed as:

$$SalVal(i,j) = \left| \sum_{m} \log\left(A_m^i + 1\right) - \sum_{n} \log\left(A_n^j + 1\right) \right| \quad . \tag{5.10}$$

The logarithm is used to reduce the dynamic range of the amplitude coefficients.

In [Gao08], authors define a discriminant center surround saliency, based on the idea that local image features are stimuli of interest when they are distinguishable from the background. An intensity map and four color channels are here used. The intensity channel is decomposed using a directional zero-mean Gabor filter at 3 spatial scales and 4 directions.

Most of the methods [Gao08], [Oliva03], [Zhang09] based on Gabor or DoG filter responses require many design parameters such as the number of filters, type of filters, choice of the nonlinearities, and a proper normalization scheme. Such methods tend to emphasize textured areas as being salient, regardless of their context.

A different technique is presented in [Seo09], where the authors introduce a nonparametric saliency estimation method. First, the detection is performed using the local regression kernels (LSK) as features. Such features capture the local structure of the data and are robust to significant noise distortion. Second, in order to determine the likelihood of saliency, called self-resemblance, they introduce a nonparametric kernel density estimation algorithm. The LSK features make it possible to accurately detect the salient objects. The main drawback is related to the high computational resources required.

Based on the definition of a semantic element, also called attention object (AO), (*e.g.*: a human face, a flower, an automobile, a text sentence), in [Chen03] authors propose assigning three attributes to each AO: the region of interest (ROI), the attention value (AV) and the minimum perceptible size (MPS). In order to determine the saliency map they adopt a three channel representation of the image based on color, intensity and orientation. In this case, the saliency attention value is computed as:

$$AV_{Saliency} = \sum_{(i,j \in R)} B_{i,j} \cdot W_{Saliency}^{i,j} \tag{5.11}$$

where $B_{i,j}$ denotes the brightness of pixel *(i, j)* in the saliency region $R$ and $W_{Saliency}^{i,j}$ is the positional weight associated to pixel *(i, j)*, defined with the help of a Gaussian template placed in the centre of the image (Figure 5.5).

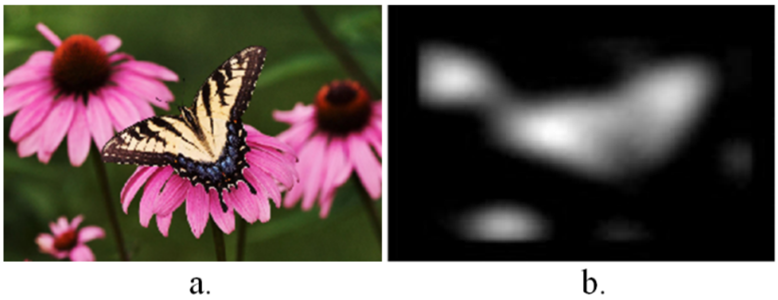


Figure 5.5. Image saliency detection using normalized Gaussian templates:
(a) Original image; (b) Saliency map.

The technique introduced in [Zhai06] is based on psychological studies and affirm that the human perception is sensitive to contrast of visual signals such as color, intensity and texture. Here, in order to determine the spatial attention model they exploit color features. The computational complexity is linear with respect to the total number of pixels in the image. The saliency map of an image is built upon the color contrast between image pixels. The saliency value of a pixel $I_k$ in an image $I$ is defined as:

$$SalS(I_k) = \sum_{\forall I_i \in I} \|I_k - I_i\| \tag{5.12}$$

where $\|\cdot\|$ represents the color distance metric.

A color quantization is applied in order to obtain a set of 256 indexed colors. Based on the obtained saliency map the authors propose a region growing technique for detecting the salient regions (Figure 5.6).

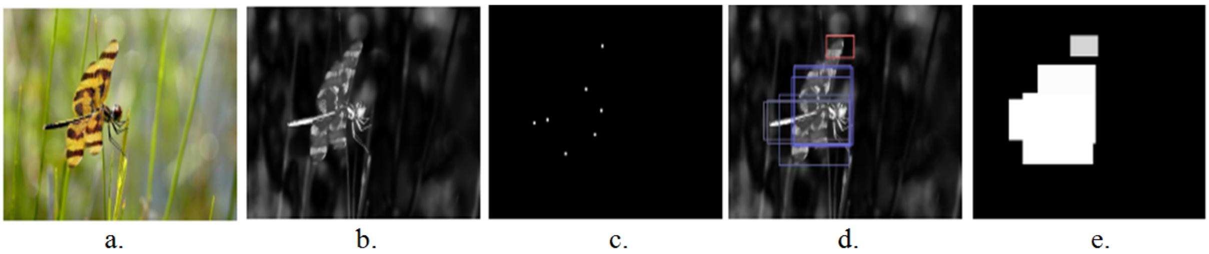


Figure 5.6. Spatial attention detection: (a) Input image; (b) Pixel level saliency map; (c) Detected attention points; (d) Expanded boxes from the attention points; (e) Saliency map.

In a general manner, global contrast based methods evaluate saliency of an image region using contrast with respect to the entire image.

A different approach is proposed in [Xie11] that use salient points to detect the corners of an object of interest. In order to provide the preliminary location for an attention region the convex hull techniques is used to enclose the detected salient points (Figure 5.7).

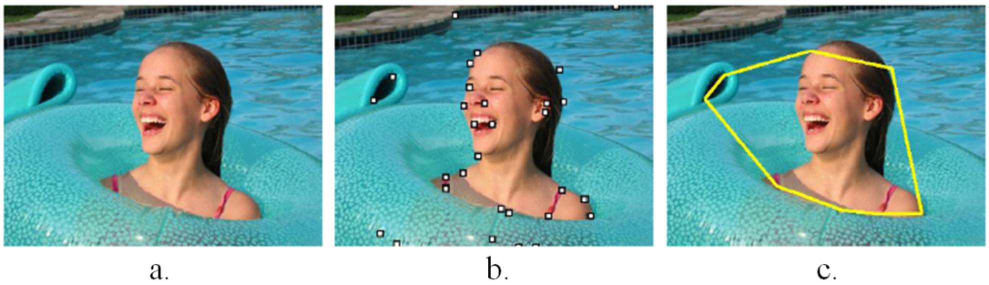


Figure 5.7. Rough saliency estimation as in [Xie11]: (a) Input image; (b) Salient points; (c) Convex hull.

After estimating the location of the rough salient regions the final saliency map is determined with the help of a Bayesian estimator.

Various studies that combine the bottom-up and top-down models for better detection performances. In most part of the cases, the top-down component is actually a human face detector [Itti01], [Suh03]. Some variation is presented in [Chen03] that combine a face and text detector to find the optimal solutions through a branch and bound algorithm.

The above presented approaches concern exclusively the case for 2D still images. Let us now analyze how the saliency detection is considered in the case of video data.

### 5.1.2. Temporal saliency detection

One of the greatest challenges in computer vision is automatic interpretation of dynamic scenes which include detection, localization and segmentation of objects and people. The video saliency detection represents a highly promising manner to understand and identify relevant features of interest and its content. While this can be performed by analyzing individual frames independently, video provides rich additional cues, which include motion of objects in the scene, temporal continuity, long range temporal object interaction and the causal relations among events [Lezama11].

Motion has a great influence in identifying the salient regions in complex, dynamic scenes. Salient motion models combined with bottom-up and top-down cues can lead to an efficient visual saliency model, which can be generated with the help of static saliency maps and motion vectors.

In [Zhai08], authors use both low level features and cognitive features, such as skin color and captions, to develop a visual attention model. They start by converting the image in the YCbCr color space. The orientation channel ($co_i$) is obtained by filtering the intensity channel ($ci_i$) in four directions with Gabor filters (GF($\theta$)):

$$co_i(\theta) = ci_i * GF(\theta), \theta \in \{0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\} \qquad (5.13)$$

The motion vectors are estimated with the help of the Ogale's algorithm [Ogale06]. Between three consecutive frames, the average directional optical flows $of(f_{i-1}, f_i)$ and $of(f_i, f_{i+1})$ are computed in both horizontal ($h$) and vertical ($v$) directions, as described in the following equation:

$$cm_i^{\Theta} = \frac{[of^{\Theta}(f_i, f_{i+1}) + of^{\Theta}(f_{i-1}, f_i)]}{2}, \Theta \in (h, v) \qquad (5.14)$$

For each frame, a pyramid is created by iteratively down-sampling the established channels. Finally, the various channels considered are combined in one single saliency profile.

In practice, the motion can be produced by two types of elements: salient (interesting) objects and background (uninteresting) objects. The salient motion is defined in [Ying-Li05] as the motion from a typical surveillance target (person of vehicle) that is opposed to other type of movements such as: camera displacements, swaying of vegetation in the wind… Based on this definition, authors propose a spatiotemporal saliency detection algorithm, dedicated to real-time surveillance videos, that is able to detect interesting objects characterized by a consistent motion over the time.

The proposed method can be summarized into the following steps. First, the region of change between successive frames is obtained by substraction. Next, the optical flow is computed with the help of the well-known Lucas-Kanade algorithm. In the third phase, pixels motion consistency over a set of successive frames in the horizontal/vertical directions is determined, by temporal filtering. Finally, a salient object is detected by combining the temporal difference images with the temporal filtered motions. This process is illustrated in Figure 5.8.

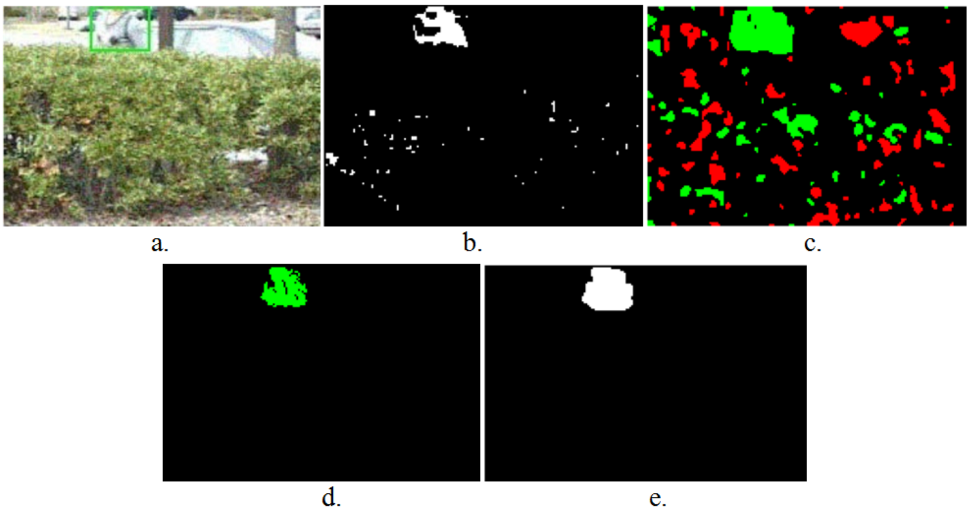


Figure 5.8. Salient motion detection [Ying-Li05]: (a) Initial frame; (b) Difference image between successive frames; (c) Horizontal optical flow; (d) Filtered optical flow; (e) Salient object.

The method introduced in [Chen08] extracts a set of salient feature points, from 3D spatiotemporal volumes of video sequences. Such feature points are further used as seeds in a region growing-based approach in order to detect the salient regions in a motion attention map. The salient feature points are determined based on a Harris detector that is constructed using a 3 x 3 second order matrix $\mu$ associated to each pixel in each frame and defined as:

$$\mu = g(x, y, t\colon \sigma_i^2, \tau_i^2) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}, \tag{5.15}$$

where $\sigma_i^2 = s \cdot \sigma_l^2$ and $\tau_i^2 = s \cdot \tau_l^2$ are the integration scales, $L_k$ is the first order Gaussian derivative through the $k$ axis and $g(x, y, t\colon \sigma_i^2, \tau_i^2)$ is a Gaussian weighting function. The pixels with significant eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ of matrix $\mu$ are considered as salient.

After detecting the seeds, a motion attention map is constructed in order to determine the spatial extent of the search window. The goal is to find salient regions that present consistent motions. The optical flow $(u, v, w)$ in the neighborhood of each pixel is estimated by solving the following structural tensor equation:

$$\mu \cdot [u\ v\ w]^T = 0_{3\times l}. \tag{5.16}$$

The continuous rank-deficiency measure $d_\mu$ is defined as:

$$d_\mu = \begin{cases} 0, & if\ rank(\mu) = 0 \\ \dfrac{\lambda_3^2}{(\dfrac{1}{2\lambda_1^2} + \dfrac{1}{2\lambda_2^2} + \varepsilon)}, otherwise \end{cases} \tag{5.17}$$

where $\varepsilon$ is a constant used to avoid the division to zero. The motion attention map is obtained after applying a median filter ($f_\mu$) on $d_\mu$ in order to keep regions with a consistent motion. Finally based on the motion attention map, the most appropriate scale for each region centered in the interest point is determined.

In [Sevilmis08], authors focus on extracting all objects existent in the video flow, based on the following set of heuristic principles:

- All objects (visually conspicuous or not) situated in the camera focus are salient;
- Any moving object is identified as representative;
- A face is always considered as salient because it indicates the presence of a person.

So, initially the smoothed (filtered to remove noise and to flatten textures) frames are segmented into regions using a graph-based technique. Then, the region saliency is computed based on the set of heuristic principles presented above. The relevant regions are then tracked trough the entire shot in order to construct a salient an object.

In [Hua09], a novel approach of distortion free video retargeting based on scale-space spatio-temporal tracking is introduced. The technique aims at transforming an existent video in order to display it on lower resolution target devices. The saliency map is firstly estimated using the residue method presented in [Hou07] that is further extended to incorporate scale-space information. If we consider a set of *n* consecutive image frames:

$$S_t^n(i,j,k) = \{I_{t-n+1}(i,j), I_{t-n+2}(i,j), \dots, I_t(i,j)\}, \tag{5.18}$$

with $k$ indexing the individual frames, the phase spectrum of the 3D Fourier transform associated to the sequence $S_t^n$ is first computed. Next, an inverse FFT is applied simultaneously with a set of smoothing operations (with various kernels) in the spatial domain. Finally, all information from each scale is combined in one single map. The different phases of the approach are illustrated in Figure 5.9.

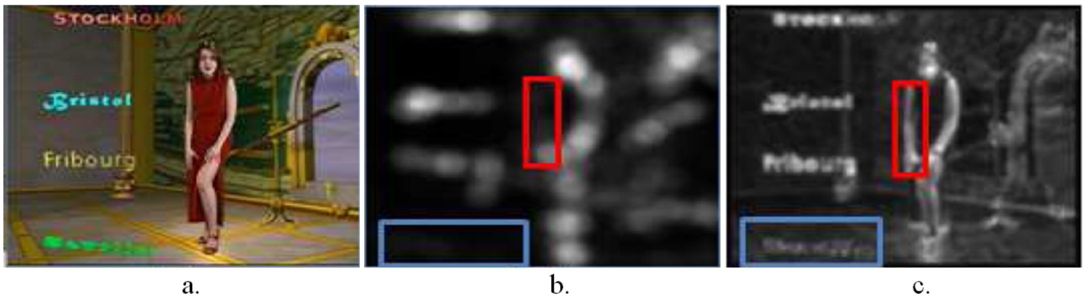


Figure 5.9. Saliency detection in videos based using [Hua09]: (a) Original image; (b) Spectrum residue; (c) Phase spectrum.

A related technique that uses the spatio-temporal saliency fusion for video retargeting is introduced in [Lu10]. The attention map in this case is based on a non-linear scheme that combines spatial and temporal information. The spatial saliency is detected by the phase spectrum of Quaternion Fourier Transform applied on each color frame, which uses multiple channels to exploit conspicuous spatial features (*e.g.* intensity, color, texture…). The temporal saliency is measured by the local motion residue, while the global motion parameters are estimated from the matched feature points using a robust affine fitting (Least Median of Squares). In this case, the technique uses a sparse optical flow estimated based on feature points.

Let us underline the nonlinear fusion principle employed here, which attempts to simulate several functions of the human perceptual system. First, in the absence of an excitation (*e.g.*, the case of a uniformly distributed texture), the attention is focused in the center of the frame and not on the borders. Second, in the case of two regions with similar spatial saliencies, a human will be focused on the region with the highest motion.

The visual saliency detection algorithm introduced in [Ma11] incorporates the motion trajectory in order to identify relevant objects. Each frame of a video flow is described using a quaternion representation (QR) which integrates the spatial image content, the motion trajectories and the temporal residuals. An overlapped block-based motion estimation (OBME) is applied in order to determine the temporal motion trajectory. After OBME, three temporal components are obtained: $MV_x(i,j)$, $MV_y(i,j)$ - the horizontal and vertical vector motion blocks centered in $(i,j)$ and $PE(i,j)$ - the motion prediction error. The final saliency map is obtained from the phase of the Fourier spectrum as described by the following equation:

$$SalV(x,y) = g(x,y) * \left\| F^{-1}(\exp(i \cdot p(x,y))) \right\|^2 \tag{5.19}$$

where $F$ and $F^{-1}$ denote the direct and inverse Fourier transforms, $p(x,y) = P(F(I(x,y)))$ is the phase spectrum, *I(x,y)* the original image and $g(x,y)$ is a Gaussian filter used for smoothing.

A multi-modal saliency detection method for audio-visual video streams is introduced in [Evangelopoulos08]. The audio saliency detection is based on signal modulation and related multi-frequency band features extracted through nonlinear operators and energy tracking. The audio signal is represented as a sum of narrowband amplitude and frequency varying, non-stationary sinusoids that are further demodulated in amplitude and frequency using a set of frequency-tuned Gabor filters. For each frame, the dominant modulation component is defined as the one that maximize the average Teager energy.

The spatio-temporal saliency map is obtained by decomposing the video into a set of features (intensity, color and orientation) that are further processed using three dimensional Gaussian filters ($G_{3D}^{\theta,\varphi}$) and their Hilbert transforms ($H_{3D}^{\theta,\varphi}$). A single spatio-temporal saliency volume is obtained based on a Principal Component Analysis.

A hybrid algorithm that includes stationary saliency models based on both top-down and bottom-up visual cues combined with motion information and prediction is introduced in [Guraya11]. The bottom-up saliency detection uses color, intensity and orientation visual feature as described in [Itti98]. Because the proposed model is applied on surveillance videos the authors also propose to integrate a face detector. The salient motion is determined based on the technique described in [Ying-Li05]. The improvement brought by the proposed method is given by the prediction algorithm that is able to compute the saliency map with the help of the previous saliency maps. So, the salient regions at frame *t+1* are predicted as a combination of the saliency information from frame *1* to *t*, updated with the motion vectors magnitude computed between frame *t* and *t+1* (Figure 5.10).

The motion saliency detection method proposed by Xue *et al* [Xue12] makes use of the low rank and sparse decomposition of video slices along X-T and Y-T planes in order to separate the foreground moving objects from the background. Each X-T and Y-T slice (S) can be decomposed as described in the following equation:

$$min\|B\|_* + \lambda\|M\|_1, \qquad \text{such that} \quad S = B + M \tag{5.20}$$

where the low-rank component B corresponds to the background, M captures the motion objects in foreground, $\lambda$ is a coefficient controlling the weight of the sparse matrix, $\|\cdot\|_*$ and $\|\cdot\|_1$respectively represent the nuclear [Jaggi10] and $L_1$ norms of the considered matrices. The slices are then integrated together using a normalization process in order to construct the final saliency map.

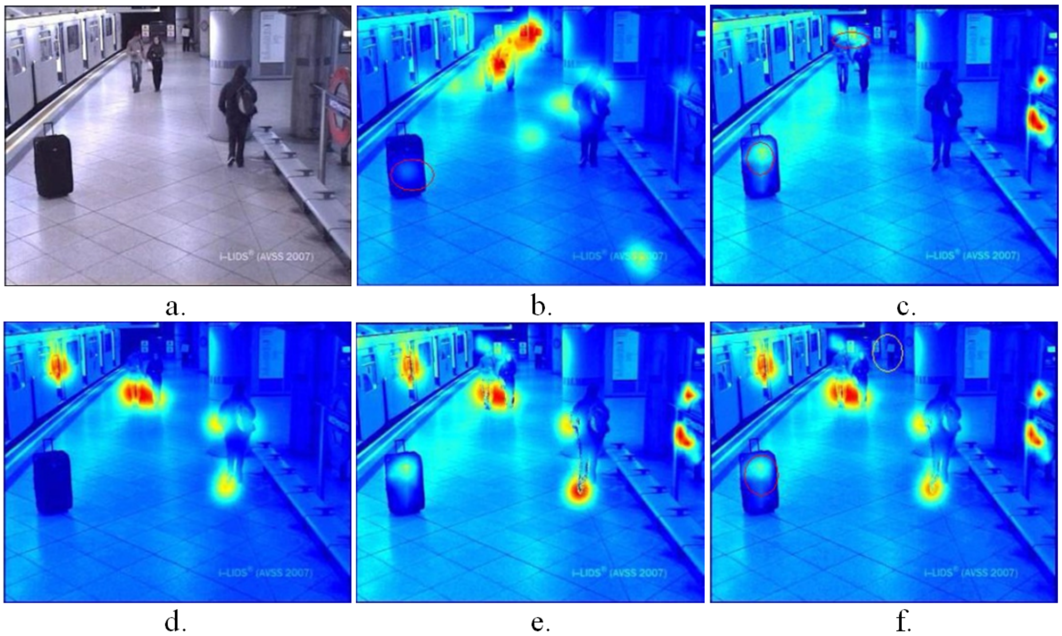


Figure 5.10. Saliency detection using predictive attention [Guraya11]: (a) Input frame; (b) Spatial saliency map; (c) Top-down attention; (d) Motion saliency map; (e) Predictive map; (f) Combined saliency map.

Whatever the features used for saliency detection, most of the previously presented methods generate an attention map without explicitly modeling the coherence of the results, either spatially or temporally. In [Wu11], authors claim that modeling such a coherence is a highly important issue that needs to be appropriately taken into account. In their approach, they consider: (1) the spatial coherence of low-level visual grouping cues (*e.g.* appearance and motion), which helps per-frame object-background separation, and (2) the temporal coherence of the object properties (*e.g.* shape and appearance), which ensures consistent object localization over time.

Here, the saliency map extraction is formulated as a binary map problem. An energy function of the binary map associated to each frame $E(A_t|I_{t,t-1}, A_{t-1})$ is defined as a linear combination of various terms, as described in the following equation:

$$E\left(A_t\middle|I_{t,t-1}, A_{t-1}\right) = S(A_t, I_t, M_t) + \alpha C_S(A_t, I_t, M_t) + \beta C_T(A_{t-1}, A_t, I_{t-1}, I_t)\ , \quad (5.21)$$

where $S(A_t, I_t, M_t)$ denotes the saliency map, $C_S(A_t, I_t, M_t)$ is the spatial coherence, $C_T(A_{t-1}, A_t, I_{t-1}, I_t)$ is the temporal coherence, $A_t$ is the salient object represented as a binary mask for each frame $I_t$, $M_t$ is the optical flow field computed from frame $I_{t-1}$ to $I_t$ while α and β are real-valued, positive weights.

The global energy defined in Equation (5.21) makes it possible to integrate both static and dynamic information within a unified framework.

Finally, let us mention the emerging saliency detection methods dedicated to the case of 3D, stereoscopic videos. An additional feature can be here exploited, which corresponds to the depth maps that can be constructed from such data. The depth information significantly affects the human perception and it is mandatory to consider it when developing the saliency map. To our very best knowledge, the only work existent in the technical literature addressing the problem of 3D attention models in presented in [Zhang10].

The authors develop a stereoscopic visual attention model with three attributes: depth information (*D*), spatial saliency map ($S_S$) and motion saliency ($S_m$):

$$S_{SVA} = \{D, S_S, S_m\} \quad (5.22)$$

The spatial saliency map is obtained by applying the method described in [Itti01]. The motion estimation is done using a block-based optical flow algorithm applied between successive frames. Regarding the depth map analysis, authors identify several major differences between 3D video and traditional movies:

- In the case of 3D videos the most interesting part are given by the regions that pop-up of the screen;
- As depth increases the amount of interest of a specific object is reducing;
- The objects situated out of the depth field of the camera system are usually not in the attention area.

Based on this assumption, the authors propose a depth-based dynamic fusion model that is able to combine all the three sources of information.

The analysis of the literature shows that for a successful detection of the salient objects, it is of outmost importance to integrate various static and dynamic visual features within a unified framework. In order to achieve this goal, we have elaborated a novel bottom-up, data driven saliency detection technique that incorporates spatial, temporal and eventually 3D information (in the case of stereoscopic data). The next section describes in details the proposed saliency detection approach, in the case of 2D videos.

## 5.2. PROPOSED SALIENT OBJECT DETECTION APPROACH

Figure 5.11 illustrates the proposed analysis framework, with the main phases involved. First, the video is temporally segmented into shots. For each determined shot, a set of representative key-frames is selected. Then, for each keyframe the salient regions are obtained, by combining the spatial and motion information in a dynamic fusion model. The selected regions are then applied as input to the GrabCut [Rother04] algorithm in order to extract the salient object.

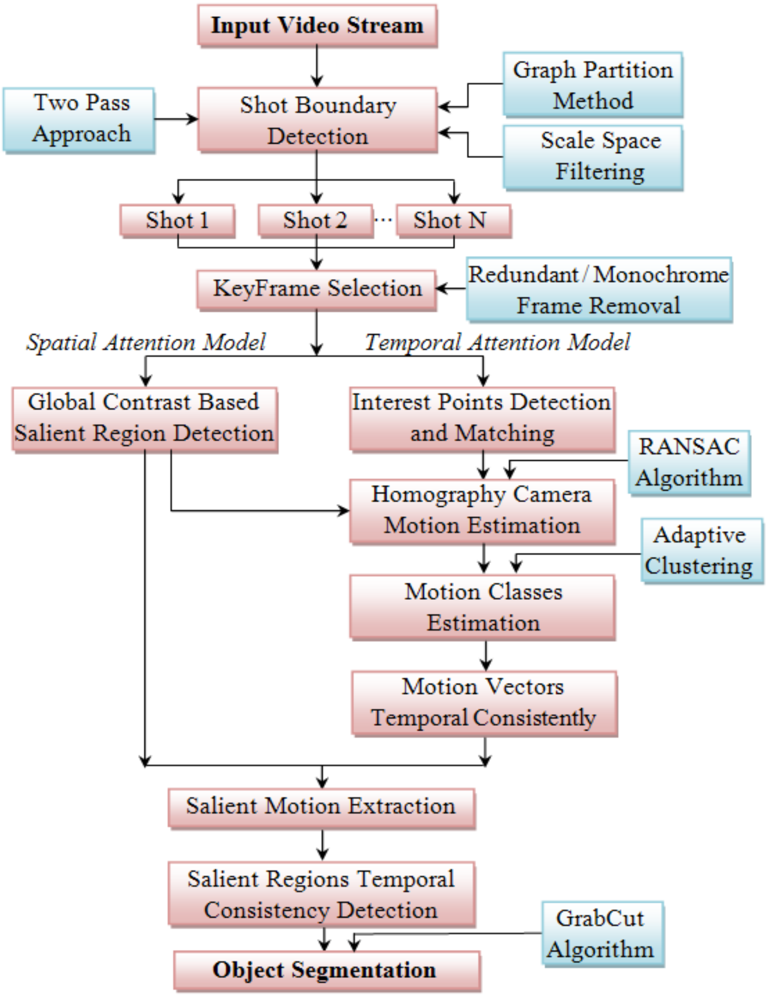

Input Video Stream
Graph Partition Method
Two Pass Approach
Shot Boundary Detection
Scale Space Filtering
Shot 1
Shot 2
...
Shot N
KeyFrame Selection
Redundant / Monochrome Frame Removal
Spatial Attention Model
Temporal Attention Model
Global Contrast Based Salient Region Detection
Interest Points Detection and Matching
RANSAC Algorithm
Homography Camera Motion Estimation
Adaptive Clustering
Motion Classes Estimation
Motion Vectors Temporal Consistently
Salient Motion Extraction
Salient Regions Temporal Consistency Detection
GrabCut Algorithm
Object Segmentation

Figure 5.11. Spatiotemporal salient object detection framework.

The spatial and temporal attention models proposed and considered are detailed in the following sections.

### 5.2.1. Spatial attention model

Current methods of saliency detection generate regions that have low resolution, poorly defined borders or are expensive to compute. Additionally, some methods produce high saliency values at the level of object edges instead of generating maps that uniformly cover the whole objects. As a consequence, such methods fail to exploit all the spatial frequency content of the considered images.

The spatial saliency model introduced in this section aims at overcoming such shortcomings and is based on an enhanced stationary saliency technique, so-called region-based contrast (RC) [Cheng11]. The goal here is to separate large scale objects from the background. A segmentation-based principle is applied. Thus, each keyframe is over-segmented into regions using the classic Mean Shift algorithm [Comaniciu02]. Let us note that other generic segmentation algorithm can also be used in this stage as graph partition strategy [Felzenszwalb04], expectation-maximization technique [Carson02], contour and texture analysis [Malik01]...

The Mean Shift segmentation is a local homogenization technique based on a clustering process that is very useful to eliminate shading or tonality differences in localized objects (Figure 5.12).

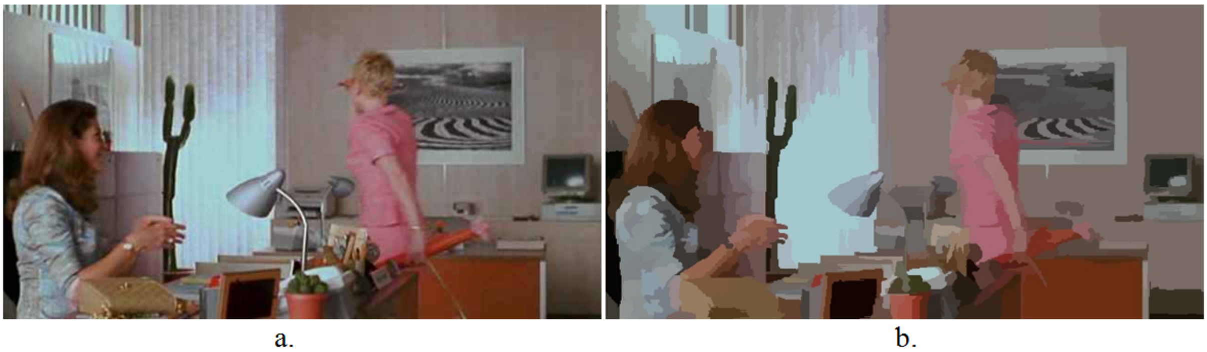


Figure 5.12. Image segmentation using the Mean Shift algorithm:
(a) Original image; (b) Segmented image.

The technique replaces each pixel value with the mean of the neighborhood pixels in a range (*r*) and whose values are within a color distance (*d*). Different parameters need to be specified:

- A distance function that determines the difference between pixels. In our case we used the Euclidian distance but any other function (*e.g.* Manhattan distance) can also be employed.
- A spatial window radius that gives the number of pixels accounted for computation. In our case the radius was fixed at 10 pixels;

- The colour window radius, that constrains the color magnitude, used to select for mean computation only the pixels having the value below the threshold. In our case the value was fixed at 10.

The spatial saliency value $S$ of each region $(r_k)$ obtained after segmentation is defined by measuring the color contrast of the considered segment with respect to all the other regions present in the image, as described in the following equation:

$$S(r_k) = \sum_{r \neq r_k} w(r_i) \cdot d_r(r_k, r_i) \ , \tag{5.23}$$

where $w(r_i)$ is the weight of region $r_i$, computed as the total number of pixels included in the region while $d_r(,)$ is the quadratic color distance metric between regions defined as:

$$d_r(r_1, r_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} p(c_{1,i}) \cdot p(c_{2,j}) \cdot \delta\big(c_{1,i}, c_{2,j}\ \big) \ , \tag{5.24}$$

where $p(c_{k,i})$ is the probability (estimated as the relative frequency of occurrence) of the $i$-th color $c_{k,i}$ among all $n_k$ colors in the $k$ region ($k = 1{,}2$). Note that the color space was initially quantized into $12^3$ different colors, each color channel being uniformly divided into 12 levels. While the quantization of colors is performed in the RGB color space, we measure the color difference in the L*a*b color space, which is perceptually more pertinent. Here, $\delta$ denotes the $L_1$ distance in the L*a*b color space.

For each region, in order to increase the influence of closer, neighboring regions and in the same time decrease the impact of regions located at farther locations (Figure 5.13) a spatial weighting term is introduced, as described in Equation (5.25).

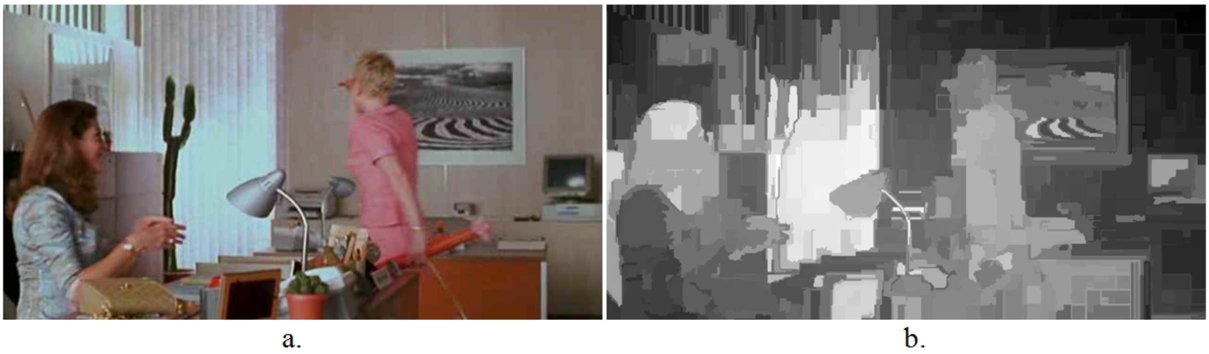


Figure 5.13. Spatial attention map extraction. (a) Original image; (b) Saliency map.

$$S(r_k) = \sum_{r \neq r_k} \exp\left(\frac{d_s(r_k, r_i)}{\sigma^2}\right) w(r_i) \cdot d_r(r_k, r_i). \tag{5.25}$$

Here, $d_s(r_k, r_i)$ is the spatial distance between the gravity centers of regions $r_k$ and $r_i$, while $\sigma^2$ is a parameter controlling the strength of the spatial weighting mechanism.

The spatial saliency is determined for each detected keyframe independently. In addition, a temporal saliency term is defined, in order to take into account the dynamic structure of the videos. This term is explicated in the next section.

### 5.2.2. Temporal attention model

The salient motion in a video can be intuitively interpreted as the movement that attracts the attention of a human subject. Most of the previously developed methods [Ying-Li05], [Belardinelli09] are based only on the temporal difference of adjacent frames and cannot effectively identify the salient motion.

We introduce a novel temporal attention technique that combines the previously described spatial (stationary) saliency model with a set of interest points ($I_p$) that are matched between successive video key-frames. In order to model the motion of moving regions, homographic transforms are used. More precisely, the algorithm consists of the following steps:

***Step 1****: Interest point detection and matching* – The Scale Invariant Feature Transform (SIFT) [Lowe04] is applied on two successive frames (by taking as starting frame each detected keyframe). The correspondence between the interest points is established using KD-tree matching technique [Panigrahy08].

For untexured regions, or for low resolution videos the SIFT descriptor is not able to detect interest points. So, in order to have a complete structure of key-points, for any part of a frame, we propose to extract at least one point for any segmented region. Each region of each detected keyframe is put into correspondence with regions from the successive frame, with the help of the template matching technique introduced in [Briechle01].

Let $I(x, y)$ denote the intensity value of the considered image at point $(x, y)$ and $N_x$, $N_y$ the image dimensions. The region is considered as a template $t$ of size $M_x \times M_y$, with $M_x$ and $M_y$ being the sizes of the region's bounding box. The position $(u_{pos}, v_{pos})$ of the pattern in the successive frame is given by the normalized cross correlation value $\gamma$ at each point $(u, v)$ of $I$ and the template $t$ (which has been shifted by $u$ steps in the $x$ direction and by $v$ steps in the $y$ direction). This procedure is illustrated in Figure. 5.14.

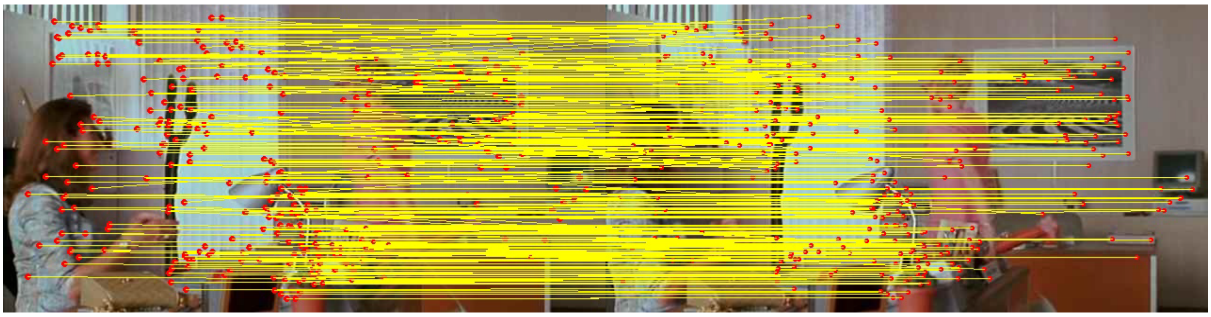


Figure 5.14. The interest points correspondences between successive frames.

The normalized cross correlation coefficient is defined as:

$$\gamma(u,v)=\frac{\sum_{x,y}(I(x,y)-\bar{I}_{u,v})(t(x-u,y-v)-\bar{t})}{\sqrt{\sum_{x,y}\left(I(x,y)-\bar{I}_{u,v}\right)^2\sum_{x,y}(t(x-u,y-v)-\bar{t})^2}} \tag{5.26}$$

where $\bar{I}_{u,v}$ denotes the mean value of $I(x,y)$ within the area of the template *t* shifted to (*u*, *v*) and it computes as:

$$\bar{I}_{u,v}=\frac{1}{M_xM_y}\sum_{x=u}^{u+M_x-1}\sum_{y=v}^{v+M_y-1}I(x,y) \tag{5.27}$$

and $\bar{t}$ is the mean value of template *t*. Due to the normalization obtained by the mean value substraction, the correlation coefficient $\gamma(u,v)$ is invariant to changes in brightness or contrast of the image. The desired position $\left(u_{pos},v_{pos}\right)$ of the pattern is equivalent to determining the maximum value of the $\gamma$ parameter.

For each matched segment an interest point is associated to, located in the gravity center of the considered segment.

Let $p_{1i}(x_{1i},y_{1i})$ be the *i*-th key point in the first image and $p_{2i}(x_{2i},y_{2i})$ be its correspondent in the second image. The associated motion vectors ($v_{ix},v_{iy}$), expressed in polar coordinates with magnitude ($D_{i(1,2)}$) and angle of motion ($\theta_{i(1,2)}$) are also computed in this step:

$$v_{ix}=x_{2i}-\ x_{1i}\ ;v_{iy}=y_{2i}-\ y_{1i}, \tag{5.28}$$

$$D_{i(1,2)}=\ \sqrt{v_{ix}^2+v_{iy}^2}\ ,\ i=\overline{1,n}, \tag{5.29}$$

$$\theta_{i(1,2)}=acos\frac{v_{ix}}{D_{i(1,2)}},\theta\in[0,2\pi] \tag{5.30}$$

where *n* is the total number of correspondences.

***Step 2*****:** *Interest points saliency initialization* – For the current keyframe, the interest point’s spatial saliency values are determined based on the technique described in Section 5.2.1. The saliency value associated to each interest point is defined as the saliency of the region it belongs to.

***Step 3****: Background / Camera motion detection* –We start our analysis by identifying a subset of *m* keypoints located in the background (Figure 5.15). An interest point $p_{1,i}$ is defined as a background point if:

$$Sal\left(p_{1,i}\right)\le T_h\ , \tag{5.31}$$

where $Sal\left(p_{1,i}\right)$ is the saliency value of point $p_{1,i}$ while $T_h$ is the average saliency value over the considered keyframe.

The subset of *m* background interest points is used to determine the global geometric transform between the selected images, by considering a homographic motion model. A mapping from $IP^2 \rightarrow IP^2$ is a projectivity if and only if there exists a non-singular 3 x 3 matrix $\boldsymbol{H}$ such that for any point $\boldsymbol{x}$ in $RP^2$ is mapped into $\boldsymbol{H} \cdot \boldsymbol{x}$. Based on the $m$ set of points and their correspondence we determine, by applying the RANSAC (*Random Sample Consensus*) [Lee07] algorithm, the optimal homographic matrix $\boldsymbol{H}$.

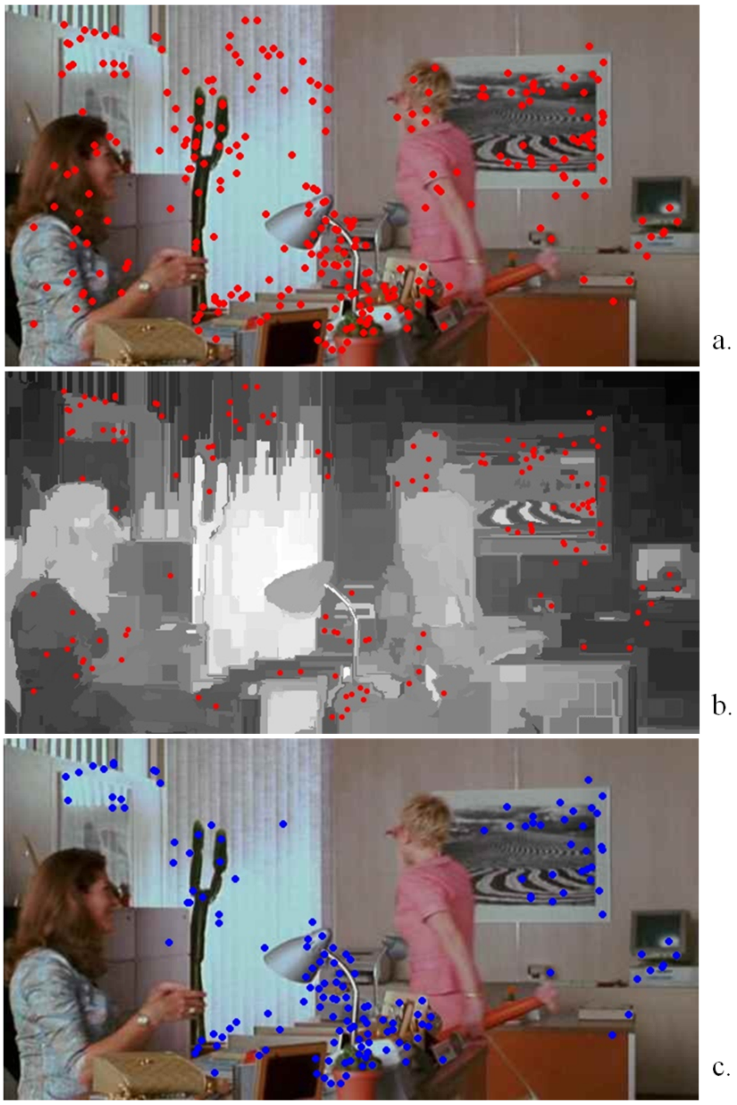


Figure 5.15. Camera motion estimation: (a) Initial interest points; (b) Subset of keypoints used for camera/background motion estimation; (c) Interest points belonging to camera / background movement.

The RANSAC technique can be summarized as follows. Starting from a random sample of 4 interest point correspondences, a homographic matrix $\boldsymbol{H}$ is computed. Then, each other pair of points is classified as an inlier or outlier depending of its concurrence with $\boldsymbol{H}$. After all of the interest points are considered for the estimation of matrix $\boldsymbol{H}$, the iteration that yields the largest number of inliers is selected.

Based on the matrix $\boldsymbol{H}$**,** for a current point $p_{1i} = [x_{1i}, y_{1i}, 1]^T$ expressed in homogeneous coordinates, its estimated correspondence position $p_{2i}^{est} = [x_{2i}^{est}, y_{2i}^{est}, 1]^T$ is determined as:

$$\begin{bmatrix} x_{2i}^{est} \\ y_{2i}^{est} \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \cdot \begin{bmatrix} x_{1i} \\ y_{1i} \\ 1 \end{bmatrix}, \tag{5.32}$$

where:

$$w = 1/(h_{20} \cdot x_{2i}^{est} + h_{21} \cdot y_{2i}^{est} + h_{22}). \tag{5.33}$$

The estimation error is defined as the difference between estimated and actual position of the considered interest point, as described in Equation (5.34):

$$\epsilon(p_{1i}, \boldsymbol{H}) = \| p_{2i}^{est} - p_{2i} \| . \tag{5.34}$$

Ideally $p_{2i}^{est} = [x_{2i}^{est}, y_{2i}^{est}, 1]^T$ should be as close as possible to $p_{2i} = [x_{2i}, y_{2i}, 1]^T$.

In the case where the estimation error $\epsilon(p_{1i}, \boldsymbol{H})$ is inferior to a predefined threshold *E*, the corresponding pixels are marked as belonging to background. The outliers, *i.e.* pixels with estimation error $\epsilon(p_{1i}, \boldsymbol{H})$ exceeding the considered threshold, are considered to belong to foreground objects.

In our experiments, the background/foreground separation threshold *E* has been set to 5 pixels.

***Step 4:*** *Estimation of motion classes* - In practice, multiple moving objects can be present in the scene. In this case, we determine a new subset of points formed by all the outliers and all the points not considered in previous step (obtained after subtracting from all the interest points the subset of *m* background points) (Figure 5.16).

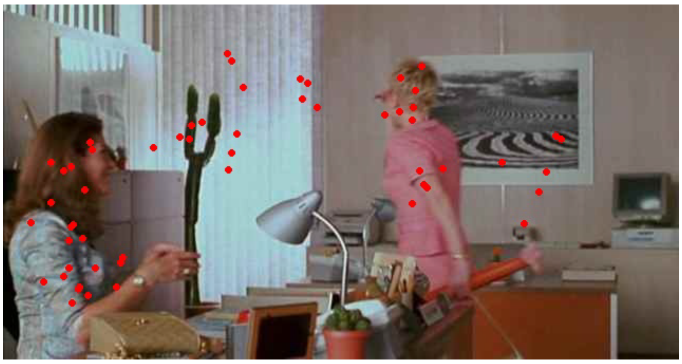


Figure 5.16. Interest points not assigned to camera / background motion.

Then the considered points are agglomeratively clustered into classes. The basic principle behind agglomerative clustering techniques is to consider each individual point as a cluster and then successively reduce the number of classes by merging the two closest clusters until all points are assigned to a category [Cimiano04]. The key operation of the proposed algorithm is the proximity computation between two interest points that are classified into clusters based on the following steps:

- *Phase I* - The motion vectors are sorted in descending order based on the number of occurrences of the motion vector angle. For the first interest point considered in the current list, a new cluster is formed ($MC_i$) having as centroid its motion vector angular value ($\theta_c$);

- *Phase II* - For all the other interest points, not assigned to any motion class, we compute the angular deviation.

$$Dev(\theta_i, \theta_c) = |\theta_i - \theta_c| \quad (5.35)$$

If $Dev(\theta_i, \theta_c)$ is beyond a predefined threshold $Th_\theta$ (set to 30 degrees in our experiments) and the motion magnitude is equal with the cluster centroid then the current point will be grouped into $MC_i$ cluster. Otherwise, a new motion class is created.

For the remaining outliers, the process is applied recursively until all points belong to a motion class (Figure 5.17).

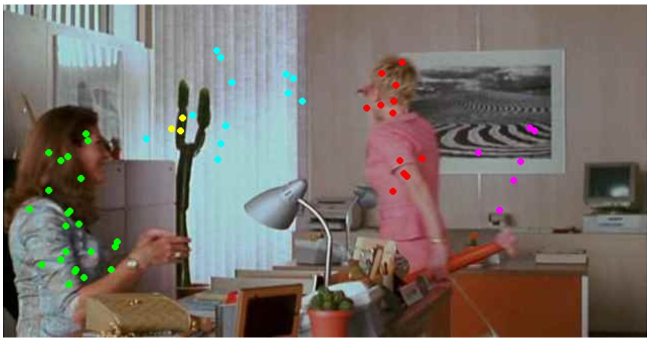


Figure 5.17. Motion classes' estimation.

***Step 5**: Motion vector temporal consistency* – In order to filter out the noisy vectors from the motion classes, which correspond to miss-matches of the considered interest points, we also compute the motion vectors with respect to the previous frame.

Based on the new angular values and magnitudes we determined the novel cluster centroid as described in *Step 4.I.* With the new centroids we determine for each point the novel angular deviation. If the conditions imposed in *Step 4* are not satisfied the interest point is excluded from the motion class and is automatically assigned to the background.

This process is based on the hypothesis that the motion should be consistent for at least 3 successive frames and makes it possible to reduce the number of outliers (Figure 5.18).

***Step 6**: Interest point refinement* – For all the interest points included in motion classes we applied next the *k-NN* algorithm [Zhang05] in order to verify that their assignment to the current class is not caused by an erroneous clustering. For the current point we determine its *k* nearest neighbors based on Euclidian distance (Figure 5.19). If at least half of the detected points do not belong to the same motion class then this point is eliminated from the motion cluster. In this case, its classification to the current group is considered erroneous and it is probably caused by template mismatching or instabilities in the homography estimation process.

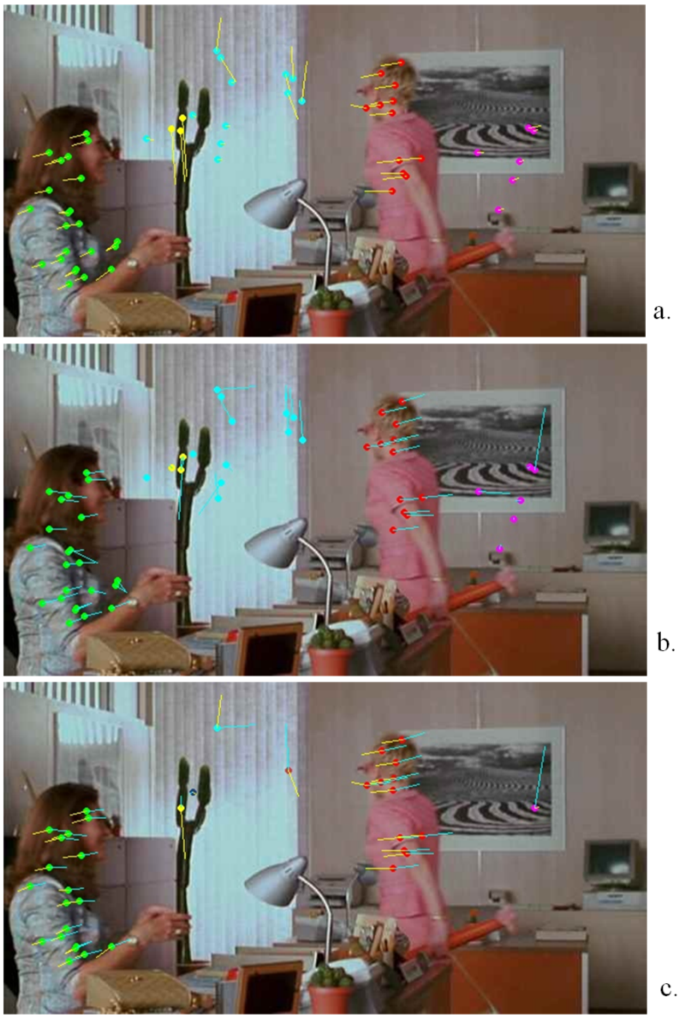


Figure 5.18. Motion vectors computed between.(a) Successive frames;
(b) Predecessor frames; (c) Motion classes correction.

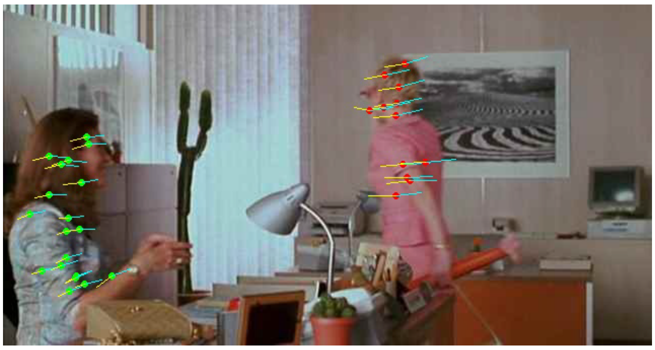


Figure 5.19. Refined salient motion classes

In our experiments parameter *k* has been set to 5. After all interest points are verified and correctly clustered into motion classes, we determine the salient movement based on the associated spatial saliency map, as described here below in Step 7.

***Step 7***: *Salient motion detection* – For all the motion classes determined at *Step 5* we compute their saliency values as:

$$SalClass(M_i) = \frac{\sum_{j=1}^{m_i} Sal(p_{1,i})}{m_i}, i = \overline{1, N}, \tag{5.36}$$

where $m_i$ is the total number of points included in motion class $M_i$, $N$ is the total number of classes and $Sal(p_{1,i})$ is the value of an interest point $p_{1,i}$ in the spatial saliency map. In this case, the salient motion is determined as:

$$SalientMotion = \max_{i=1,N}\{SalClass(M_i)\} \tag{5.37}$$

***Step 8****: Salient region detection* – Using the interest points included in the salient motion class we determine next the corresponding salient regions. We have superimposed the salient interest points over the segmented image (Figure 5.12). A region is considered as salient if its associated interest point belongs to the salient motion class.

In Figure 5.20 it is presented the temporal saliency map obtained. The regions marked with red contain at least one interest point belonging to the salient motion class while all other regions (without salient points) are marked with black.

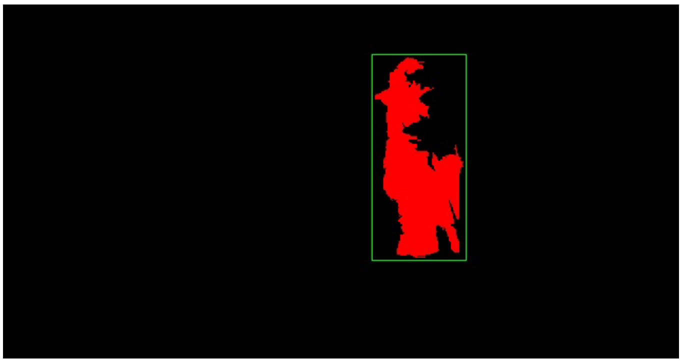


Figure 5.20. Salient regions detection based on interest points.

***Step 9:*** *Object temporal consistency* – In order to enhance the robustness and to reduce the computational time of our salient object detection method, we check in this stage for the temporal consistency of the detected regions. We assume that the salient region should generally be a smooth function in time, except for discontinuities at object borders or occlusions (in which case the region area will drastically change).

We propose to consider the sequence of salient regions as a three-dimensional, binary function $r_{crt}$ *(x, y, t)* with *(x, y)* being the spatial coordinates and *t* being the temporal coordinate. In this step we search for regions, between successive frames, that preserve the object area ($O_A$) as much as possible.

The detected object area region ($O_A$) is tracked during an interval of $T$ successive frames, with the help of a template matching technique if the following condition is satisfied:

$$O_A(f_{crt}) \bigcap O_A(f_{ant}) \geq 0.8 \cdot O_A(f_{crt}) \tag{5.38}$$

where $O_A(f_{crt})$ denotes the object area in the current frame, $O_A(f_{ant})$ represents the area of the salient object in the previous frame (Figure 5.21). The result of the tracking provides the regions' object support over time. In the case where the tracking is successful (in the sense of Equation (5.38)), the steps 1 to 8 are no longer applied to the corresponding frames, which makes it possible to speed up the detection process.

If the condition presented in Equation (5.38) is not satisfied a new object area is constructed using the salient regions from the current frame. In this case the tacking is no longer performed and for the following frames the algorithm will begin with the first step. In our experiments, we have considered a value of 15 frames for parameter *T*.

The spatial and temporal attention models are now combined in order to produce the final video saliency map. In scenes with no independently moving objects, the system detects a single motion class, corresponding to the camera motion. In this case, the segmentation is performed based solely on the spatial attention model.

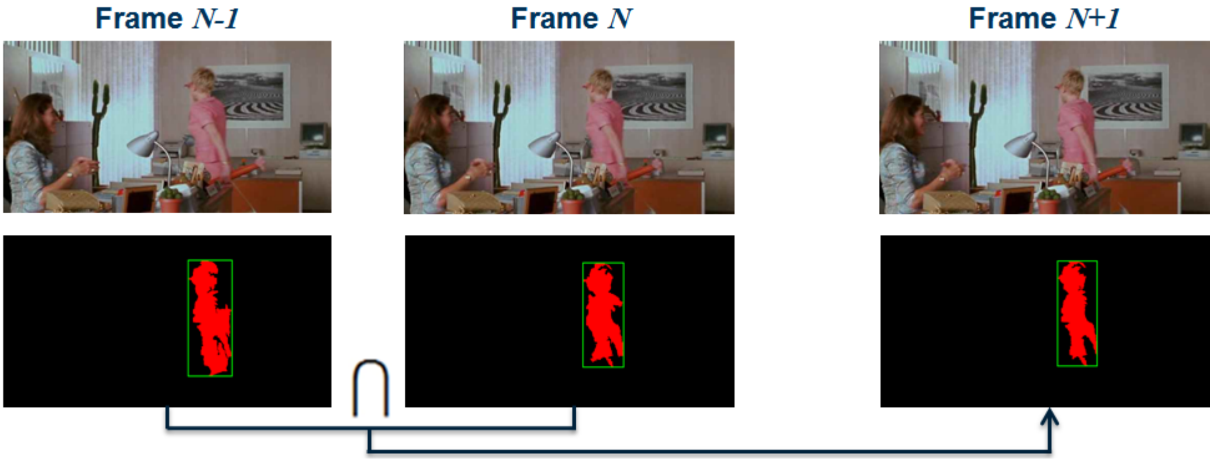


Figure 5.21. Area consistency between successive frames.

***Step 10:*** *Object detection* – The object is extracted with the help of the GrabCut algorithm (Figure 5.22) [Rother04], which is automatically initialized with a ternary saliency map obtained after *step 8*. More precisely, the pixels belonging to the salient region are labeled as certainly foreground, the regions outside the green rectangle (Figure 5.21) are marked as certain background, while the pixels inside the green rectangle not belonging to the salient regions are marked as probably foreground.

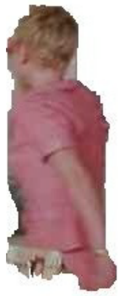


Figure 5.22. Salient object detection.

### 5.2.3. Experimental evaluation

We tested the proposed methodology on a set of 20 general purpose videos. The video database is composed from the following subsets:

- *Set1*: 8 videos from the TRECVid 2001/2002 evaluation campaigns (which are freely available on Internet (www.archive.org and www.open-video.org));
- *Set 2*: 6 videos proposed in [Fukuchi09], for which the authors provide also the ground truth, (available at www.brl.ntt.co.jp/people/akisato/saliency3.html);
- *Set 3:* 6 videos selected from [Lezama11] a database used to evaluate the performance of video segmentation algorithms, (available at www.di.ens.fr/willow/research/video).

Some samples of the videos considered for evaluation and illustrated in this chapter, together with the corresponding ground truth salient object, are presented in Figures 5.23, 5.24 and 5.25

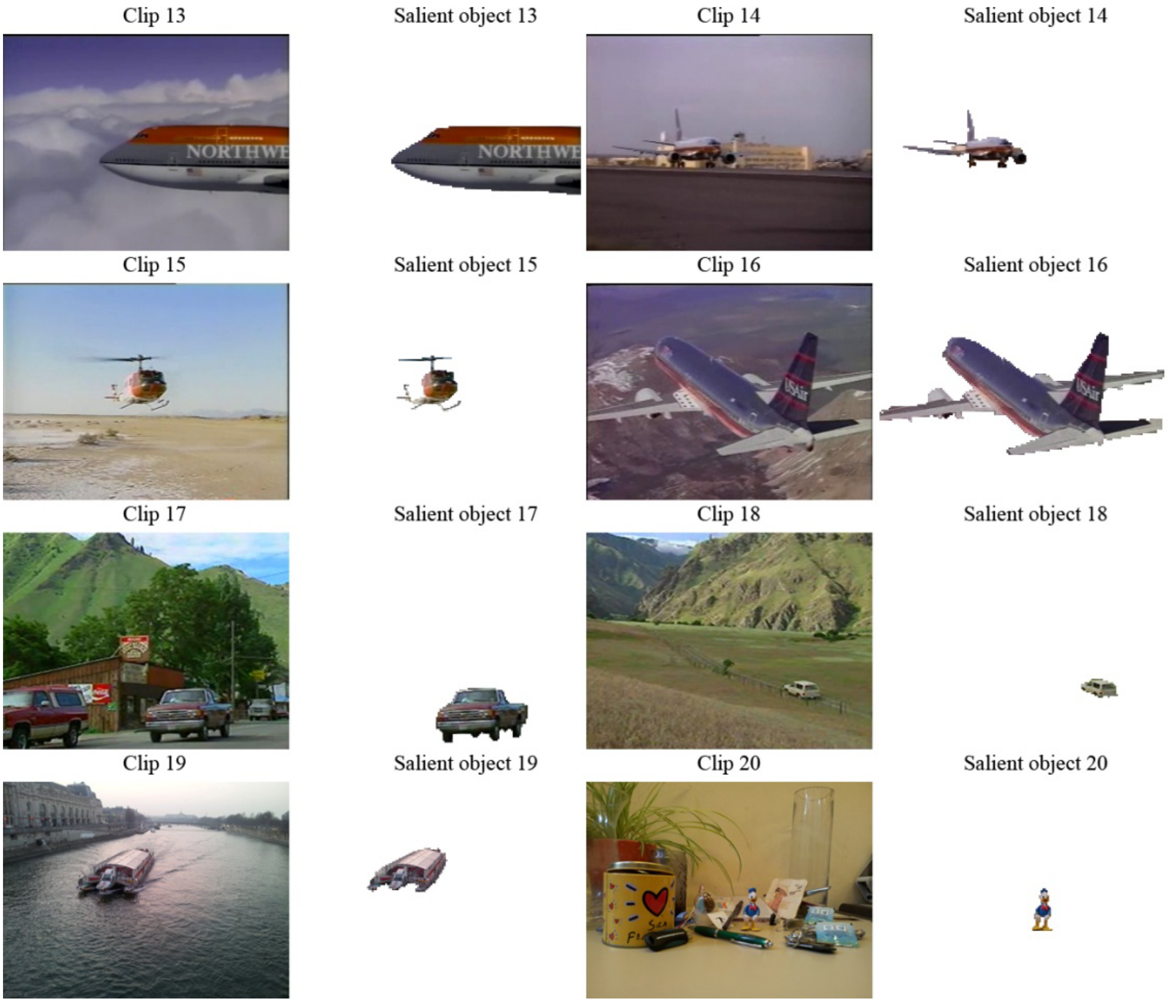


Figure 5.23. Video database - Set 1.

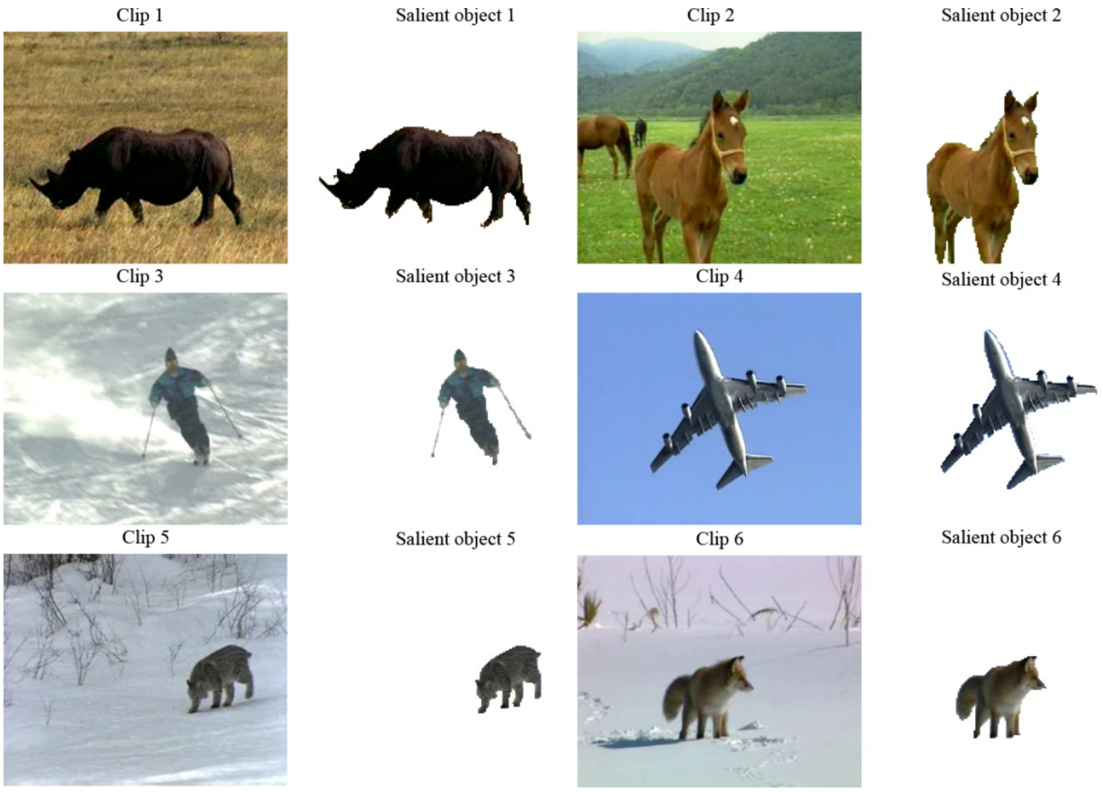


Figure 5.24. Video database - Set 2.

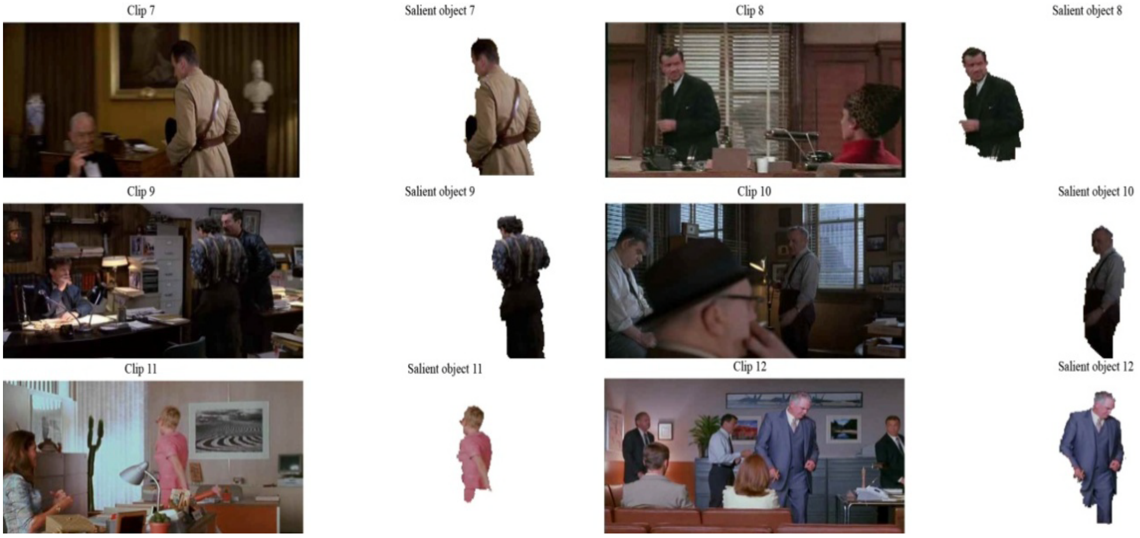
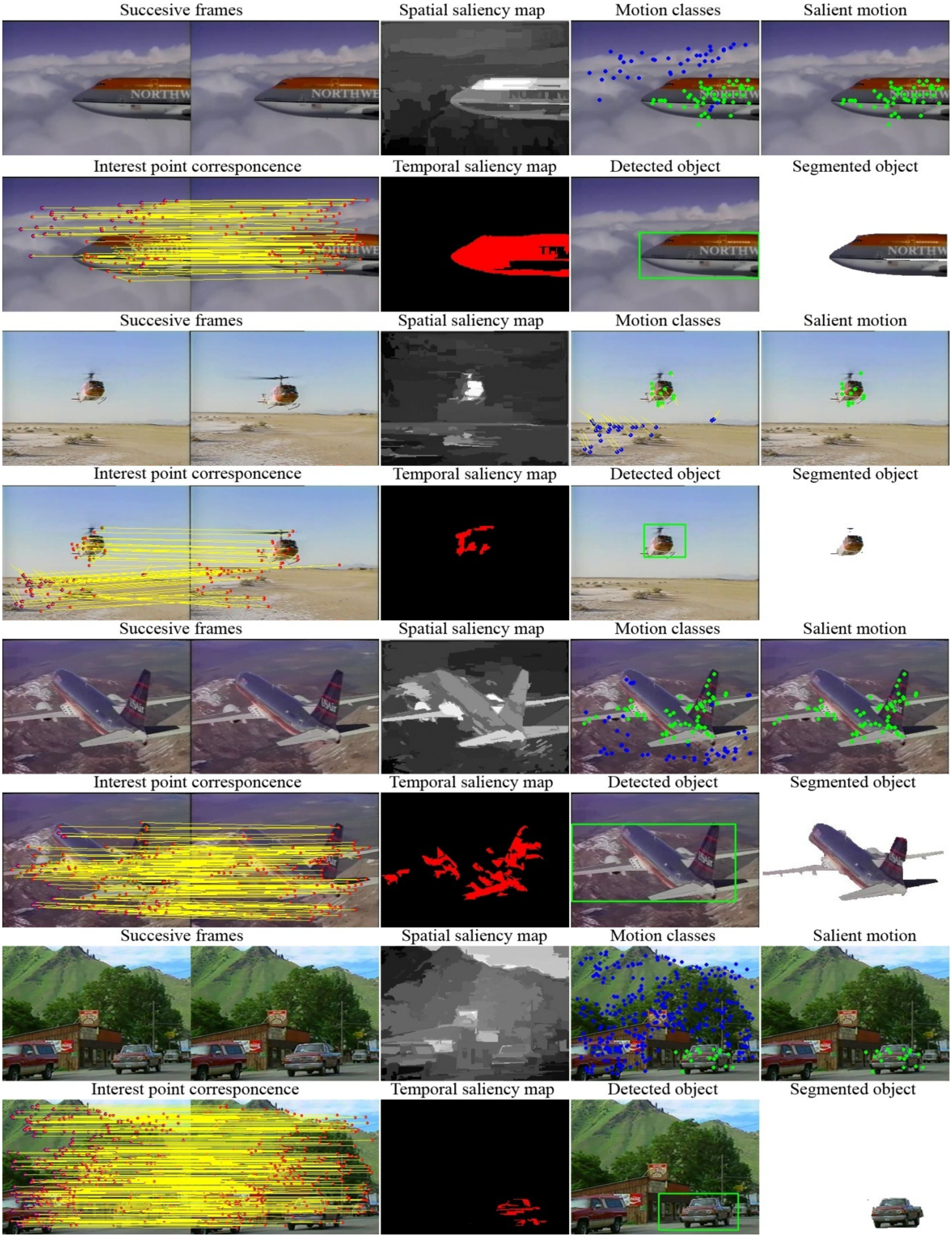


Figure 5.25. Video database - Set 3.

Each video segment contains a single salient object, corresponding to different semantic types: humans performing various activities, animals on the wild, vehicles (both on the ground and air). The videos in *Set 1* are mostly documentaries, noisy, and vary in style and date of production, with a resolution of 341 x 256 pixels. The videos in *Set 2* (with a resolution of 352 x 258 pixels) contain important camera and object movement, but with smooth background or without excessive texture, while the movies from *Set 3* (with a resolution of 640 x 264 pixels) include dark, cluttered and highly dynamic scenes which make them very challenging for an automatic object extraction system. In addition, various types of both camera and multiple object motions are presented.

Some object detection results are presented in Figures 5.26, 5.27, 5.28 and 5.29. Let us first note that for videos with rich texture or including multiple objects, the result of a spatial attention model is, in most of the cases, unrepresentative (Column - Spatial saliency map). However, after incorporating the information associated to the temporal attention model, the method successfully detects the relevant moving regions.



Succesive frames, Spatial saliency map, Motion classes, Salient motion

Interest point corresponcence, Temporal saliency map, Detected object, Segmented object

Succesive frames, Spatial saliency map, Motion classes, Salient motion

Interest point corresponcence, Temporal saliency map, Detected object, Segmented object

Succesive frames, Spatial saliency map, Motion classes, Salient motion

Interest point corresponcence, Temporal saliency map, Detected object, Segmented object

Succesive frames, Spatial saliency map, Motion classes, Salient motion

Interest point corresponcence, Temporal saliency map, Detected object, Segmented object

Figure 5.26. Experimental results obtained on videos in Set 1

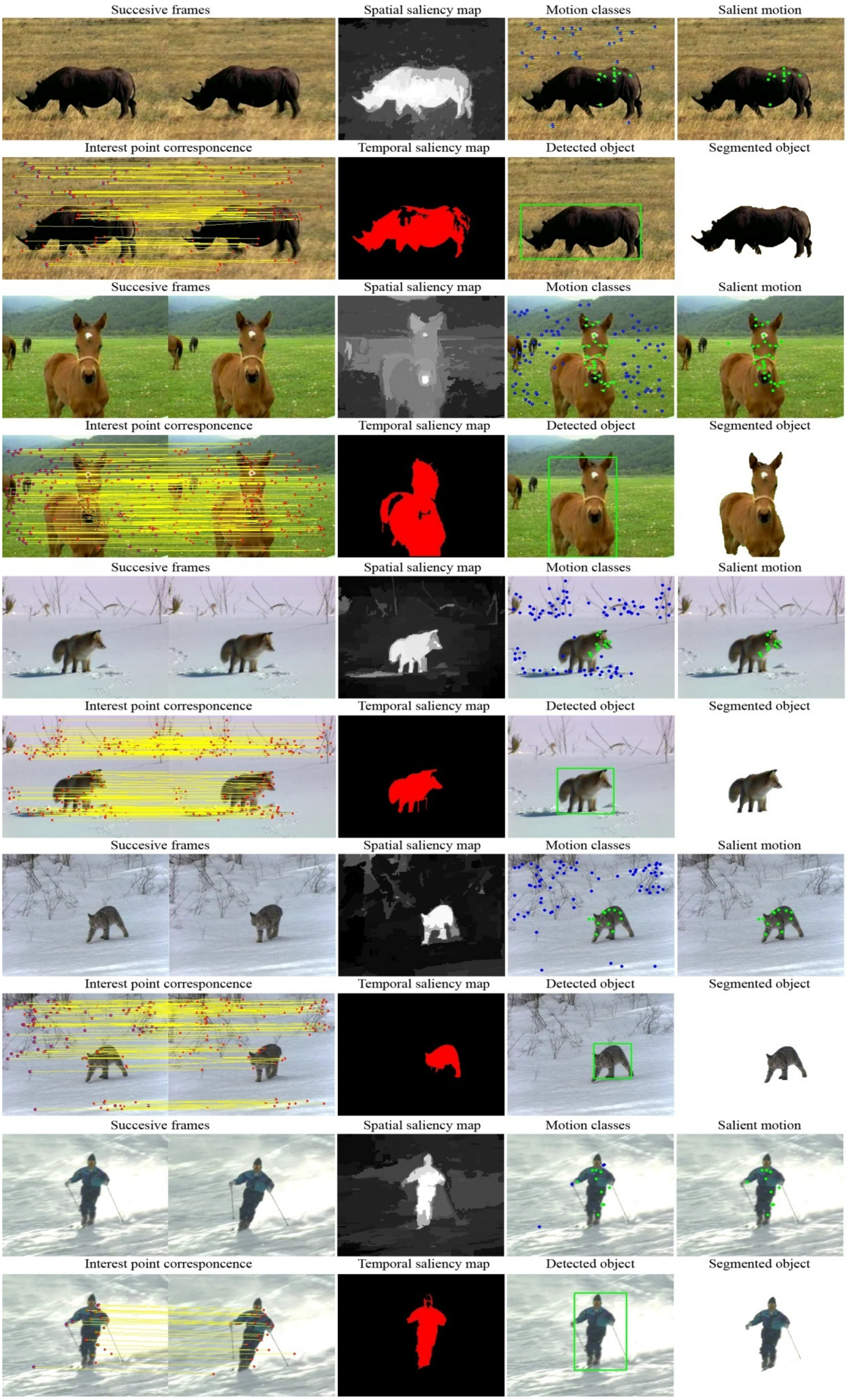


Figure 5.27. Experimental results obtained on videos in Set 2

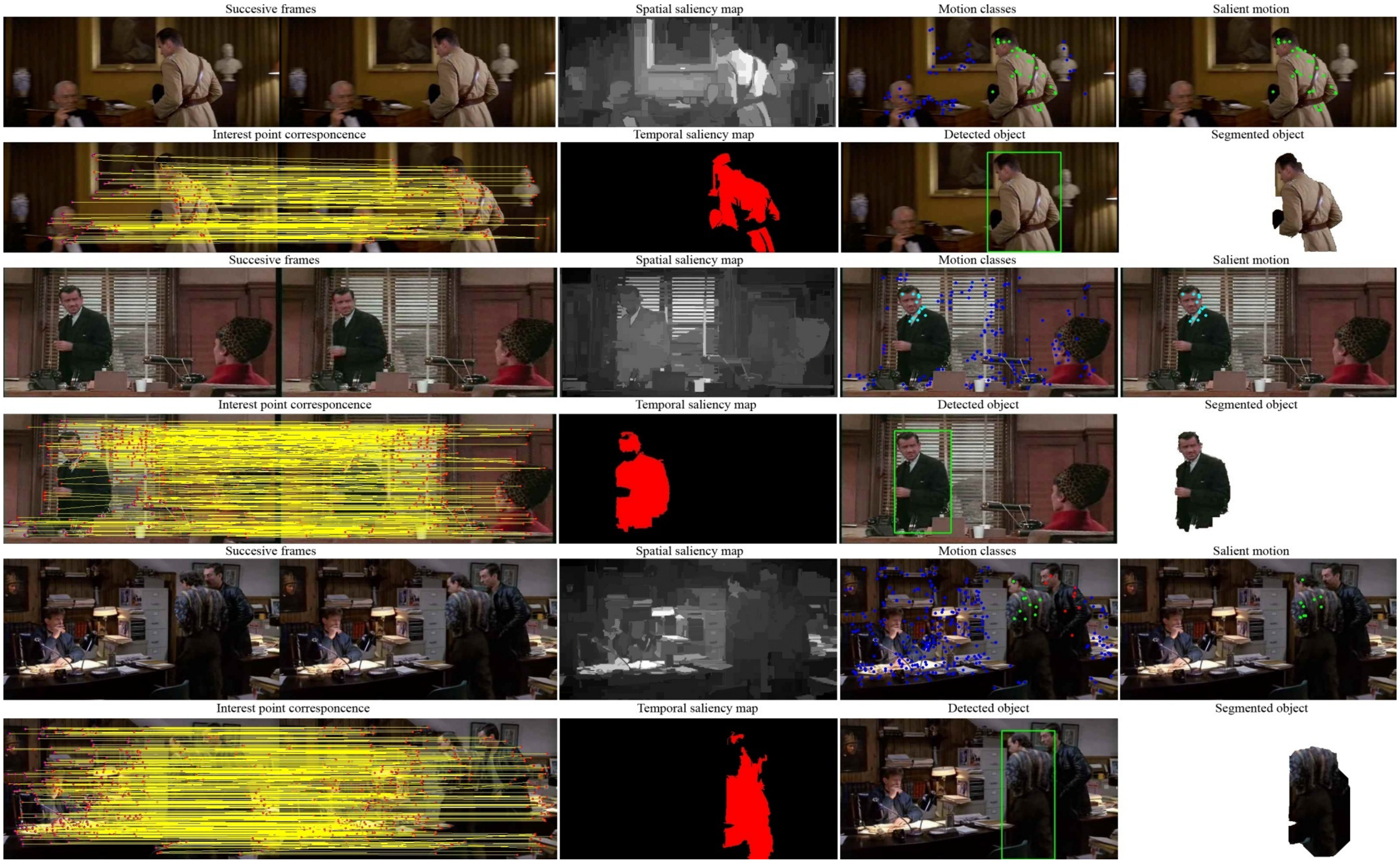


Figure 5.28. Experimental results obtained on video Set 3 (Part 1)

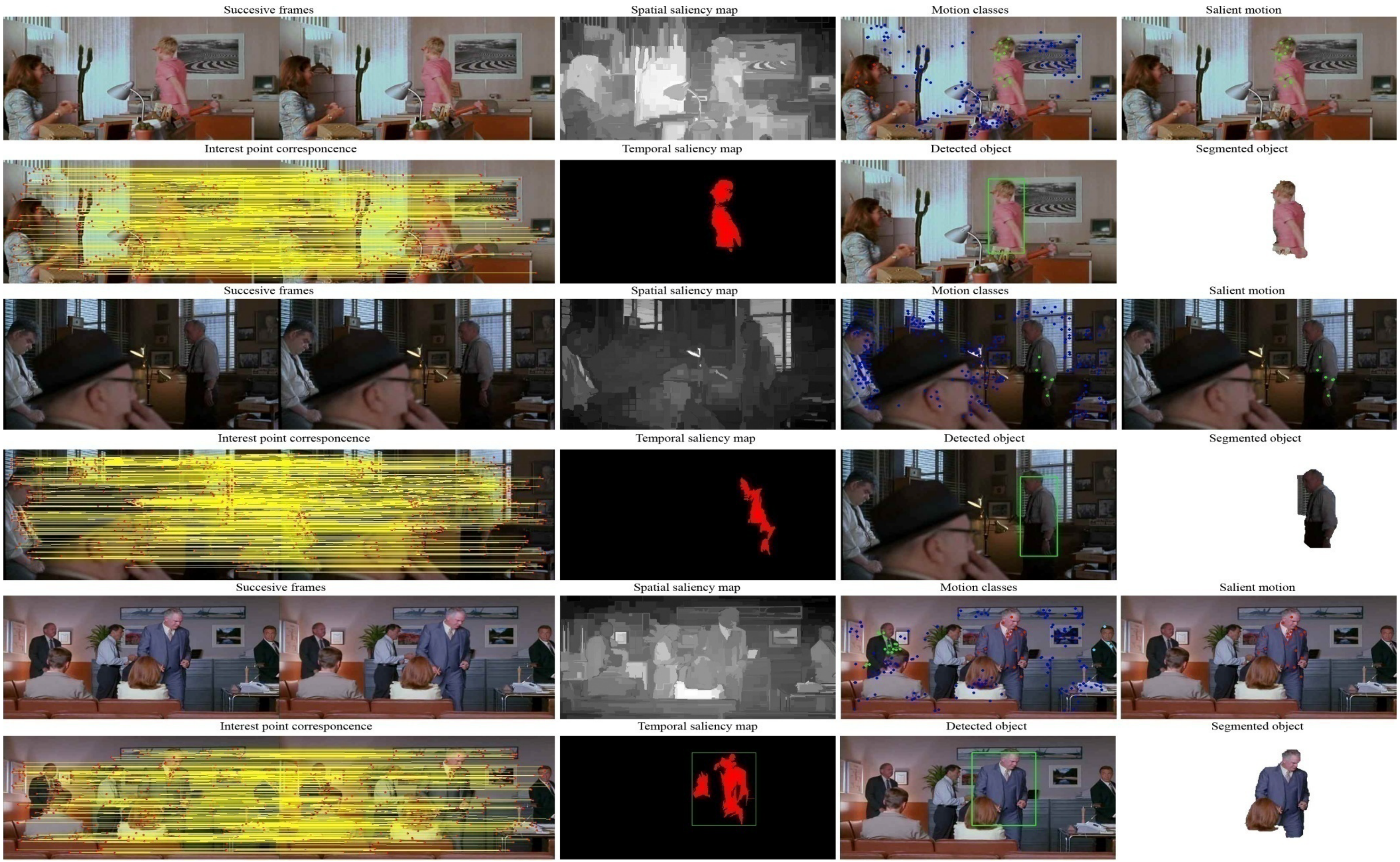


Figure 5.29. Experimental results obtained on video Set 3 (Part2)

For example; if we consider the case of the clip 17 in Figure 5.26, the salient object is a moving car of small size with similar color features as the background. As it can be observed, the spatial attention model detects as salient the sky, but after incorporated the motion information the system is able correct identify the car.

For the videos presented in Figure 5.27 in which the salient object is of large dimensions, located in the centre of the image and without textured background or multiple moving objects the results obtained by the spatial attention models are more than satisfactory and the temporal attention model only consolidates the obtained result.

The impact of motion information is even more important for scenes with rich texture, as in the case of the videos presented in Figure 5.28 and 5.29. Here, the output of a spatial saliency system is useless because the technique is not able to distinguish between different types of regions. However, with the help of the dynamic model the temporal attention becomes dominant and we are able to identify were interesting action happen. For these videos, the spatial attention model returns an erroneous salient map because all the objects presented in the videos are described by the same set of colors. But after incorporating the motion information the system performers well and detects were a relevant change occurs.

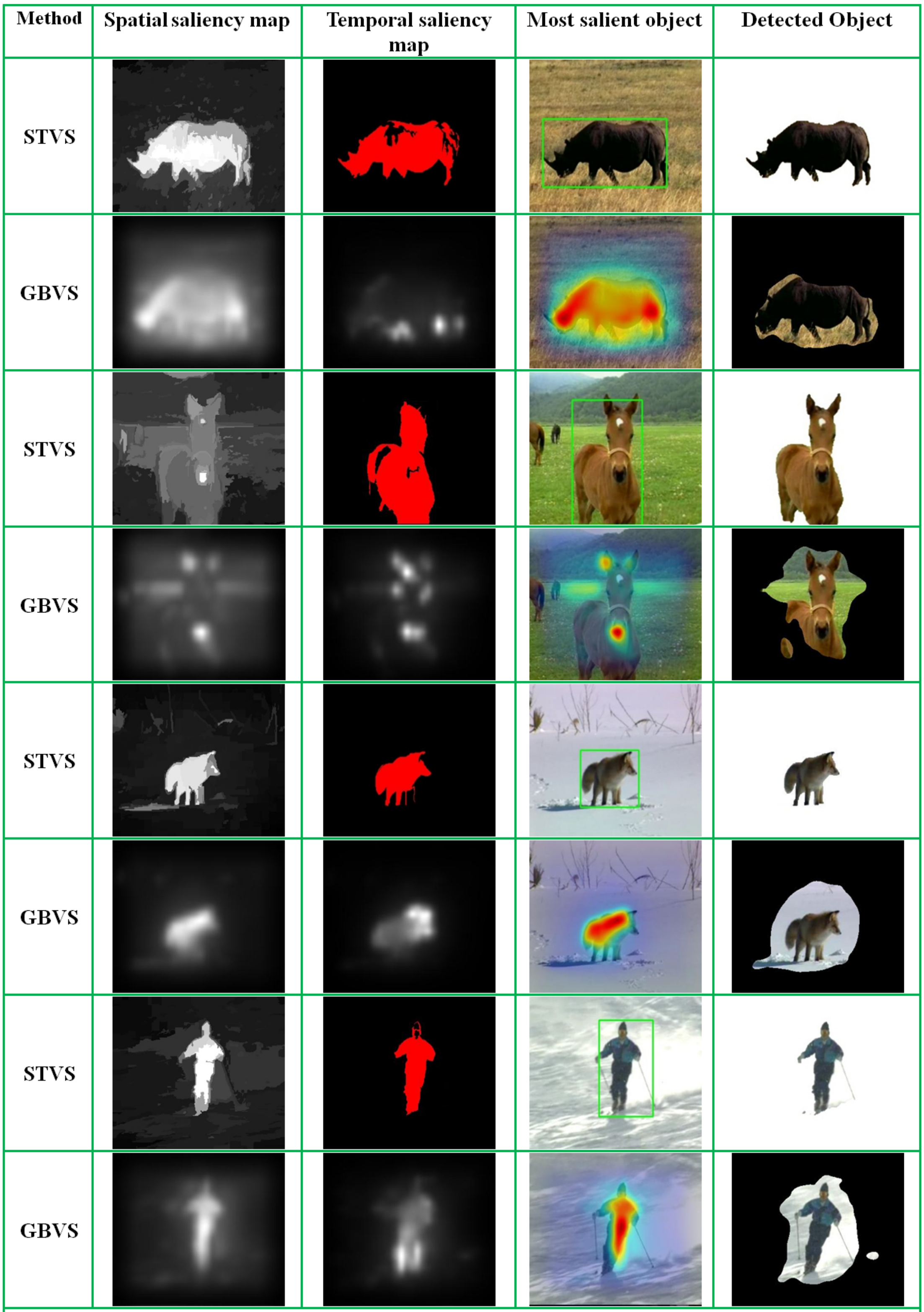
In order to further demonstrate the quality of the proposed technique (Spatio-temporal visual saliency (STVS)) we compared it with one of the most representative methods existent in the technical literature called the Graph-Based Visual Saliency (GBVS) [Harel07], using the author’s own implementation which is available on the web [*www.klab.caltech.edu/~harel/share/**gbvs**.php*]. In Figure 5.30 and 5.31 we present the experimental results obtained. We used for evaluation the same video database of 20 movies considered.

After analyzing the experimental results presented in Figure 5.30 and 5.31 we determine that both methods are able to correctly identify the position of a salient object. However, the GBVS technique suffers from the following limitations:

- The salient regions are not uniformly highlighted, in this case the accent is put on the center or on distinctive elements (*e.g.* colors),
- The object boundaries are not well defined. In the case the technique returns a region and not an object,
- The high frequencies in the saliency map (colored with red) are generated in most of the cases by texture, noise or block artifacts.

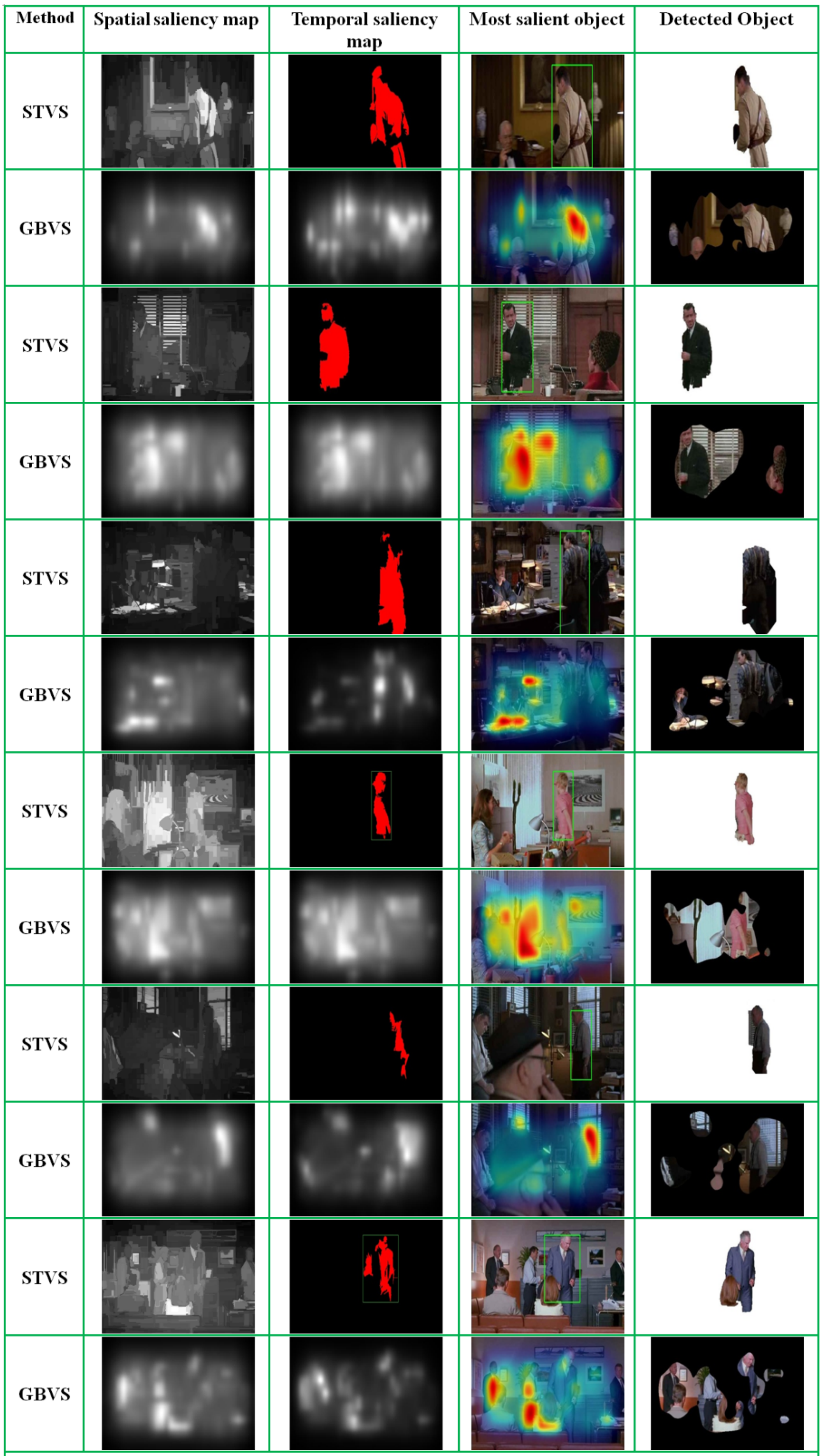
In contrast, the proposed method makes it possible to successfully achieve highly accurate detection results and overcomes, in most cases, such limitations.

In the second phase of our experimental evaluation, we have examined the impact of different parameters involved in our method, on the detection performances. We started by determining the effect the background/foreground separation threshold *E* (Equation (5.34)) on the camera motion estimation. We used for evaluation the video from *Set 3* because in this case the number of interest points correctly matched is larger than 100, which allow us to determine accurately the influence of the considered parameter in the saliency map extraction process.



STVS – Spatio-Temporal Visual Saliency ; GBVS – Graph Based Visual Saliency

Figure 5.30. Comparative evaluation of the proposed technique and the method introduces in [Harel07] on Set 2.



STVS – Spatio-Temporal Visual Saliency ; GBVS – Graph Based Visual Saliency

Figure 5.31. Comparative evaluation of the proposed technique and the method introduces in [Harel07] on Set 3.

As evaluation metrics, we have considered the traditional Recall (*R*), Precision (*P*) and F1 norm (*F1*) measures, defined as follows:

$$R = \frac{D}{D + MD}, \; P = \frac{D}{D + FA} \text{ and } F1 = \frac{2 \times P \times R}{P + R} \qquad (5.39)$$

where *D* is the number of the detected interest points belonging to the background, *MD* is the number of missed detections, and *FA* is the number of false alarms (points that should be included in motion classes but are erroneous considered as belonging to the background). Ideally, all three parameters should be equal to 100%, which correspond to the case where all existing interest points are assigned correctly to motion classes, without any false alarm.

The obtained results are summarized in Table 5.1.

Table 5.1. Camera motion estimation for different threshold parameters.

| **Video title** | **Nr. of interest points** | **Threshold (pixels)** | **False Alarm (FA)s** | **Missed Detection (MD)** | **Correctly Detected (D)** | **Precision** | **Recall** | **F1 score** |
|---|---|---|---|---|---|---|---|---|
| Clip 7 | 107 | **2** | 0 | 13 | 92 | 1 | 0,8761 | 0,934 |
| | | **5** | 0 | 11 | 95 | 1 | 0,8962 | 0,945 |
| | | **7** | 2 | 8 | 99 | 0,9801 | 0,9252 | **0,951** |
| | | **10** | 6 | 5 | 104 | 0,9454 | 0,9541 | 0,949 |
| Clip 8 | 218 | **2** | 0 | 21 | 197 | 1 | 0,9036 | 0,949 |
| | | **5** | 0 | 14 | 204 | 1 | 0,9357 | **0,966** |
| | | **7** | 3 | 12 | 206 | 0,98563 | 0,9449 | 0,964 |
| | | **10** | 8 | 4 | 214 | 0,9631 | 0,9596 | 0,961 |
| Clip 9 | 306 | **2** | 7 | 23 | 283 | 0,9758 | 0,9248 | 0,949 |
| | | **5** | 11 | 18 | 288 | 0,9632 | 0,9411 | **0,952** |
| | | **7** | 18 | 15 | 291 | 0,9417 | 0,9509 | 0,946 |
| | | **10** | 26 | 11 | 295 | 0,919 | 0,9640 | 0,940 |
| Clip 10 | 163 | **2** | 0 | 15 | 148 | 1 | 0,9079 | 0,951 |
| | | **5** | 3 | 12 | 151 | 0,9805 | 0,9263 | **0,952** |
| | | **7** | 7 | 10 | 153 | 0,9562 | 0,9386 | 0,947 |
| | | **10** | 9 | 8 | 155 | 0,9451 | 0,9509 | 0,948 |
| Clip 11 | 172 | **2** | 0 | 12 | 160 | 1 | 0,9302 | 0,963 |
| | | **5** | 0 | 11 | 161 | 1 | 0,9360 | 0,966 |
| | | **7** | 2 | 7 | 164 | 0,9879 | 0,959 | **0,973** |
| | | **10** | 4 | 6 | 166 | 0,9764 | 0,9651 | 0,970 |
| Clip 12 | 264 | **2** | 0 | 14 | 250 | 1 | 0,9469 | 0,972 |
| | | **5** | 2 | 12 | 252 | 0,9921 | 0,9545 | **0,972** |
| | | **7** | 5 | 10 | 254 | 0,9806 | 0,9621 | 0,968 |
| | | **10** | 8 | 7 | 257 | 0,9698 | 0,9734 | 0,968 |

The following conclusion can be highlighted: with the increase of the threshold parameter the number of interest points assigned to the background is naturally increasing. In this case, the algorithm cannot correctly distinguish between motion classes and camera motion (a large number of false alarms) because the allowed estimation error is too big. If the threshold has a low value then the number of missed detected points (points that should belong to the background) is large. In most of the cases these points are errors of motion estimation, not correctly matched between successive frames and present large magnitude values of the motion vectors. It should be noted that these points are further eliminated using step 5 of the proposed algorithm. As it can be notice our algorithm is not sensitive with the variation of the threshold parameter. So, a value between 2 and 10 pixels for the threshold will not affect the system overall efficiency situated around 95% in terms of F1 score.

Let us now analyze the effect of the angular deviation threshold ($Th_{\theta}$) has on motion classes estimation. We used for evaluation the video from *Set 3* because in this case the number of motion classes is superior to 2 which allow us to determine exactly the influence of the considered parameter in the saliency map extraction process.

The results presented in Table 5.2 show the impact the angular deviation threshold has on motion vectors clustering.

Table 5.2. Motion classes estimation for different angular deviation values.

| **Video title** | **Nr. of interest points** | **Threshold (degrees)** | **False Alarm (FA)** | **Missed Detection (MD)** | **Correctly Detected (D)** | **Precision** | **Recall** | **F1 score** |
|---|---|---|---|---|---|---|---|---|
| Clip 7 | 17 | **15** | 0 | 12 | 5 | 1 | 0,2941 | 0,4545 |
| | | **25** | 0 | 5 | 12 | 1 | 0,7058 | 0,8275 |
| | | **35** | 0 | 2 | 15 | 1 | 0,8823 | 0,9375 |
| | | **45** | 0 | 0 | 17 | 1 | 1 | **1** |
| Clip 8 | 12 | **15** | 8 | 1 | 11 | 0,5789 | 0,9166 | 0,7096 |
| | | **25** | 8 | 0 | 12 | 0,6 | 1 | **0,75** |
| | | **35** | 11 | 0 | 12 | 0,5217 | 1 | 0,6857 |
| | | **45** | 12 | 0 | 12 | 0,5 | 1 | 0,6666 |
| Clip 9 | 16 | **15** | 0 | 12 | 4 | 1 | 0,25 | 0,4 |
| | | **25** | 0 | 7 | 9 | 1 | 0,5625 | 0,72 |
| | | **35** | 0 | 5 | 11 | 1 | 0,6875 | 0,8148 |
| | | **45** | 0 | 5 | 11 | 1 | 0,6875 | **0,8148** |
| Clip 10 | 34 | **15** | 4 | 10 | 24 | 0,8571 | 0,7058 | 0,7741 |
| | | **25** | 4 | 6 | 28 | 0,875 | 0,8235 | 0,8484 |
| | | **35** | 4 | 5 | 29 | 0,8787 | 0,8529 | 0,8656 |
| | | **45** | 5 | 4 | 60 | 0,8529 | 0,8787 | **0,8656** |
| Clip 11 | 42 | **15** | 6 | 22 | 20 | 0,7692 | 0,4761 | 0,5882 |
| | | **25** | 7 | 18 | 24 | 0,7741 | 0,5714 | 0,6575 |
| | | **35** | 7 | 14 | 28 | 0,8 | 0,6667 | 0,7272 |
| | | **45** | 10 | 12 | 30 | 0,75 | 0,7142 | **0,7317** |
| Clip 12 | 30 | **15** | 0 | 11 | 19 | 1 | 0,6333 | 0,7755 |
| | | **25** | 0 | 7 | 23 | 1 | 0,7666 | 0,8679 |
| | | **35** | 2 | 5 | 25 | 0,9259 | 0,8333 | 0,8771 |
| | | **45** | 2 | 5 | 25 | 0,9259 | 0,8333 | **0,8771** |

With the increase of the maximum value allowed for the angular deviation the number of missed detected (not clustered into motion classes) interest points is reduced due to a higher level of freedom when estimating the current position of an interest point between successive frames. In the same time the number of false alarms, the number points erroneous assigned to motion classes is also increasing due to the motion displacement inconsistency. A value between 25 and 35 degree for angular deviation will ensure a compromise between the motion classes correctly detected and the number of false alarms.

The promising results obtained, led us to investigate how the proposed spatio-temporal saliency model can be enhanced with other visual information. In particular, we have considered the issue of extending the proposed method to stereoscopic video data, as described in the next section.

### 5.2.4. Stereoscopic visual attention model

In the case of 3D videos the depth perception it is another important factor that affects the human visual attention much more than any motion or texture contrast existent for traditional 2D videos. Because the stereoscopic perception can also be represented as a 2D video and its associated depth (which indicates the relative distance between video objects and the camera) we introduce an enhanced salient object detection system that integrates a stereoscopic visual attention model.

Depth maps are generated from the disparities between neighbouring views. Then a novel dynamic fusion model is developed that integrates all information from each saliency map (spatial, temporal and depth).

The depth maps are generated based on the technique presented in [Smith09]. For a stereo image pair $I_1$ and $I_2$ we determine the associated disparity maps $D_1$ and $D_2$ by minimizing the following objective function $\Phi$:

$$\Phi(D_1, D_2) = \Phi_{ph}(D_1, D_2) + \Phi_{sm}(D_1) + \Phi_{sm}(D_2) \tag{5.40}$$

where $\Phi_{ph}$ measure the photo consistency and $\Phi_{sm}$ regularizes the depth maps.
In order to determine the photo consistency we used the following equation that incorporates the geometric visibility:

$$\Phi_{ph}(D_1, D_2) = \sum_{p \epsilon I_1} \phi_{ph}(d_p, d_q) \tag{5.41}$$

where $d_p = D_1(p)$ is the disparity for pixel $p$ in $I_1$, $q = p + D_1(p)$ is the $p^{\text{th}}$ corresponding pixel in $I_2$ and $d_q = D_2(q)$ is the disparity for $q$ in $I_2$. $\phi_{ph}$ is defined as:

$$\phi_{ph} = \begin{cases} 0, if\ d_p < d_q \\ \min(0, |x| - val) \cdot \left(\left\|c_p - c_q\right\|^2\right), if d_p = d_q, \\ \infty, if\ d_p > d_q \end{cases} \tag{5.42}$$

where $c_p = I_1(p), c_q = I_2(q)$ and *val* is a robust measure for photo consistency.

The regularization term $\Phi_{sm}(D)$ is given by the following formula:

$$\Phi_{sm}(D) = \sum_{p \in I} \phi_{sm}(d_p, \{d_q\}_{q \epsilon N_p}) , \tag{5.43}$$

where $\phi_{sm}$ models the correlation between the disparity $d_p$ for pixel $p$ and the other disparity $d_q$ in p’s neighborhood $N_p$.

After computing the disparity maps (Figure 5.32) as presented above we include the depth information in the proposed spatio-temporal attention model.

In this case, the *Step 7* of our algorithm (*cf.* Section 5.2.1) is modified as follows:

***Step 7***: *Salient motion detection* – For all the motion classes determined at *Step 3* and *4* we compute their depth values as:

$$DepthClass(M_i) = \frac{\sum_{j=1}^{m_i} Depth(p_{1,i})}{m_i}, i = \overline{1, N}\,, \qquad (5.44)$$

where $m_i$ is the total number of points included in motion class $M_i$, $N$ is the total number of classes and $Depth(p_{1,i})$ is the value of an interest point $p_{1,i}$ in the image depth map. In this case, the salient motion is determined as:

$$SalientMotion = \max_{i=1,N}\{DepthClass(M_i)\}. \qquad (5.45)$$

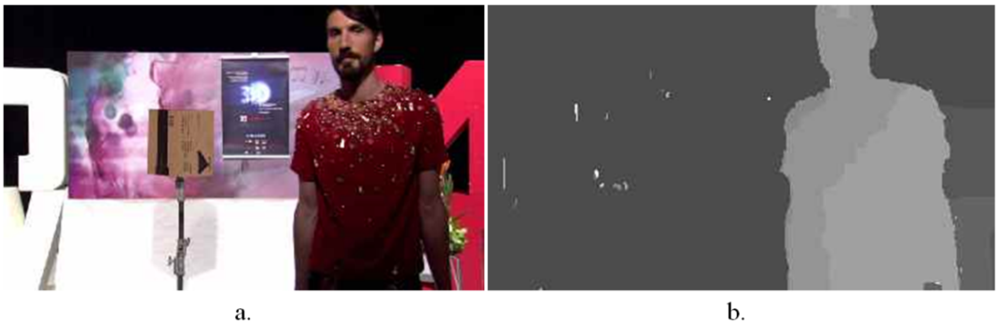


Figure 5.32. Disparity map extraction. (a) Original image; (b) Disparity map.

In order to evaluate the performance of the proposed stereoscopic visual attention model we perform detection experiments on 3D videos provided by the Heinrich Hertz Institute [Feldmann08] and from the videos acquired within the framework of the French FUI7 3D-LIVE project (www.3dlive-project.com).

Figure 5.33 and 5.34 illustrate some experimental results obtained.

For sequences with complex background the spatial attention model is not accurate enough to detect the interest object in 3D videos. When large motion contrasts are encountered in video stream the method tends to focus on these regions as potential attention areas. However, this is not always true, for example when an attention object has a shadow the shadow has the same high motion contrast but is not an interesting area. This situation can be avoided by using the depth maps. Obviously, we cannot expect to detect the visual attention only by using the disparity maps because the area situated near the camera focus in not always an attention area. Therefore we incorporated all information in order to take a decision and to overcome the inherent shortcomings of each individual attention model.

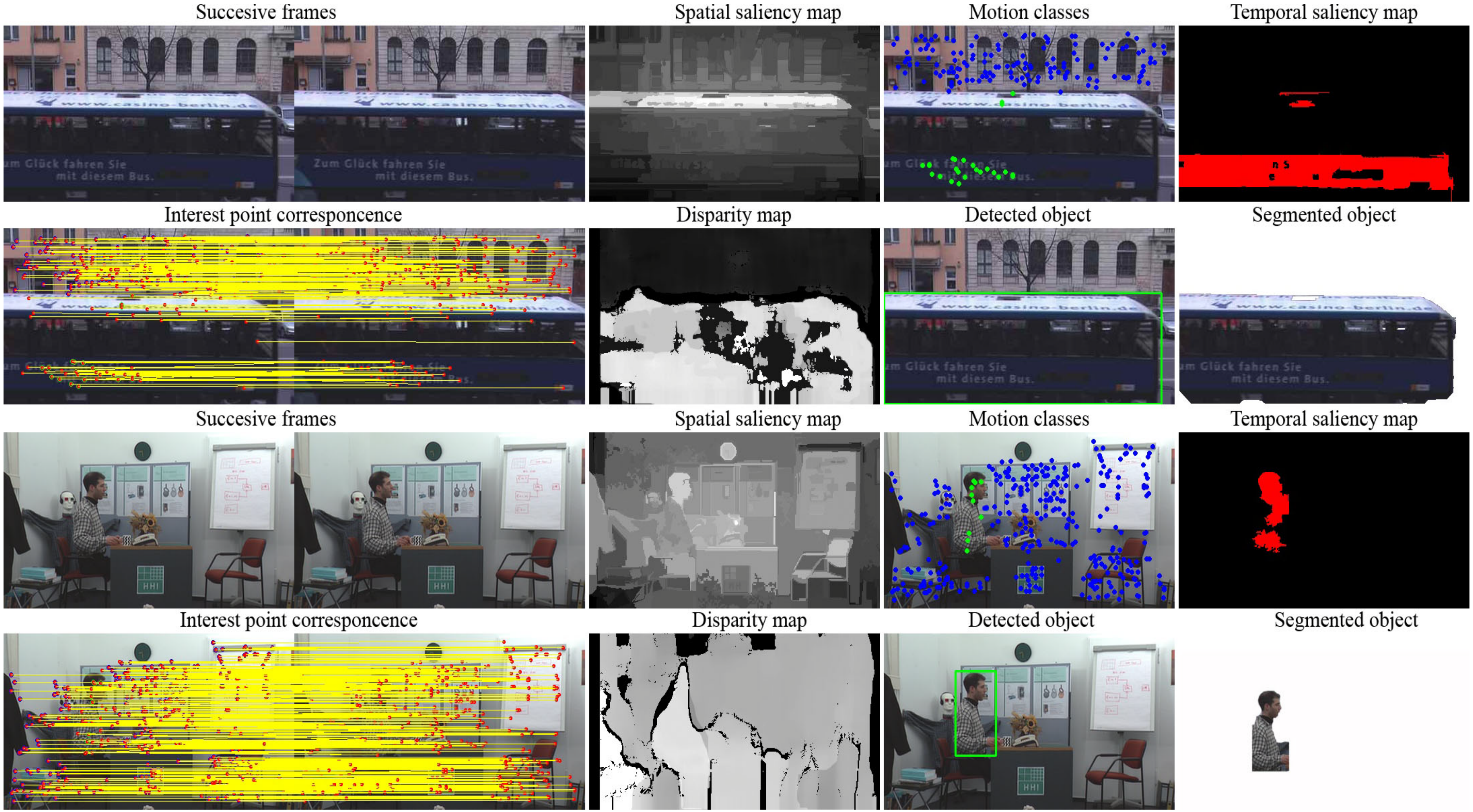


Figure 5.33. Experimental results obtained on 3D videos from Heinrich Hertz Institute

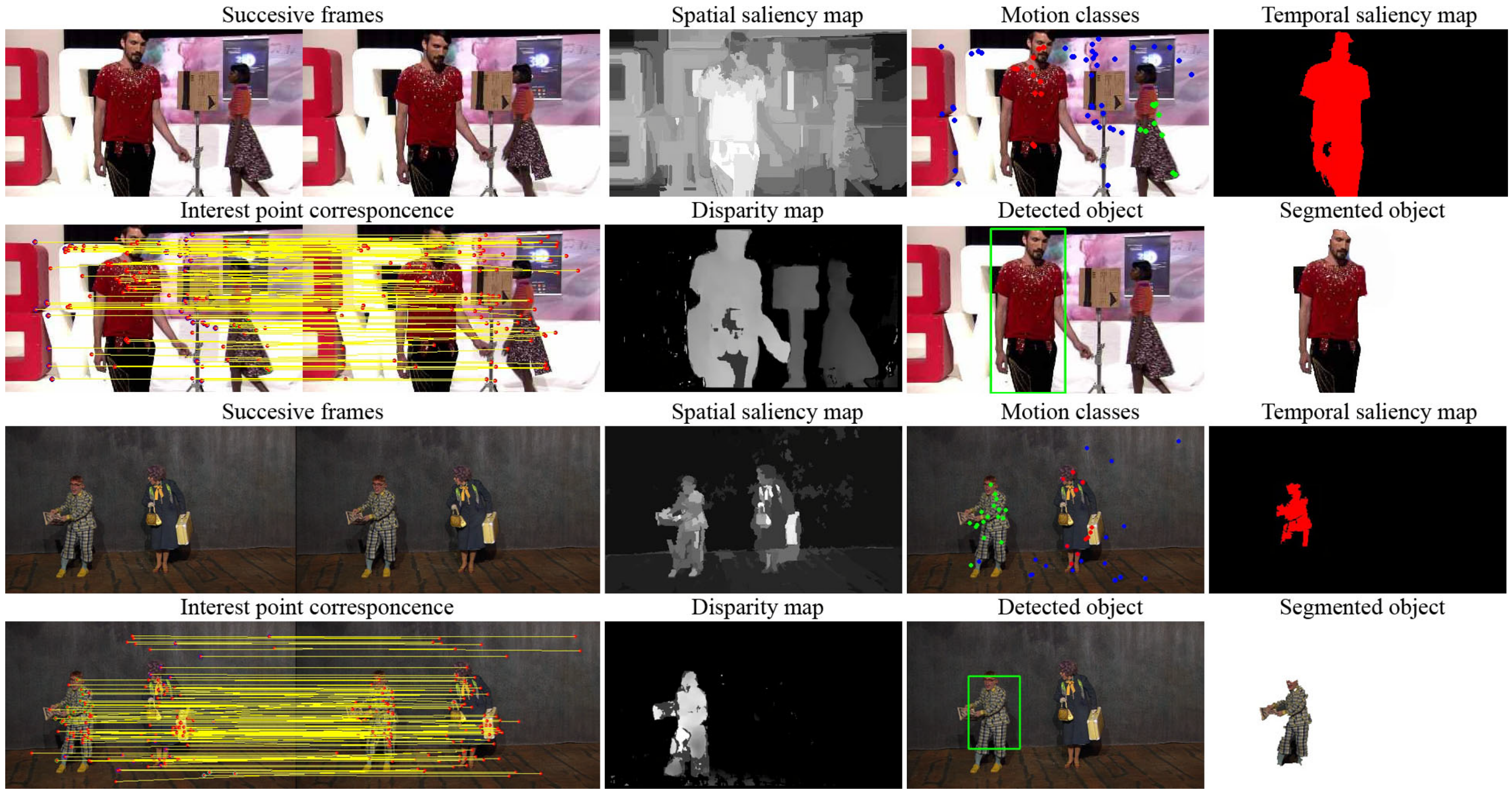


Figure 5.34. Experimental results obtained on 3D videos from the 3D-LIVE project.

## 5.3. CONCLUSIONS

In this chapter we have addressed the problem of salient object detection in image/video streams. First, we have drawn the state of the art in the field for both still images and videos. The analysis of the literature reveals the importance of considering the motion information. Thus, existing video-dedicated methods combine relevant motion models with spatial attention in order to build efficient saliency models. However, in practice, the video motion can be caused by the salient objects, but also by background objects or camera movement. In this case, different types of motion need to be analyzed and appropriately taken into account.

Then, we have proposed an automatic salient object extraction system based on a spatiotemporal attention detection framework. The spatial model is developed starting from the region-based contrast, while the temporal model rely on the interest points correspondence, geometric transforms (between keyframes based on a homographic motion model), motion classes estimation (using agglomerative clustering and *k-NN* algorithm) and regions temporal consistency.

The technique was validated on three video sets including more than 20 videos and it is characterized by robustness with complex background distracting motions and does not require any initial knowledge about the object size or shape. The various experimental results and comparisons, with other relevant methods existent in the technical literature, demonstrate the effectiveness of the proposed algorithm.

Finally, we have extended the proposed algorithm to 3D stereoscopic videos by incorporating the information extracted from disparity maps. The proposed model is not only able to efficiently simulate stereoscopic visual attention of humans, but can also eliminate noise and maintain high robustness for videos presenting objects with shadows and various artifacts.

# 6. CONCLUSIONS AND PERSPECTIVES

In this thesis, we have proposed a novel methodological framework for high level temporal video structuring and segmentation, which includes shot boundary detection, keyframe extraction, scene identification and salient object detection/segmentation.

In Chapter 2 we introduced an enhanced shot segmentation technique based on graph partition method and multi-resolution analysis. The key stage of our algorithm concerns the scale space filtering of the derivatives of the similarity vector associated to the graph partition model. Notable, this mechanism makes it possible to enhance the robustness of the detector with respect to camera/large object motion. The filtering stage helps eliminating signal variations caused by motions, while preserving the peaks corresponding to the true transitions. The experimental results clearly demonstrate the superiority of our approach compared with the most salient methods existent in the technical literature, for both abrupt and gradual transitions, with global gains in terms of recall and precision rates of 9.4% and 7.7%, respectively.

In the next stage we have introduced a two-pass analysis process, aiming at reducing the computational complexity. The experimental results obtained demonstrate the improvement due to our technique with respect to the state of the art algorithms, with savings in computational time greater than 25%, for equivalent detection performances.

The video abstraction issue has been tackled in Chapter 3. Here, a novel leap extraction technique has been introduced, which generates static storyboards by selecting a variable number of representative frames based on the input video content variation. The proposed approach makes it possible to increase the computational efficiency with more than 23% and ensures that the story board captures all informational content of the original movie without any irrelevant.

The extracted keyframes are exploited to form scenes (Chapter 4) defined as a collection of shots that present the same theme and share similar coherence in space and time. By exploiting the observation that shots belonging to the same scene have similar visual features, we introduce a novel temporally constrained clustering algorithm that uses adaptive thresholding and neutralized shots. The experimental evaluation conducted on a large set of videos validates our approach, when using as representative features either interest points extracted using SIFT descriptor or HSV colour histogram. The evaluation is done using traditional metrics as precision, recall and F1 norm. The F1 measure, when applying our method on sitcoms scene detection is 84%, while for DVD chapter the detection performance is 77%.

In Chapter 5, we considered the issue of visual saliency defined as the perceptual quality that allows an object, person or pixel to stand out from his neighbors by capturing our attention. In this context, we have introduced a novel bottom-up approach for modeling the spatiotemporal attention in videos. The spatial model is developed starting from a region-based contrast measure associated to individual keyframes. The temporal model relies on interest points correspondence, geometric transforms (*i.e.* homographic motion model), motion classes estimation (using agglomerative clustering) and regions temporal consistency. Finally, the interest object is extracted with the help of GrabCut segmentation which takes as input to the saliency map previously determined. The technique is robust to complex background distracting motions and does not require any initial knowledge about the object size or shape. The various experimental results and comparisons with existent methods demonstrate the effectiveness of the proposed technique.

Our perspectives of future work concern the integration of our method within a more general framework of video indexing and retrieval applications, including object recognition methodologies. On one hand, this can further refine the level of description required in video indexing applications. On the other hand, identifying similar objects in various scenes can be helpful for the scene identification process and for semantic concept detection.

The semantic concept detection will be a natural way of exploiting the results presented in this work, by annotating the identified features existent in the extracted keyframes. Such concepts can be objects, activities, events, scenes…. and can serve for automatic indexing and organization of multimedia collections.

# List of publications

[1]. R. Tapu, T. Zaharia, “Salient object detection based on spatiotemporal attention models”, IEEE International Conference on Consumer Electronics (*ICCE*), Las Vegas, Nevada, USA – accepted for publication (BDI).

[2]. R. Tapu and T. Zaharia, “Video Structuring: From Pixels to Visual Entities”, 20th European Signal Processing Conference (EUSIPCO-2012), ISSN 2076-1465, pp. 1583-1587, Bucharest, Romania, 27-31 August 2012 (ISI).

[3]. R. Tapu and T. Zaharia, „Scene change detection with temporally constrained clustering”, 3rd International Conference on Future Computer and Communications – ICFCC 2011, ISBN: 978-0-7918-5971-1, pp. 71-76, Iasi, Romania, 3-5 June 2011 (ISI).

[4]. R. Tapu and T. Zaharia, “A Complete Framework for Temporal Video Segmentation”, The 1st International Conference on Consumers Electronics, ICCE-2011, ISBN: 978-1-4577-0234-1, pp. 156-160, Berlin, Germany, 2011(BDI).

[5]. R. Tapu and T. Zaharia, “Automatic Multilevel Temporal Video Structuring”, Fifth IEEE International Conference on Semantic Computing, ICSC-2011, ISBN: 978-1-4577-1648-5, pp.387-394, Palo Alto, California, USA, 2011 (BDI).

[6]. R. Tapu and T. Zaharia, “High Level Video Temporal Segmentation”, 7th International Symposium on Visual Computing, ISVC-2011, ISBN: 978-3-642-24027-0, Part I, LNCS 6938, pp. 226–237, Las Vegas, Nevada, USA, 2011 (BDI).

[7]. R. Tapu and T. Zaharia, “Video Segmentation and Structuring for Indexing Applications”, International Journal of Multimedia Data Engineering and Management (IJMDEM), ISSN: 1947-8534, Volume 2, Issue 4, pp.38-58, 2011.

[8]. R. Tapu, T. Zaharia and F. Preteux, “A scale-space filtering-based shot detection algorithm”, IEEE 26th Convention of Electrical and Electronics Engineers in Israel, ISBN: 978-1-4244-8682-3, pp.919-923, Eilat, Israel, 2010 (BDI).

# Bibliography

[Achanta08] R. Achanta, F. Estrada, P. Wils, and S. Susstrunk, "*Salient region detection and segmentation*", International Conference on Computer Vision Systems, 2008.

[Achanta09] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "*Frequency-tuned salient region detection*", In CVPR, pp. 1597–1604, 2009.

[Adams00] B. Adams, C. Dorai, and S. Venkatesh, "*Towards automatic extraction of expressive elements from motion pictures*," in Proc. IEEE International Conference on Multimedia and Expo, vol. II, pp. 641–645, IEEE, New York, NY, USA, July 2000.

[Akutsu92] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba. "*Video Indexing Using Motion Vectors*", Proc. SPIE Visual Communications and Image Processing, Vol. 1818, pp. 1522-1530, 1992.

[Alattar97] A. M. Alattar, "*Detecting Fade Regions in Uncompressed Video Sequences*", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3025-3028, 1997.

[Aner02] A. Aner, J. R. Kender, "*Video summaries through mosaic-based shot and scene clustering*", In Proc. European Conf. Computer Vision, pp. 388–402, 2002.

[Anguita10] D. Anguita, A. Ghio, N. Greco, L. Oneto, S. Ridella, "*Model selection for support vector machines: Advantages and disadvantages of the machine learning theory*", In Proc. of the Int. Joint Conference on Neural Networks, 2010.

[Arman93] F. Arman, A. Hsu, M.Y. Chiu, "*Image processing on compressed data for large video database*". ACM International Conference on Multimedia, pp. 267–272, August, 1993.

[Belardinelli09] A. Belardinelli, F. Pirri, and A. Carbone, "*Motion Saliency Maps from Spatiotemporal Filtering. In Attention in Cognitive Systems*", Lecture Notes in Artificial Intelligence, pp. 112-123, 2009.

[Bescos00] J. Bescos, J. Menendez, G. Cisneros, J. Cabrera, J. Martinez, "*A unified approach to gradual shot transition detection*", In Proc. IEEE ICIP '00, pages Vol III: 949–952, 2000.

[Boccignone05] G. Boccignone, A. Chianese, V. Moscato, A. Picariello, "*Foveated shot detection for video segmentation*," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 3, pp. 365–377, Mar. 2005.

[Bouthemy99] Bouthemy P., Garcia C.: "*Scene segmentation and image feature extraction for video indexing and retrieval*", Proceeding on International Conference on Visual Information and Information Systems, pp. 245–252, 1999.

[Briechle01] K. Briechle and U. Hanebeck, "*Template matching using fast normalized cross correlation*", In Proc. of SPIE AeroSense Symposium, volume 4387, Orlando, Florida, USA, 2001.

[Bruyne08] S. Bruyne, D. Deursen, J. Cock, W. Neve, P. Lambert, and R. Walle, "*A compressed-domain approach for shot boundary detection on H.264/AVC bit streams*," J. Signal Process.: Image Commun., vol. 23, no. 7, pp. 473–489, 2008.

[Burges06] J.C. Burges, "*A Tutorial on Support Vector Machines for Pattern Recognition*", Kluwer Academic Publishers, Boston, pp.1-43, 2006.

[Bursuc10] A. Bursuc, T. Zaharia, B. Delezoide, F. Preteux, "*OVIDIUS: An on-line video retrieval platform for multi-terminal access*", International Workshop on Content-Based Multimedia Indexing (CBMI), pp.1-6, 23-25 June 2010

[Calic02] J. Calic, E. Izquierdo, "*Temporal segmentation of mpeg video streams*", EURASIP Journal on Applied Signal Processing, pp. 561–565, June 2002.

[Cardan01]. D. Cardani, "*Adventures in HSV Space*", The advanced Developers Hands on Conference July 2001.

[Carson02] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: "*Image segmentation using expectation-maximization and its application to image querying*", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24(8), pp.1026–1038, 2002.

[Cernekova03] Z. Cernekova, C. Kotropoulos, I. Pitas, "*Video shot segmentation using singular value decomposition*," in Proc. 2003 IEEE Int. Conf. Multimedia and Expo, vol. 2, pp. 301–302, 2003.

[Cernekova06] Z. Cernekova, I. Pitas, C. Nikou, "*Information theory-based shot cut/fade detection and video summarization*", IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 1, pp. 82-91, 2006

[Chaisorn02] Chaisorn L., Chua T.S., Lee C.H.: "*The segmentation of news video into story units*", IEEE proceeding on International Conference on Multimedia and Expo, pp. 73–76, 2002.

[Chandrasekaran09] S. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky, "*Rank-sparsity incoherence for matrix decomposition*," preprint, 2009.

[Chasanis08] V. Chasanis, A. L. Galatsanos, "*Video rushes summarization using spectral clustering and sequence alignment*", In: TRECVID BBC rushes summarization workshop (TVS'08), ACM international conference on multimedia, Vancouver, pp 75–79, 2008.

[Chasanis08] V. Chasanis, A. Likas, N. Galatsanos, "*A Support Vector Machine Approach for Video Shot Detection*", New Direct. in Intel. Interac. Multimedia, SCI 142, pp.45-54, 2008.

[Chasanis09] Chasanis V., Kalogeratos A., Likas A: "*Movie segmentation into scenes and chapters using locally weighted bag of visual words*", In Proceedings of the ACM International Conference on Image and Video Retrieval, pages 35:1–35:7, ACM Press, 2009.

[Chen03] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong- Jiang Zhang, and He-Qin Zhou, "*A visual attention model for adapting images on small displays*," ACM Multimedia Systems Journal, pp. 353–364, 2003.

[Chen08] Y.-T. Chen, C.-S. Chen, "*Fast human detection using a novel boosted cascading structure with meta stages*," IEEE Trans. Image Process., vol. 17, no. 8, pp. 1452–1464, Aug. 2008.

[Chen08] Duan-Yu Chen; Hsiao-Rong Tyan; Sheng-Wen Shih; Liao, H.-Y.M: "*Dynamic Visual Saliency Modeling for Video Semantics*," Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on , vol., no., pp.188-191, 15-17 Aug. 2008.

[Cheng11] M. Cheng, N. Zhang, G.and Mitra, X. Huang, and S. Hu, "*Global Contrast based Salient Region Detection*" in IEEE CVPR, pp. 409–416, 2011.

[Choubey97] S. K. Choubey, V. V. Raghavan, “*Generic and fully automatic content-based image retrieval using color*,” Pattern Recog. Lett., vol. 18, no. 11–13, pp. 1233–1240, 1997.

[Chung00] M. Chung, “*A scene boundary detection method*”, In Proc. IEEE ICIP ’00, pages Vol III: pp. 933–936, 2000.

[Cimiano04] P. Cimiano, A. Hotho, S. Staab, “*Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text”*, In Proceedings of the European Conference on Artificial Intelligence, pages 435–439, 2004.

[Ciocca05] G. Ciocca, R. Schettini, “*Dynamic key-frame extraction for video summarization*,” in *Internet Imaging VI*, vol. 5670 of *Proceedings of SPIE*, pp. 137–142, San Jose, California, USA, January 2005.

[Comaniciu02] D. Comaniciu and P. Meer, “*Mean Shift: A Robust Approach Toward Feature Space Analysis*”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002.

[Cooper07] M. Cooper, T. Liu, E. Rieffel, ”*Video segmentation via temporal pattern classification*”, Transactions on Multimedia, 9(3), pp. 610–618, 2007.

[Costaces06] C. Costaces, N. Nikoladis, I. Pitas, “*Video shot detection and condensed representation*” IEEE Signal Processing Magazine, pp. 153-163, March 2006.

[Ding01] C. Ding, X. He, H. Zha, M. Gu, H.D. Simon, “*A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering*,” Proc. Int’l Conf. Data Mining, pp. 107-114, Nov. 2001.

[Dirfaux00] Dirfaux, F., “*Key frame selection to represent a video*”, IEEE International Conference on Image Processing 2000 vol.2, page(s): 275 - 278, 2000.

[Doulamis00] A. D. Doulamis, N. Doulamis, S. Kollias, “*Non-sequential video content representation using temporal variation of feature vectors*”, IEEE Transactions on Consumer Electronics, vol. 46, no. 3, August 2000.

[Drew99] M. Drew, J.Wei, Z. Li, “*Illumination-invariant image retrieval and video segmentation*”, Pattern Recognition, 32(8), pp. 1369–1388, August 1999.

[Duan11] L. J. Duan, C. P.Wu, J.Miao, L. Y. Qing, and Y. Fu, “*Visual saliency detection by spatially weighted dissimilarity*,” in IEEE Int. Conf. Comput. Vision Pattern Recogn, Colorado Springs, CO, Jun. 2011.

[Evangelopoulos08] Evangelopoulos, G.; Rapantzikos, K.; Potamianos, A.; Maragos, P.; Zlatintsi, A.; Avrithis, Y.; , "*Movie summarization based on audiovisual saliency detection*," Image Processing, 2008. ICIP 2008. 15th IEEE International Conference, pp.2528-2531, 2008.

[Fang11] Y. Fang, W. Lin, B. Lee, C. Lau and C. Lin, “*Bottom-Up Saliency Detection Model Based on Amplitude Spectrum*”, MMM 2011 Part I, LNCS 6523, pp.370-380, 2011.

[Fauvet04] B. Fauvet, P. Bouthemy, P. Gros, F. Spindler, “*A geometrical key-frame selection method exploiting dominant motion estimation in video*,” in Proceedings of the 3rd International Conference on Image and Video Retrieval (CIVR ’04), pp. 419–427, Dublin, Ireland, July 2004.

[Feldmann08] Ingo Feldmann, Marcus Müller, Frederik Zilly, Ralf Tanger, Karsten Müller, Aljoacha Smolic, Peter Kauff, and Thomas Wiegand: “*HHI Test Material for 3D Video*”, ISO/IEC JTC1/SC29/WG11, MPEG08/M15413, Archamps, France, May 2008.

[Felzenszwalb04] P. F. Felzenszwalb and D. P. Huttenlocher, "*Efficient graph-based image segmentation*," International Journal on Computer Vision, vol. 59, no. 2, pp. 167–181, 2004.

[Feng08] H. Feng, Z. Shu-mao, D. Ying-shuang, "*The analysis and improvement of Apriori algorithm*", Journal of Communication and Computer, ISSN1548-7709, Volume 5, No.9 (Serial No.46), Sep. 2008.

[Ferman98] A. M. Ferman, A. M. Tekalp, R. Mehrotra, "*Effective content representation for video*," in Proc. 5th IEEE Int. Conf. Image Processing (ICIP'98), Chicago, IL, pp. 521–525, 1998.

[Fernando01] W.A.C.Fernando, C.N.Canagarajah, D.R.Bull, "*Scene change detection algorithms for content-based video indexing and retrieval*", IEE Electronics and Communication Engineering Journal, pages 117–126, Jun 2001.

[Freund97] Y. Freund, R. Schapire, "*A decision-theoretic generalization of on-line learning and an application to boosting*", Journal of Computer and System Sciences, *55*, pp. 119–139, 1997.

[Frintrop07] S. Frintrop, M. Klodt, and E. Rome, "*A real-time visual attention system using integral images*", International Conference on Computer Vision Systems, 2007.

[Fu09] X. Fu, J. Zeng, "*An Improved Histogram Based Image Sequence Retrieval Method*", Proceedings of the International Symposium on Intelligent Information Systems and Applications (IISA'09), pp. 015-018, 2009.

[Fukuchi09] Ken Fukuchi; Miyazato, K.; Kimura, A.; Takagi, S.; Yamato, J; "*Saliency-based video segmentation with graph cuts and sequentially updated priors*," Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on , pp.638-641, 2009.

[Furth95] B. Furht, S. W. Smoliar, H. Zhang, "*Video and Image Processing in Multimedia Systems*", Norwell, MA: Kluwer, 1995.

[Gao07] D. Gao and N. Vasconcelos, "*Bottom-up saliency is a discriminant process*", IEEE Conference on Computer Vision, 2007.

[Gao08] D. Gao, S. Mahadevan, and N. Vasconcelos, "*On the plausibility of the discriminant center-surround hypothesis for visual saliency*", Journal of Vision, vol. 8(7), pp. 1–18, 2008.

[Gargi00] U. Gargi, R. Kasturi, S. H. Strayer, "*Performance characterization of video shot-change detection methods*," IEEE Trans. Circuits and Systems for Video Technology, Vol.CSVT-10, No.1, pp.1-13, 2000.

[Girgensohn99] A. Girgensohn, J. Boreczky, "*Time-Constrained Keyframe Selection Technique*," in IEEE Multimedia Systems '99, IEEE Computer Society, Vol. 1, pp. 756- 761, 1999.

[Gomes99] J. Gomes, L. Darsa, B. Costa, and L. Velho, "*Morphing and Warping of Graphical Objects*", Morgan Kaufmann Publichers Inc, 1999.

[Gozalez93] R.C.Gozalez, R.E.Woods,: "*Digital image processing*", Addison Wesley, 1993.

[Green95] P. Green, "*Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*," Biometrika, vol. 82, pp. 711–732, 1995.

[Guimaraes03] S. Guimaraes, A. de Albuquerque Araujo, M. Couprie, N. Leite, "*An approach to detect video transitions based on mathematical morphology*", In Proc. IEEE ICIP '03, volume 2, pages 1021–1024, 2003.

[Guironnet07] M. Guironnet, D. Pellerin, N. Guyader, P. Ladret, "*Video summarization based on camera motion and a subjective evaluation method*", EURASIP Journal on Image and Video Processing, pp. 12 – 26, 2007.

[Gunsel98] B. Gunsel, A. M. Tekalp, "*Content-based video abstraction*", in *Proc. ICIP '98*, Chicago, IL, vol. III, pp. 128–132, 1998.

[Guo10] C. Guo and L. Zhang, "*A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression*," In IEEE Trans. Image Processing, Vol. 19(1), pp. 185-198, 2010.

[Guraya11] F.F.E. Guraya, F.A. Cheikh, "*Predictive Visual Saliency Model for Surveillance Video*", EUSIPCO, Barcelona Spain, pp. 554-558, 2011.

[Hampapur94] A. Hampapur, R. Jain, T.Weymouth, "*Digital video segmentation*," in Proc. ACM Multimedia, pp. 357–364, 1994.

[Han 99] K. Han, A. Tewfik., "*Minimization of the spurious shot boundaries using principal components decomposition and progressive nonlinear filter*", In Proc. IEEE ICIP '99, volume 4, pp. 147–151, 1999.

[Hanjalic02] A. Hanjalic, "*Shot-Boundary Detection: Unraveled and Resolved?*", IEEE Trans. Circuits and Systems for Video Technology, vol. 12, no. 2, pp. 90-105, 2002.

[Hanjalic99] A. Hanjalic, H. J. Zhang, "*An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis*", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 8, Dec. 1999.

[Hanjalic99] Hanjalic A., Lagendijk R.L., Biemond J.: "*Automated high-level movie segmentation for advanced video-retrieval systems*", IEEE Transactions on Circuits Systems Video Technology vol. 9(4), pp.580–588, 1999.

[Harel07] J. Harel, C. Koch, and P. Perona. "*Graph-based visual saliency*", Advances in Neural Information Processing Systems, pp. 545- 554, 2007.

[Hendrickson00] B. Hendrickson, T. G. Kolda, "*Graph partitioning models for parallel computing*", Parallel Computing Journal, Nr. 26, pp. 1519-1534, 2000.

[Hou07] X. Hou and L. Zhang, "*Saliency detection: A spectral residual approach*", IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[Hu04] Y. Hu, X. Xie, W.-Y. Ma, L.-T. Chia, and D. Rajan, "*Salient region detection using weighted feature maps based on the human visual attention model*", Pacific Rim Conference on Multimedia, 2004.

[Hua09] G. Hua *et al.*, "*Efficient Scale-space Spatiotemporal Saliency Tracking for Distortion-Free Video Retargeting*," in *ACCV*, 2009.

[Huang98] Huang J., Liu Z., Wang Y.: "*Integration of audio and visual information for content-based video segmentation*", IEEE proceeding on International Conference on Image Processing, pp. 526–530, 1998.

[Itti01] L. Itti and C. Koch, "*Computational modeling of visual attention*", Nature Reviews Neuroscience, vol. 2(3), 2001.

[Itti98] L. Itti, C. Koch, E. Niebur, "*A model of saliency-based visual attention for rapid scene analysis*", IEEE TPAMI, 20(11), pp. 1254–1259, 1998.

[Jaffre04] G. Jaffre, P. Joly, and S. Haidar, "*The Samova shot boundary detection for TRECVID evaluation 2004*," in Proc. TRECVID Workshop, 2004.

[Jaggi10] M. Jaggi and M. Sulovsky, "*A simple algorithm for nuclear norm regularized* *problems*”, in International *Conference* on Machine Learning (ICML), 2010.

[Jain88] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, New York, 1988.

[Joly07] Philippe Joly, Jenny Benois-Pineau, Ewa Kijak, Georges Quénot. *The ARGOS campaign: Evaluation of Video Analysis Tools.* Dans : *Signal Processing : Image Communication*, Elsevier, Vol. 22 N. 7-8, p. 705-717, 2007.

[Kelm09] P. Kelm, S. Schmiedeke, T. Sikora, “*Feature-Based Video Key Frame Extraction for Low Quality Video Sequences*”, 10th Workshop on Image Analysis for Multimedia Interactive Services, pp.25-28, 2009.

[Kernighan70] B. Kernighan, S. Lin, "*An Efficient Heuristic Procedure for Partitioning Graphs*," Bell System Technical Journal, Vol. 49, pp. 291-307, Feb. 1970.

[Kim11] W. Kim, C. Jung, and C. Kim, “*Spatiotemporal saliency detection and its applications in static and dynamic scenes*,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 4, pp. 446–456, 2011.

[Kolmogorov06] S. Kolmogorov and C. Rother. “*Comparison of energy minimization algorithms for highly connected graphs*”, In *ECCV*, pages II: 1–15, 2006.

[Lee07] J. J. Lee and G. Y. Kim. “*Robust estimation of camera homography using fuzzy RANSAC*”, International Conference on Computational Science and Its Applications, 2007.

[Lee07] J. J. Lee and G. Y. Kim. “*Robust estimation of camera homography using fuzzy RANSAC*”, International Conference on Computational Science and Its Applications, 2007.

[Lefevre07] S. Lefevre, N. Vincent, "*Efficient and robust shot change detection*," Journal of Real-Time Image Processing, vol. 2, pp. 23-34, 2007.

[Leszczuk02] M. Leszczuk, Z. Papir, “*Accuracy versus speed tradeoff in detecting of shots in video content for abstracting digital video libraries*,” in *Lecture Notes In Computer Science*. London, U.K.: Springer-Verlag, vol. 2515, pp. 176–189, 2002.

[Lezama11] Lezama, J.; Alahari, K.; Sivic, J.; Laptev, I., "Track to the future: *Spatio-temporal video segmentation with long-range motion cues*," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference, pp.3369-3376, 20-25 June 2011.
*www.klab.caltech.edu/~harel/share/**gbvs**.php*

[Li01] Li, Y., Zhang, T., Tretter, ”*An overview of video abstraction techniques*”, Tech. Rep. HP-2001-191, HP Laboratory, July 2001.

[Li03] W.K. Li , S.H. Lai, “*Storage and retrieval for media databases,*” *Proc. SPIE*, vol. 5021, pp. 264-271, Jan. 2003.

[Li03] Y. Li, S. Narayanan, C.-C.J. Kuo., “*Movie Content Analysis, Indexing and Skimming via Multimodal Information*”, Video Mining, Chapter 5, Eds. Kluwer Academic Publishers, 2003.

[Li04] X.Li, Y.Lu, D.Zhao, W.Gao, S.Ma, “*Enhanced direct coding for bipredictive pictures*”, Proc. IEEE ISCAS, vol.3, pp.785- 788, 2004.

[Li08] H. Li, K. N. Ngan, “*Saliency model-based face segmentation and tracking in head-and-shoulder video sequences*,” Journal of Visual Communication Image Representation, vol. 19, pp. 320–333, 2008.

[Lienhart01] Lienhart, R, “*Reliable transition detection in videos: a survey and practitioner’s guide*”, Int. Journal of Image and Graphics 1(3), pp. 469–486, 2001.

[Lienhart97] R. Lienhart, S. Pfeiffer, W. Effelsberg, “*Video Abstracting*”, Communications of the ACM, pp 1 -12, 1997.

[Lienhart98] R. Lienhart, W. Effelsberg, and R. Jain.: “*A systematic analysis of various methods to compare video sequences*”, Proc. SPIE 3312, Storage and Retrieval for Image and Video Databases, pp. 271-282. 1998.

[Lienhart99] R. Lienhart, “*Comparison of automatic shot boundary detection algorithms*,” in *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Databases VII*, vol. 3656, pp. 290–301, 1999.

[Lienhart99] Lienhart R., Pfeiffer S., Effelsberg W.: “*Scene determination based on video and audio features*”, IEEE proceeding on International Conference on Multimedia Computing and Systems, pp.685–690, 1999.

[Liu03] T. Liu, M. Zhang, F.H. Qi, “*A novel video key-frame-extraction algorithm based on perceived motion energy model*”, IEEE Trans. Circuits and Systems for Video Technology, pp. 1006-1013, 2003.

[Liu06] [6]. F. Liu, M. Gleicher, “*Region enhanced scale-invariant saliency detection*,” in Proc. IEEE ICME, pp. 1477–1480, 2006.

[Liu93] B. Liu and A. Zaccarin, “*New fast algorithms for the estimation of block motion vectors*,” IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 2, pp. 148–157, 1993.

[Lowe04] Lowe, D., “*Distinctive image features from scale-invariant keypoints*”, International Journal of Computer Vision, pp. 1-28, 2004.

[Lu10] T. Lu, Z. Yuan *et al*., “*Video Retargeting with nonlinear spatio-temporal saliency fusion*,” in *ICIP*, pp: 1801 – 1804, 2010.

[Luo09] J. Luo, C. Papin, K. Costello, “*Towards Extracting Semantically Meaningful Key Frames from Personal Video Clips: From Humans to Computers*”, IEEE Trans. Circuits Syst. Video Techn. 19 (2), pp. 289–301, 2009.

[Lupatini98] G. Lupatini, C. Saraceno, R. Leonardi, ‘”*Scene Break Detection: A Comparison”*, Research Issues in Data Engineering, Workshop on Continuos Media Databases and Applications, pages. 34- 41, 1998.

[Ma11] Lin Ma; Songnan Li; Ngan, K.N, "*Motion trajectory based visual saliency for video quality assessment*," Image Processing (ICIP), 18th IEEE International Conference, pp.233-236, 2011.

[Malik01] J. Malik, S. Belongie, T. Leung, and J. Shi, “*Contour and texture analysis for image segmentation*”, International Journal of Computer Vision, vol. 43(1), pp. 7–27, 2001.

[Mas03] J. Mas, G. Fernandez, "*Video Shot Boundary Detection Based on Color Histogram*", TRECVID Workshop 2003, 2003.

[Matsumoto06] K.Matsumoto, M.Naito, K.Hoashi, F.Sugaya, “*SVM-Based Shot Boundary Detection with a Novel Feature*,” In Proc. IEEE Int. Conf. Multimedia and Expo, pp.1837–1840, 2006.

[Meng95] J. Meng, Y. Juan, S.F. Chang, “*Scene change detection in a MPEG compressed video sequence*”. *SPIE*, 2419, pp. 14–25, February 1995.

[Mills92] M. Mills, “A magnifier tool for video data”, *Proc. of ACM Human Computer*

*Interface*, pp. 93-98, May 1992.

[Muratov11] Muratov, O.; Zontone, P.; Boato, G.; De Natale, F.G.B.; "*A segment-based image saliency detection*," Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on , pp.1217-1220, 22-27 May 2011.

[Nagasaka91]. Nagasaka A, Tanaka Y, "*Automatic video indexing and fullvideo search for object appearances*", Proc 2nd Working Conf Visual Database Systems, pp. 119-133, 1991.

[Nagasaka92] A. Nagasaka, Y. Tanaka, "*Automatic video indexing and full-video search for object appearances*," in Visual Database Systems II, E. Knuth and L. M.Wegner, Eds. Amsterdam, The Netherlands: North-Holland, pp. 113–127, 1992.

[Ngo02] Ngo C.W., Zhang H.J: "*Motion-based video representation for scene change detection*", International Journal on Computer Vision, vol. 50(2), pp.127–142, 2002.

[Ngo03] C.-W. Ngo, "*A robust dissolve detector by support vector machine*," in *Proc. ACM Multimedia*, pp. 283–286, 2003.

[Ogale06] A.S.Ogale and Y.Aloimonos, "*A roadmap to the integration of early visual modules*," International Journal on Computer Vision: Special issue on early cognitive vision, 2006.

[Oliva03] A. Oliva, A. Torralba, M. Castelhano, and J. Henderson, "*Top-down control of visual attention in object detection*", In Proceedings of International Conference on Image Processing, pp. 253–256, 2003.

[Osian04] M. Osian, L. V. Gool, "*Video shot characterization*", Machine Vision Application, vol. 15, pp. 172–177, 2004.

[Otsuji91] K. Otsuji, Y. Tonomura, Y. Ohba, "*Video browsing using brightness data*", SPIE Visual Communications and Image Processing, 1606: pp 980–989, 1991.

[Otsuji94] K. Otsuji, Y. Tonomura, "*Projection detecting filter for video cut detection. Multimedia Systems*", pp.205 –210, 1994.

[Panigrahy08] R. Panigrahy, "*An improved algorithm finding nearest neighbor using kd-trees*", In Proceedings of the 8th Latin American conference on Theoretical informatics, LATIN'08, pp. 387–398, 2008.

[Park05] M. Park, R. Park, S. Lee, "*Efficient shot boundary detection for action movies using blockwise motion-based features*", In International Symposium on Visual Computing (ISVC '05), pp. 478–485, 2005.

[Pass96] G. Pass, R. Zabih, and J. Miller. "*Comparing Images Using Color Coherence Vectors*", Proc. ACM Multimedia 96, Boston, MA, USA, pp. 65-73, No5. 1996.

[Pei00] S. Pei, Y. Chou, "*Efficient and effective wipe detection in mpeg compressed video based on the macroblock information*", In Proc. IEEE ICIP '00, volume 3, pp. 953–956, 2000.

[Petersohn08]. Christian Petersohn. "*Improving scene detection by using gradual shot transitions as cues from film grammar*", In Proc. IS&T/SPIE Electronic Imaging 2008, Multimedia Content Access: Algorithms and Systems II, 2008.

[Pfeiffer96] S. Pfeiffer, R. Lienhart, S. Fischer, W. Effelsberg, "*Abstracting digital movies automatically*", Journal of Visual Communication and Image Representation, vol. 7, no. 4, pp. 345-353, Dec. 1996.

[Porter00] S.V. Porter, M. Mirmehdi, B.T. Thomas, "*Video cut detection using frequency* *domain correlation*”, 15th International Conference on Pattern Recognition, pages 413–416, Barcelona, Spain 2000.

[Rasheed03] Rasheed Z., Shah M.: “*Scene detection in Hollywood movies and TV shows*”, IEEE proceeding on Computer Vision and Pattern Recognition, pp.343–348, 2003.

[Rasheed05] Z. Rasheed, M. Shah, “*Detection and Representation of Scenes in Videos*”, IEEE Transactions on Multimedia, Vol. 7, No 6, pp. 1097-1105, Dec. 2005.

[Rasheed05] Z. Rasheed, Y. Sheikh, and M. Shah, “*On the use of computable features for film classification*”, IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 1, pp. 52–64, Jan. 2005.

[Robles04] O. Robles, P. Toharia, A. Rodriguez, L. Pastor, “*Using adaptive thresholds for automatic video cut detection*”, In Proceedings of TRECVID 2004, 2004.

[Rother04] C. Rother, S. Kolmogorov, and A. Blake, “*Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts*”, Proc. ACM SIGGRAPH, pp. 309-314, 2004.

[Sakarya10] Sakarya U., Telatar Z., *Video scene detection using graph-based representations*, International Journal on Signal Processing, vol.25, pp.774-783, 2010.

[Seo09] H.J. Seo and P. Milanfar, “*Static and Space-Time Visual Saliency Detection by Self-Resemblance*,” Journal of Computer. Vision, vol. 9, no. 12, no. 15, pp. 1-27, 2009.

[Sethi95] I. K. Sethi, N. Patel, “*A statistical approach to scene change detection*,” in Proc. IS&T/SPIE Conf. Storage and Retrieval for Image and Video Databases III, vol. SPIE 2420, pp. 329–338, 1995.

[Sevilmis08] Tarkan Sevilmis, Muhammet Bastan:”*Automatic detection of salient objects and spatial relations in videos for a video database system*”, Image and Vision Computing, Volume 26, pp. 1384–1396, 2008.

[Shahraray95] B. Shahraray, ‘‘*Scene change detection and content-based sampling of video sequences*,’’ in Digital Video Compression: Algorithms and Technologies, Proc. SPIE 2419, pp. 2–13,1995.

[Shahraray95] B. Shahraray, ‘‘*Scene change detection and content-based sampling of video sequences*,’’ in Digital Video Compression: Algorithms and Technologies, Proc. SPIE 2419, pp. 2–13,1995.

[Shi00] J. Shi, J. Malik, “*Normalized Cuts and Image Segmentation*”, Proc. Computer Vision and Pattern Recognition, pp. 731-737, June 2000.

[Smith09] Smith, B.M.; Li Zhang; Hailin Jin; "*Stereo matching with nonparametric smoothness priors in feature space*," Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp.485-492, 20-25 June 2009.

[Smoliar94] S. W. Smoliar, H.-J. Zhang, "*Content-based video indexing and retrieval*", IEEE Multimedia, vol. 1, pp.89-95, 1994.

[Stefanidis00] A. Stefanidis, P. Partsinevelos, P. Agouris, P. Doucette, “*Summarizing video datasets in the spatiotemporal domain*”, Proc. of 11th International Workshop on Database and Expert Systems Applications, pp. 906-912, Sep. 2000.

[Su11] Hsiao-Hang Su , Tse-Wei Chen , Chieh-Chi Kao , Winston H. Hsu , Shao-Yi Chien, “*Scenic photo quality assessment with bag of aesthetics-preserving features*”, Proceedings of the 19th ACM international conference on Multimedia, 2011.

[Sugihara94] Sugihara, K., “*Robust gift wrapping for the three-dimensional convex hull*”, Journal of Computer Systems and Science, vol. 49, pp. 391- 407, 1994.

[Suh03] B. Suh, H. Ling, B. Bederson, and D. Jacobs, “*Thumbnail cropping and its effectiveness*”, In ACM User Interface Software and Technology, 2003.

[Sun00] X. D. Sun, M. S. Kankanhalli, “*Video summarization using R-sequences*”, *Real-time Imaging*, pp. 449-459, Dec. 2000.

[Sun08] Z. Sun, K. Jia, H. Chen, “*Video Key Frame Extraction Based on Spatial-Temporal Color Distribution*,” in Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 196–199, 2008.

[Taniguchi95] Y. Taniguchi, “*An intuitive and efficient access interface to real-time incoming video based on automatic indexing*”, Proc. of ACM Multimedia, pp. 25-33, Nov. 1995.

[Tapu10] R. Tapu, T. Zaharia and F. Preteux, “*A scale-space filtering-based shot detection algorithm*”, IEEE 26-th Convention of Electrical and Electronics Engineers in Israel (BDI), ISBN: 978-1-4244-8682-3, pp.919-923, Eilat, Israel, 2010.

[Tapu11] R. Tapu and T. Zaharia, „*Scene change detection with temporally constrained clustering*”, 3rd International Conference on Future Computer and Communications – ICFCC 2011 (ISI), ISBN: 978-0-7918-5971-1, pp. 71-76, Iasi, Romania, 3-5 June 2011.

[Tapu11] R. Tapu and T. Zaharia, “*A Complete Framework for Temporal Video Segmentation*”, The 1st International Conference on Consumers Electronics, ICCE-2011, ISBN: 978-1-4577-0234-1, pp. 156-160, Berlin, Germany, 2011(BDI).

[Tapu11] R. Tapu and T. Zaharia, “*Automatic Multilevel Temporal Video Structuring*”, Fifth IEEE International Conference on Semantic Computing (IEEE International Workshop on Semantic Multimedia), ICSC-2011, ISBN: 978-0-7695-4492-2, pp. 387-394, Palo Alto, California, USA, 2011 (BDI).

[Tapu11] R. Tapu and T. Zaharia, “*High Level Video Temporal Segmentation*”, 7th International Symposium on Visual Computing, ISVC-2011, ISBN: 978-3-642-24027-0, Part I, LNCS 6938, pp. 226–237, Las Vegas, Nevada, USA, 2011 (BDI).

[Tavanapong04] Tavanapong W., Zhou J.,: “*Shot clustering techniques for story browsing*”, IEEE Transaction on Multimedia, vol.6(4), pp. 517–526, 2004.

[Truong00] B. T. Truong, C. Dorai, S. Venkatesh, “*Improved fade and dissolve detection for reliable video segmentation*”, in Proc. IEEE Int. Conf. Image Processing (ICIP 2000), vol. 3, pp. 961-964, 2000.

[Truong03] Truong B.T., Venkatesh S., Dorai C.: “*Scene extraction in motion picture*”, IEEE Transactions on Circuits Systems and Video Technology vol. 13(1), pp.5–15, 2003.

[Truong07] B. Truong, S. Venkatesh, “*Video abstraction: A systematic review and classification*”, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), v.3 n.1, p.3-es, February 2007.

[Urhan07] O. Urhan, K.M. Gullu, S. Erturk, “*Shot-cut detection for b&w archive films using best-fitting kernel*”, Int. Journal of Electronics and Communications, 61(7):463–468, 2007.

[Viola01] P. Viola and M. Jones, “Rapid *Object Detection Using a Boosted Cascade of*

*Simple Features*", in Computer Vision and Pattern Recognition (CVPR), pp. 511-518, 2001.

[Volkmer04] T. Volkmer, S. Tahaghoghi, H. Williams, "*Gradual Transition Detection Using Average Frame Similarity*", Computer Vision and Pattern Recognition Workshop 2004, pp. 138 – 146, 2004.

[Wang03] S. Wang, J. Siskind, "*Image Segmentation with Ratio Cut*," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 6, pp. 675-690, June 2003.

[Wang08] Z. Wang and B. Li, "*A two-stage approach to saliency detection in images*", IEEE Conference on Acoustics, Speech and Signal Processing, 2008.

[Wolf96] W. Wolf, "*Key frame selection by motion analysis*", *ICASSP'96*, vol. 2, pp. 1228-1231, 1996.

[Wu11] Yang Wu, Nan Ning Zheng, ZeJian Yuan, HuaiZu Jiang and Tie Liu, "*Detection of salient objects with focused attention based on spatial and temporal coherence*", Chinese Science Bulletin Volume 56, Number 10, pp. 1055-1062, 2011.

[Xie04] Xie L., Xu P.: "*Structure analysis of soccer video with domain knowledge and hidden Markov models*", Journal on Pattern Recognition Letter, vol. 25(7), pp.767–775, 2004.

[Xie11] Yulin Xie; Huchuan Lu, "*Visual saliency detection based on Bayesian model*," Image Processing (ICIP), 2011 18th IEEE International Conference on , pp.645-648, 11-14 Sept. 2011.

[Xue12] Yawen Xue, Xiaojie Guo, Xiaochun Cao, "*Motion saliency detection using low-rank and sparse decomposition*", International Conference on Acoustic, Speech and Signal Processing, 2012.

[Yeo95] B.-L. Yeo, B. Liu, "*Rapid scene analysis on compressed video*," IEEE Trans. Circuits Systems Video Technology, vol. 5, pp. 533–544, Dec. 1995.

[Yeung95] M. M. Yeung, B. Liu, "*Efficient matching and clustering of video shots*", *Proc. of IEEE ICIP'95*, vol. I, pp. 338-341, 1995.

[Yeung98] Yeung M., Yeo B-L.: "*Segmentation of video by clustering and graph analysis*", Journal of Computer Vision and Image Understanding 71(1), pp. 94–109, 1998.

[Ying-Li05] Tian, Ying-Li; Hampapur, Arun; , "*Robust Salient Motion Detection with Complex Background for Real-Time Video Surveillance*," Application of Computer Vision WACV/MOTIONS Volume 1. Seventh IEEE Workshops, vol.2, pp.30-35, 2005.

[Yu04] X.-D. Yu, L. Wang, Q. Tian, P. Xue, "*Multi-level video representation with application to keyframe extraction*," The 10th International Multi-Media Modeling Conference, Australia, pp. 568 – 575, 2004.

[Yu97] H. Yu, G. Bozdagi, S. Harrington, "*Feature-based Hierarchical Video Segmentation*", IEEE International Conference on Image Processing (ICIP'97), Vol. 2, pp. 498-501, 1997.

[Yuan07] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, B. Zhang, "*A formal study of shot boundary detection*", IEEE Trans. Circuits Systems Video Technol., vol. 17, no. 2, pp. 168–186, Jun. 2007.

[Zabih94] Ramin Zabih and John Wood, "*Non_parametric local transforms for computing visual correspondence*", in European Conference on Computer Vision, number 801, in LNCS, pages 151-158, 1994

[Zabih95] R. Zabih, J. Miller, K. Mai, ‘‘*A feature-based algorithm for detecting and classifying scene breaks,*’’ Proc. ACM Multimedia *95*, pp. 189–200, San Francisco, 1995.

[Zhai06] Zhai Y., Shah M.: “*Video scene segmentation using Markov Chain Monte Carlo*”, IEEE Transaction on Multimedia, vol. 8(4), pp.686–697, 2006.

[Zhai06] Y. Zhai and M. Shah, “*Visual attention detection in video sequences using spatiotemporal cues*”, In ACM Multimedia, pages 815–824, 2006.

[Zhai08] G. Zhai, Q. Chen, X. Yang, and W. Zhang, “*Scalable Visual Sensitivity Profile Estimation*,” IEEE International Conference on Acoustics, Speech, and Signal Processing 2008, pp. 873-876, March, 2008.

[Zhang05] M.-L. Zhang, Z.-H Zhou, “*A k-Nearest Neighbor Based Algorithm for Multi-label Classification*”, 1st IEEE International Conference on Granular Computing, 2005.

[Zhang09] L. Zhang, M. Tong, and G. Cottrell, “*SUNDAy: Saliency using natural statistics for dynamic analysis of scenes*”, International Conference on Computer Vision, 2009.

[Zhang10] Y. Zhang, G. Jiang, M. Yu, and K. Chen, “*Stereoscopic visual attention model for 3-D video,*” in Proc. Multimedia Modeling, pp. 314–324, 2010.

[Zhang93] H. J. Zhang, A. Kankanhalli, S. W. Smoliar, ‘‘*Automatic partitioning of full-motion video,*’’ Multimedia Systems no. 1, pp. 10–28, 1993.

[Zhang99] H. Zhang, J. Wu, D. Zhong, S. W. Smoliar, “*An integrated system for content-based video retrieval and browsing*”, Pattern Recognit., vol. 30, no. 4, pp. 643–658, 1999.

[Zhao07] Zhao Y.J., Wang T.: “*Scene segmentation and categorization using NCuts*”, IEEE proceeding on Computer Vision and Pattern Recognition, pp. 343–348, 2007.

[Zheng05] W. Zheng, J. Yuan, H. Wang, F. Lin, B. Zhang, “*A novel shot boundary detection framework,*” in Proc. SPIE Vis. Commun. Image Processing, vol. 5960, pp. 410–420, 2005.

[Zhu09] S. Zhu, Y. Liu, “*Automatic scene detection for advanced story retrieva*l”, Expert Systems with Applications, 36(3, Part 2), pp. 5976 – 5986, 2009.

[Zhu09] Zhu S., Liu Y.,: “*Scene segmentation and semantic representation using a novel scheme*”, Multimedia Tools Application, vol.42, pp.183–205, 2009.

[Zhuang98] Y. Zhuang, Y. Rui, T. Huang, S. Mehrotra, "*Adaptive Key Frame Extraction Using Unsupervised Clustering*," in Proc. ICIP '98, Vol. I, pp. 866-870, 1998.

[Žiberna04] Žiberna, A., Kejžar, N., and Golob P., 2004, “*A Comparison of Different Approaches to Hierarchical Clustering of Ordinal Data*,” Metodološki zvezki, Vol. 1, No. 1, 2004, 57-73.

# Abstract

Recent advances in telecommunications, collaborated with the development of image and video processing and acquisition devices has lead to a spectacular growth of the amount of the visual content data (still images, video streams, 2D graphical elements, 3D models...) stored, transmitted and exchanged over Internet. Within this context, elaborating efficient tools to access, browse and retrieve video content has become a crucial challenge.

From the spatio-temporal structural point of view, a digital video can be decomposed into four different levels of detail, corresponding to scenes/chapters, shots, keyframes and objects. The detection of the structural elements represents a key and mandatory stage that needs to be performed prior to any effective description/classification of video documents. In this thesis, we notably tackle this issue, and propose solutions for the detection of each of the above-mentioned structural elements involved.

In Chapter 2 we introduce and validate a novel shot boundary detection algorithm able to identify abrupt (*i.e.,* cuts) and gradual transitions (*i.e.*, fades, wipes…). The technique is based on an enhanced graph partition model, combined with a multi-resolution analysis and a non-linear filtering operation. The global computational complexity is reduced by implementing a two-pass approach strategy.

In Chapter 3 the video abstraction problem is considered. In our case, we have developed a keyframe representation system (based on a leap-extraction algorithm) that extracts a variable number of images from each detected shot, depending on the visual content variation.

The Chapter 4 deals with the issue of high level semantic segmentation into scenes. Here, a novel scene/DVD chapter detection method is introduced and validated. Spatio-temporal coherent shots are clustered into the same scene based on a set of temporal constraints, adaptive thresholds and with the help of a new concept - neutralized shots. Concerning the keyframe visual similarity involved in the above-described process, we have considered two different approaches, based on chi-square distance between HSV color histograms and the number of matched interest points extracted based on SIFT descriptors.

Chapter 5 considers the issue of object detection and segmentation. Here we introduce a novel spatio-temporal visual saliency (STVS) system based on: region contrast, interest points correspondence, geometric transforms, motion classes' estimation (using agglomerative clustering) and regions temporal consistency. The proposed technique is extended on 3D videos by representing the stereoscopic perception as a 2D video and its associated depth. The technique is robust to complex background distracting motions and does not require any initial knowledge about the object size or shape.