Video transition identification based on 2𝔻 image analysis

Silvio Jamil Ferzoli Guimarães

14 March 2003

Video transition identification based on $2\mathbb{D}$ image analysis

This text corresponds to the Thesis which is presented by Silvio Jamil Ferzoli Guimarães and appreciates by Jury.

Belo Horizonte-Brasil, 14 March 2003 .

Prof. Dr. Arnaldo de Albuquerque Araújo
(Advisor)

Thesis presented to the Departamento de Ciência da Computação, UFMG, as a partial requeriment to obtain the title of PhD in Computer Science.

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

Video transition identification based on $2\mathbb{D}$ image analysis

Silvio Jamil Ferzoli Guimarães

14 March 2003

Jury:

Prof. Dr. Arnaldo de Albuquerque Araújo (Advisor)
DCC-UFMG

Prof. Dr. Michel Couprie (Co-advisor)
ESIEE-FRANCE

Prof. Dr. Neucimar Jerônimo Leite (Co-advisor)
IC-UNICAMP

Prof. Dr. Roberto de Alencar Lotufo
FEEC-UNICAMP

Prof. Dr. Jacques Facon
PUC-PR

Prof. Dr. Mário Fernando Montenegro Campos
DCC-UFMG

Prof Dr. Rodrigo Lima Carceroni
DCC-UFMG

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

Identificação de transições em vídeo baseada na análise de imagens $2\mathbb{D}$

Silvio Jamil Ferzoli Guimarães

14 de Março de 2003

Banca:

Prof. Dr. Arnaldo de Albuquerque Araújo (Orientador)
DCC-UFMG

Prof. Dr. Michel Couprie (Co-orientador)
ESIEE-France

Prof. Dr. Neucimar Jerônimo Leite (Co-orientador)
IC-UNICAMP

Prof. Dr. Roberto de Alencar Lotufo
FEEC-UNICAMP

Prof. Dr. Jacques Facon
PUC-PR

Prof. Dr. Mário Fernando Campos Montenegro
DCC-UFMG

Prof. Dr. Rodrigo Lima Carceroni
DCC-UFMG

Université de Marne-la-Vallée

Identification de transitions dans des séquences d'images vidéo basée sur l'analyse d'images $2\mathbb{D}$

Silvio Jamil Ferzoli Guimarães

14 Mars 2003

Jury:

Prof. Dr. Neucimar Jerônimo Leite (Président)
IC-UNICAMP-Brésil

Prof. Dr. Arnaldo de Albuquerque Araújo (Directeur)
DCC-UFMG-Brésil

Prof. Dr. Michel Couprie (Co-directeur)
ESIEE-France

Prof. Dr. Roberto de Alencar Lotufo (Rapporteur)
FEEC-UNICAMP-Brésil

Prof. Dr. Sylvie Philipp-Foliguet (Rapporteur)
ENSEA-France

Aos meus pais, Luiz e Saada, e à minha amada esposa, Tatiane.

Agradecimentos (in portuguese and in french)

À Deus, por ter me dado saúde e coragem para enfrentar este desafio.

Aos meus pais, Luiz e Saada, pelas imensas horas de conforto e sabedoria a mim dispensados dando-me força para continuidade.

Ao Prof. Arnaldo e ao Prof. Neucimar, pela ajuda, companheirismo e pelas brilhantes idéias que foram determinantes para o término deste trabalho. Não podendo esquecer também da ajuda referente as infinitas correções de artigos.

Je remercie à Michel Couprie pour tant m'aider pendant mon sejour en France, avec beaucoup de patience et compréhension car je ne suis pas expert de la langue française. Je le remercie aussi pour ses idées et pour les discussions toujours intéressantes.

Ao grupo de trabalho do NPDI do DCC-UFMG, em especial ao Paulo e ao Camillo. E também, ao grupo de trabalho do LA^{2}SI-ESIEE, em especial à Yukiko, ao Nivando, ao Christophe e ao Marco pelas diversas discussões que só levaram ao crescimento pessoal e profissional.

Aos meus amigos e colegas do DCC, em especial ao pessoal da minha turma, Lucila, Mark, Fátima, Maria de Lourdes e, de novo, Paulo.

Aos meus pais, ao CNPq e à CAPES pelo imprescindível apoio financeiro durante toda a minha jornada .

"Eu não me envergonho de corrigir e mudar minhas opiniões, porque não me envergonho de raciocinar e aprender."

(Alexandre Herculano)

Abstract

The video segmentation problem consists in the identification of the boundary between consecutive shots in a video sequence. The common approach to solve this problem is based on the computation of dissimilarity measures between frames. In this work, the video segmentation problem is transformed into a problem of pattern detection, where each video event is represented by a different pattern on a $2\mathbb{D}$ spatio-temporal image, called visual rhythm. To cope with this problem, we consider basically morphological and topological tools that we use in order to identify the specific patterns that are related to video events such as cuts, fades, dissolves and flashes. To compare different methods we define two new measures, the robustness and the gamma measures. In general, the proposed methods present the quality measures better than the other methods used to comparison.

Résumé

Le problème de la segmentation de séquences d'images vidéos est principalement associé au changement de plan. L'approche courante pour résoudre ce problème est basée sur le calcul de mesures de dissimilarités entre images. Dans ce travail, le problème de la segmentation de séquences vidéo est transformé en un problème de détection de motifs, où chaque évènement dans la vidéo est representé par un motif différent sur une image $2\mathbb{D}$, apellée rythme visuel. Cette image est obtenue par une transformation spécifique de la vidéo. Pour traiter ce problème, nous allons considérer principalement des outils morphologiques et topologiques. Nous montrons commment identifier grâce à ces outils, les motifs spécifiques qui sont associés à coupures, fondus et fondus enchaînés, ainsi qu'aux flashs. Dans l'ensemble, les méthodes proposées dans cette thèse obtiennent des indices de qualité meilleurs que les autres méthodes auxquelles nous les avons comparées.

Resumo

O problema de segmentação em vídeo consiste na identificação dos limites entre as tomadas em um video. A abordagem clássica para resolver este problema é baseada no cálculo de medidas de dissimilaridade entre quadros. Neste trabalho, o problema de segmentação em vídeo é transformado em um problema de detecção de padrões, onde cada evento de vídeo é transformado em diferentes padrões em um imagem espaço-temporal $2\mathbb{D}$, chamada ritmo visual. Para tratar este problema, nós consideramos basicamente ferramentas morfológicas e topológicas com o objetivo de identificar os padrões específicos que são relacionados à eventos do vídeo, como cortes, fades, dissolves e flash. Para comparar os diferentes métodos, nós definimos duas novas medidas de dissimilaridade, a robusteza e a medida gama, que relacionam as medidas básicas de qualidade com um família de limiares. Os resultados obtidos a partir dos métodos propostos, definidos em termos de medidas de qualidade, são melhores que os resultados dos outros métodos usados como critério de comparação.

Contents

List of Definitions

List of Figures

Chapter 1

Introduction

Traditionally, visual information has been stored analogically and indexed manually. Nowadays, due to the improvements on digitalization and compression technologies, database systems are used to store images and videos, together with their meta-data and associated taxonomy. Unfortunately, these systems are still very costly.

Meta-data include bibliography information, capture conditions, compression parameters, etc. The taxonomy is a hierarchy of subjective classes (people, nature, news) used to organize image/video in different subjects, such as humor, politic, people, etc. A good selection of meta-data and taxonomy must incorporate special features of the application that, in general, represent the first step to create and use a large image/video database. Obviously, there are many constraints related to the use of these indexes: manual annotation represents a big problem in large databases (time-consuming); the domain of the application and the personal knowledge bias the choice of these indexes, etc. Unfortunately, the existing indexes are always limited in the sense of capturing the salient content of an image ("a picture is worth a thousand words") [11, 14, 44, 3, 7, 20].

Multimedia content analysis and content-based indexing represent together a promising direction for the above methodology. Many systems that consider image/video queries based on content have been developed [29, 43, 15, 70, 8, 48, 6]. In the last few years, progress in image/video searching tools for large database systems has been steady. To build a query in these tools, one can use sketches, selection of visual features (color, texture, shape and motion), examples and/or temporal/spatial features.

Concerning video, the indexing problem becomes much more complex, because it

involves the identification and understanding of fundamental units, such as, scene and shots. Fundamental units can be semantically and physically sub-divided. To decrease the number of items to index, we may consider the semantic units, the scenes, instead of consider the physical units (shots). However, due to the non-structured format, the size and the general content of a video, this indexing is non trivial [58, 15, 70]. Therefore, automatic methods for video indexing are extremely relevant in modern applications, in which speed and precision of queries are required. An example of the use of this index is video browsing, where it is necessary to segment a video in fundamental units without knowing the nature and type of the video [60]. Another video problem is the detection of specific events, such as the identification of the instant in which a predator attacks its "prey" in a documentary video [38, 37].

The *World Wide Web* presents itself as an enormous repository of information, where thousands of documents are added, replaced or deleted daily. Among these documents, there are large image/video collections that are produced by satellite systems, scientific experiments, biomedical systems, etc.

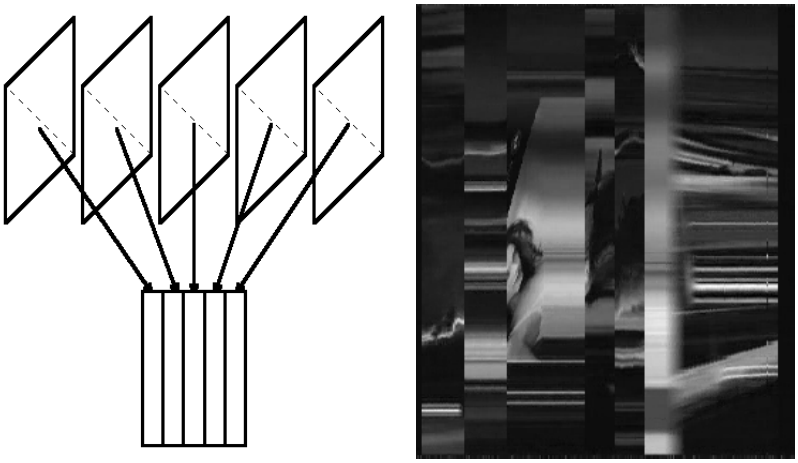
Thus, the need of efficient systems to process and index the information is unquestionable, mainly if we are interested in information retrieval. The most important problem is associated with the exponential increase of the *Internet*, and the consequent increase in the number of duplicated documents. According to [16], 46% of all textual documents have a duplicate. Considering multimedia documents, the problem is more complicated because the same documents are spread in several machines around the world to facilitate *download*, mainly, due to their size.

The need of video contents analysis tools has led many research groups to tackle this matter, for which different tools have been proposed. Generally, these tools present the following processing steps [72]:

- video parsing - temporal segmentation that involves identification of different shots, and camera operation analysis;
- summarization - simplification of video content to facilitate the video browsing;
- classification and indexing - to facilitate the video retrieval.

Differently from the main works presented in literature, which use dissimilarity measure concepts for video parsing (as it will be discussed in Chapter 3), in this work, we will

consider the problem of video analysis as an image segmentation task. In other words, the problem of video segmentation will be transformed into a 2𝔻 image segmentation problem, in which the video content is represented by a 2𝔻 image, called visual rhythm (VR). Informally, each frame is transformed into a vertical line on the VR image, as illustrated in Fig. 1.1(a). Taking advantage of the fact that each event is represented by a specific pattern in this image, different methods for video analysis based on 2𝔻 image analysis have been proposed in the literature [71, 17, 57, 36].



(a) (b)

Figure 1.1: Video transformation: (a) simplification of the video content by transformation of each frame into a column on VR; (b) a real VR, obtained by the principal diagonal sub-sampling (Chapter 4).

1.1 Our contribution

The general aspects of the video segmentation problem based on 2𝔻 image analysis has been already studied in other works, which involve the use of: statistical methods [17], texture analysis with Markov models [57], texture analysis with Fourier Transforms [1] and Hough Transforms [42]. However, the computational cost of these approaches is high, and most important, their results are not satisfactory. In this work we propose new methods to solve the problem of video segmentation based on 2𝔻 image analysis using pre-existing tools and developing new tools in the domains of mathematical morphology and digital topology. In Fig. 1.2 we illustrate the general framework of the approach used in this

work. Firstly, the video is transformed into $2\mathbb{D}$ images. Operators of image processing are applied to these images to identify some specific events, like cuts, fades, dissolves and flashes.

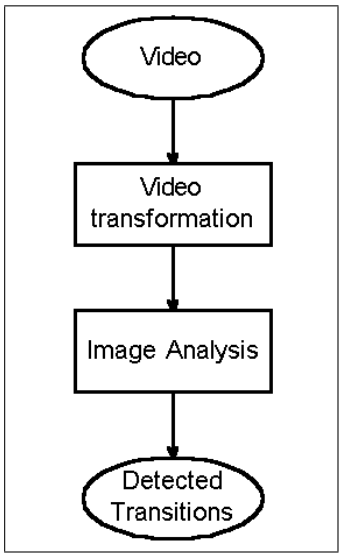


Figure 1.2: General framework for our approach to the video segmentation problem.

In the following, we enumerate the major contributions of our work:

- development of video transition identification methods that present low rates of false detection and high rates of correct detection [36].
- definition and exploitation of a new transformation of the video content, named visual rhythm by histogram. After application of this transformation, it is possible to identify cuts, fades and flashes [31, 36, 33].
- a new approach to identify gradual video transitions. This approach is based on the application of a sharpening method [35] .
- a specific method to detect fade [33].
- two methods to detect flashes [36].

1.2 Organization of the text

We use some basic concepts of Mathematical Morphology and Digital Topology (for detailed explanations see [68, 63, 5, 45]). The text is organized as follows.

Chapter 2 - Video model. We discuss the video model taking into account different types of transitions and camera operations.

Chapter 3 - Video analysis. We present related works, considering two different approaches to tackle the video segmentation problem.

Chapter 4 - Video transformation from $2\mathbb{D} + t$ to $1\mathbb{D} + t$. We define a video transformation in which video events are transformed into image patterns.

Chapter 5 - Introduction for video transition identification. We introduce the video transition identification problem. We summarize some patterns in which we are interested, and also the correspondence between the patterns and the transitions.

Chapter 6 - Image analysis operators. We present and define some basic image operators that will be used in our work.

Chapter 7 - Cut detection. A method for cut detection is presented. It takes advantage of the fact that each video event is transformed into a specific pattern on the transformed image.

Chapter 8 - Flash detection. Two methods to identify flashes are presented. One method is a variant of the cut detection method (Chapter 7) and the other considers a new approach based on the filtering of a data structure created from statistical values of the transformed image.

Chapter 9 - Gradual transition detection. Two methods are used to identify gradual transitions. One method considers a natural approach in which a multi-scale analysis is used. The other method tries to eliminate the gradual transitions by applying a sharpening operator followed by an analysis of the sharp transitions.

Chapter 10 - Specific fade detection. We propose a specific fade detection method. Here, we use a new video transformation that considers the histogram information.

Chapter 11 - Conclusions and future works. Finally, we conclude by presenting some future works and discussing the main contributions of our work.

Part I

Theoretical background

Chapter 2

Video model

Video is a medium for storing and communicating information. The two most important aspects of video are its contents and its production style [39]. The video content is the information that is being transmitted, and the video production style is associated with the category of a video, such as commercial, drama, science fiction, etc. In this chapter, we will define some of the concepts used in literature (like shot, scene and keyframe). Also, we will present the most popular types of transitions (like cut, dissolve, fade and wipe) and types of camera works (like zoom and pan).

2.1 Basic definitions

Let $\mathbb{A} \subset \mathbb{N}^2$, $\mathbb{A} = \{0, ..., \mathrm{H}-1\} \times \{0, ..., \mathrm{W}-1\}$, where H and W are the height and the width in pixels of each frame, respectively.

Definition 2.1 (Frame) *A frame f_t is a function from $\mathbb{A}$ to $\mathbb{N}$ where for each spatial position (x, y) in $\mathbb{A}$, $f_t(x, y)$ represents the grayscale value of the pixel (x, y) at time t.*

Definition 2.2 (Video) *A video* V*, in domain $2\mathbb{D} + t$, is a sequence of frames f_t and can be described by*

$$\mathrm{V} = (f_t)_{t \in [0, duration-1]} \tag{2.1}$$

where *duration* is the number of frames contained in the video. The number of frames is directly associated with the frequency and the duration of visualization. Generally, the

huge amount of information represents a problem for video information storage. Thus, to facilitate the task, file compression techniques, such MPEG 1, MPEG 2, MPEG 7 and M-JPEG are normally used. In this work, all videos are compressed in MPEG 1 format. To obtain information for each frame, we decompress only the specific frame without decompressing the whole video file. Also, each frame is represented in the RGB (*Red*, *Green*, *Blue*) color space. To facilitate our analysis, we consider only grayscale images, in which grayscale values are obtained by a simple operation of averaging of the RGB values.

A video is typically composed by a large quantity of information that is hard to be physically or semantically related. This is why there is the need to segment a video.

The video fundamental unit is a shot (defined below) that captures a continuous action considering the recording by a single camera, where the camera motion (e.g., zoom and pan) and the object motion are permitted. According to [4], a scene is composed by a small number of shots. While a shot represents a physical video unit, a scene represents a semantic video unit. To summarize, a scene is a shot grouping that is composed by a frame sequence, as illustrated in Fig. 2.1.

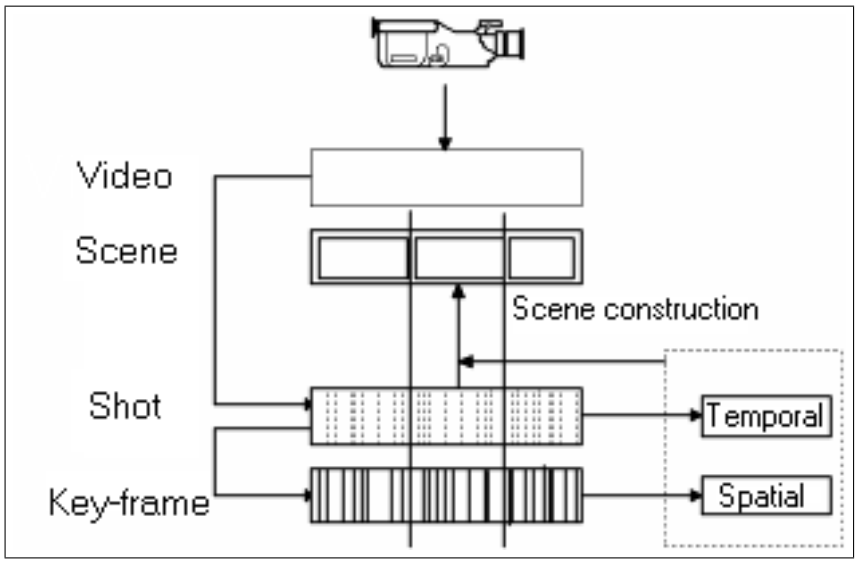


Figure 2.1: Video hierarchical representation [61].

Definition 2.3 (Shot [4]) *Shot is the fundamental unit of a video, because it captures a continuous action from a single camera. A shot represents a spatio-temporally coherent*

frame sequence.

Definition 2.4 (Scene [4]) *A scene is usually composed by a small number of inter-related shots that are unified by similar features and by temporal proximity.*

Definition 2.5 (Key-frame) *A key-frame is a frame that represents the content of a logical unit, as a shot or scene. This content must be the most representative as possible.*

A shot consists of a sequence of frames with no significant change in the frame content. So, by definition, two consecutive frames belong to the same shot if their content is very similar (or exactly equal). This is true if $f_t(x, y) \cong f_{t+1}(x, y)$ for all $t \in [1, duration]$, where *duration* means shot size. To study the behavior of consecutive frames, we will define a $1\mathbb{D}$ image, g, that consists of pixel values $f_t(x, y)$ in a spatial position (x, y) at each time t.

The shots can be grouped into many ways according to the desired effect. For example, when we search for a video that transmits the idea of motion, the shots must change rapidly. After that, we describe the transition types without considering the subjective and intentional aspects.

2.2 Types of transitions

Usually, the process of video production involves two different steps [39], shooting and editing, the former is for production of shots and the later is for compilation of the different shots into a structured visual presentation. This compilation is associated with the type of transition between consecutive shots.

Definition 2.6 (Transition) *All edition effect that permits the passing from one shot to another is called transition. Usually, a transition consists of the insertion of a series of artificial frames produced by an editing tool.*

To facilitate the understanding of transition types, we will introduce some notations. The shot before a transition is denoted by P and the shot after a transition by Q. $P(t)$ will represent the frame at time t in shot P. The transition between two shots P and Q, denoted by T, begins at time t_0 and ends at time t_1. A frame $T(t)$ represents a frame at time t in the transition T, where $t \in]t_0, t_1[$.

The choice of transitions in an editing process is associated with the features and style of the video. The simplest transitions are cut, dissolve, fade and wipe. These transitions are usually subdivided into sharp transitions (cuts) and gradual transitions (dissolves, fades and wipes).

2.2.1 Cut

The simplest transition is the cut, which is characterized by the absence of new frames between consecutive shots. Fig. 2.2 illustrates an example of the cut.

Definition 2.7 (Cut (sharp transition)) *A cut is a type of transition where two consecutive shots are concatenated, i.e., no artificial frame is created and the transition is abrupt.*

Figure 2.2: Cut example.

2.2.2 Fade

According to [7], the fade process is characterized by a progressive darkening of a shot until the last frame becomes completely black, or inversely. A more general definition is given by [50], where the black frame is replaced by a monochromatic one. Fades can be subdivided into two types: fade-in and fade-out.

Definition 2.8 (Fade-out) *A fade-out is characterized by a progressive disappearing of the visual content of the shot P. The last frame of a fade-out is a monochromatic frame G (e.g., white or black). The frames of a fade-out can be obtained by*

$$T_{f_o}(t) = \alpha(t) \times G + (1 - \alpha(t)) \times P(t) \tag{2.2}$$

where $\alpha(t)$ is a monotonic transformation function that is usually linear and $t \in \ ]t_0, t_0 + duration[$ where *duration* represents the duration of the fade.

Definition 2.9 (Fade-in) *A fade-in is characterized by a progressive appearing of the visual content of the shot Q. The first frame of the fade-in is a monochrome frame G (ex. white or black). The frames of a fade-in can be obtained by*

$$T_{f_i}(t) = (1 - \alpha(t)) \times G + \alpha(t) \times P(t) \tag{2.3}$$

Fig. 2.3 illustrates an example of the fade-out.

Figure 2.3: Example of fade-out.

2.2.3 Dissolve

Inversely to the cut, the dissolve is characterized by a progressive change of a shot P into a shot Q.

Definition 2.10 (Dissolve) *A dissolve is a progressive transition of a shot P to a consecutive shot Q with non-null duration. Each transition frame can be defined by*

$$T_d(t) = (1 - \alpha(t)) \times P(t) + \alpha(t) \times Q(t) \tag{2.4}$$

Fig. 2.4 illustrates an example of a dissolve.

Figure 2.4: Example of dissolve.

2.2.4 Wipe

The wipe is characterized by a sudden change of pixel values according to a systematic way of chosing the pixels to be changed. Next, two different kinds of wipe are defined.

Definition 2.11 (Horizontal wipe) *In a horizontal wipe between two shots P and Q, a "vertical line" is horizontally shifted left or right subdividing a frame in two parts, where in one side of this line we have contents of P, and in the other side the new shot Q.*

Definition 2.12 (Vertical wipe) *In a vertical wipe between two shots P and Q, a "horizontal line" is horizontally shifted up or down subdividing a frame in two parts, where in one side of this line we have the content of P, and in another side the new shot Q.*

Fig. 2.5 and Fig. 2.6 illustrate the examples of horizontal and vertical wipes, respectively.

Figure 2.5: Example of a horizontal wipe (right to left).

Figure 2.6: Example of a vertical wipe (down to up).

2.2.5 Transition classification

In [39, 22], one can find a classification of different types of transitions considering editing aspects. The classes of transitions are defined as follows:

TYPE 1: IDENTITY CLASS - the editing process does not modify either shot, it does not

have a geometric model and the duration of this transition is null. The cut is the only transition of this class.

Type 2: Spatial class - only the spatial aspect between the two shots is manipulated, so there is a geometric model and the duration of this transition is not null, but the duration of a pixel transformation is null. The wipe is an example of this class.

Type 3: Chromatic class - only the color space is manipulated without a geometric model because the whole image is modified at the same time, and the duration of this transformation is not null. The fade-in, fade-out and dissolve are examples of this class.

Type 4: Spatio-Chromatic class - both color space and spatial aspect are manipulated. The morphing is a kind of this class.

2.3 Camera work

The basic camera operations [2], as illustrated in Fig. 2.7, include: panning - camera horizontal rotation; zooming - varying of focusing distance; tracking - horizontal traverse movement; booming - vertical traverse movement; dollying - horizontal lateral movement; tilting - vertical camera rotation.

Generally, panning and tilting are used to follow moving objects without changing the location of the camera. Zooming indicates a change in "concentration" about something. Tracking and dollying often represent a change in viewing or following moving objects with a camera motion. Fig. 2.8 illustrates an example of a zoom-in.

2.4 Conclusions

In this chapter, we have shown some transition types used in video editing, like cut, dissolve and fade. We also have shown the most common camera operations, like zoom and pan. The choice of the events in the video process creation reflects the target of the video, and also the video style. The event identification (camera operation detection and

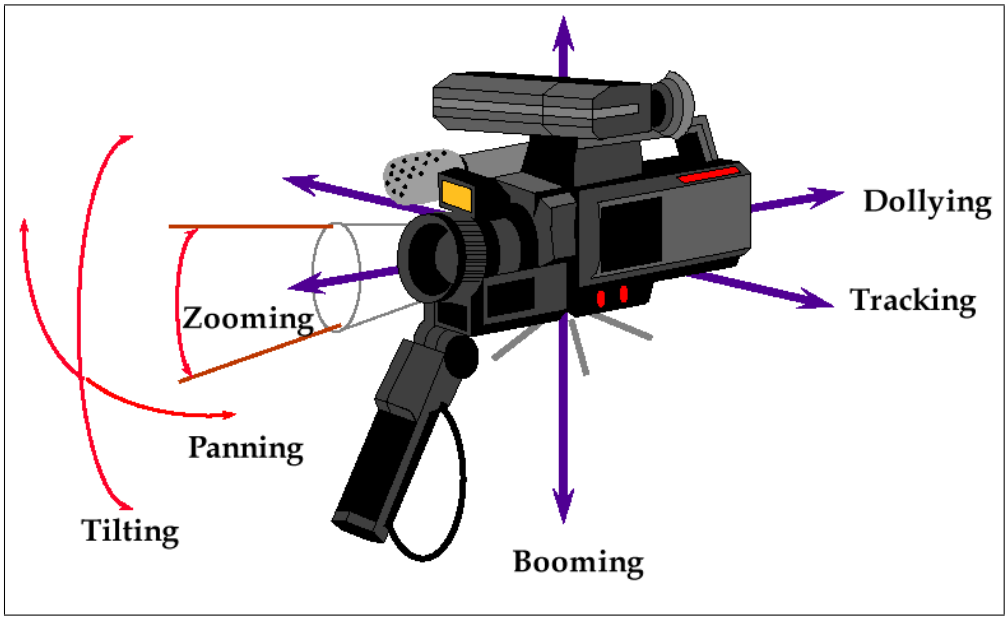


Figure 2.7: Camera basic operations.

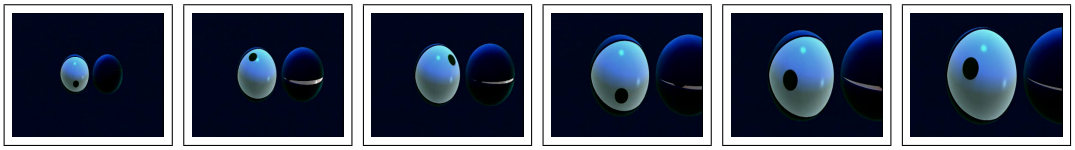


Figure 2.8: Example of zoom-in.

transition detection) is an enormous and promising domain of study. In this work, we consider only the transition detection problem.

Chapter 3

Video analysis

Excellent works can be found in literature which summarize and review the main video problems and the way to cope with them [53, 12, 72, 7]. In general, the first step in video analysis is video parsing, which can be divided in:

- video segmentation, which consists in dividing the video in fundamental units according to the semantic level considered, like shot, scene, etc;
- camera work analysis, which consists in identifying camera operations, like zoom and pan;
- key-frame extraction, which consists in identifying the representative frames of each fundamental unit.

Following the parsing, other tasks can be considered, like classification, summarization and indexing. Classification is concerned with separating scenes or shots in different classes. According to [72], summarization is essential in building a video retrieval system where we need to have compact video representations to facilitate browsing. And finally, video indexing is fundamental for querying large video databases, where the quality and speed of responses depend on the video descriptor used in the indexing phase.

Usually, video problems are basically related to the analysis of visual features, such as, color, texture, shape and motion. In this work, we will review some of the most popular methods to cope with video segmentation problem and we will describe some quality measures that can be used to compare different methods related to video parsing.

3.1 Video segmentation

The video segmentation problem can be considered as a problem of dissimilarity between images (or frames). Usually, the common approach to deal with this problem is based on the use of a dissimilarity measure which allows the identification of the boundary between consecutive shots. The simplest transitions between two consecutive shots are sharp and gradual transitions [39]. A sharp transition (cut) is simply a concatenation of two consecutive shots. When there is a gradual transition between two shots, new frames are created from these shots [39].

In the literature, we can find different types of dissimilarity measures used for video segmentation, based on different visual features, such as pixel-wise comparison, histogram-wise comparison, edge analysis, etc. If two frames belong to the same shot, then their dissimilarity measure should be small, and if two frames belong to different shots, this measure should be large, but in the presence of effects like zoom, pan, tilt and flash, this measure can be affected. Hence, the choice of a meaningful measure is essential for the quality of the segmentation results. Another approach to the video segmentation problem is to transform the video into a $2\mathbb{D}$ image, and to apply image processing methods to extract specific patterns related to each transition. In Fig. 3.1, we illustrate a diagram that represents the main approaches for video segmentation which are subdivided into dissimilarity measures and $2\mathbb{D}$ image analysis. In the next sections we will discuss these approaches and, also, their features and limitations for video segmentation.

3.1.1 Approaches based on dissimilarity measures

The use of dissimilarity measures is based on the following hypothesis: "Two or more frames belong to the same shot if their dissimilarity measure is *small*". The definition of dissimilarity measures is a bit fuzzy, and it usually depends on the type of visual feature used to compare two or more frames. Visual features can be broadly classified into four categories: color, texture, shape and motion.

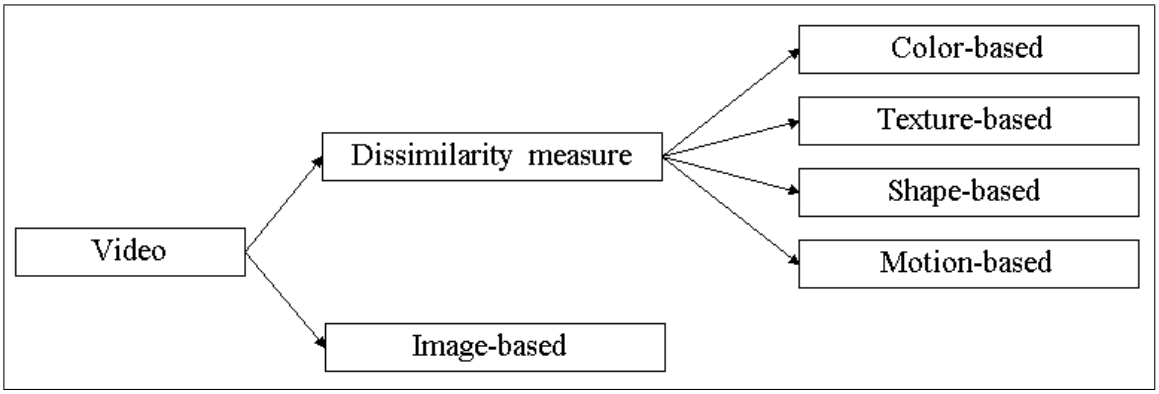


Figure 3.1: Two different approaches for video segmentation.

3.1.1.1 Visual features

In what follows, we describe the visual features and make some important remarks about the most popular methods for video segmentation.

Color-based. The color information is an important image content component, because it is robust with respect to object distortion, object rotation and object translation [66]. The simplest approach used for video segmentation is based on pixel-wise comparisons between corresponding pixels in consecutive frames. In [79], a simple technique is defined to detect a qualitative change in the content between two consecutive frames. The sensibility to camera motion represents a problem of this technique, but this influence can be minimized by the use of a smoothing filter. Another pixel-wise dissimilarity measure is associated with the average of the pixel difference between two consecutive frames.

Typically, the image color information is represented by an image histogram [6]. The image histogram (H_f) with color in the range $[0, L-1]$ is a discrete function $p(i) = n_i/n$, where i is a color of a pixel, n_i is the number of pixel in the image f with color i, n is the total number of pixels in the image and $i = 0, 1, 2, ..., L-1$. Generally, $p(i) = n_i/n$ represents an estimation of the probability of color occurrence i. Taking advantage of the global information and the invariance to rotation and translation, some dissimilarity measures based on histogram can be found in [79, 7, 69, 46], among them, are the histogram intersection and the histogram difference. Another possibility to use color information is based on statistical measures. [65] used some measures for cut detection, like the χ^2 test. Also, with the purpose to reproduce the creation process of the gradual

transition from mathematical equations, [26] studied the behavior of gradual transitions, as fades and dissolves, but his method fails in small transitions (transitions with duration smaller than 15 frames).

Texture-based. Texture is an important element for the human visual system that helps to provide in a scene the idea of depth and orientation of surfaces. The texture feature represents a very important descriptor of natural images, and is also related to a visual pattern with some homogeneity property [19, 24, 40, 18, 21, 51].

Other representations for texture information include Markov random fields, Gabor transforms and wavelet transforms [49]. For video segmentation based on texture, [73] used wavelet transform for cut detection.

Shape-based. Shape is an important criterium to characterize objects and it is related to their profile and their physical structures [9, 55, 67, 27, 59]. Its use is more frequent where the image objects are similar to color and texture, as is the case of medical images. In image retrieval applications, the shape feature can be considered as global or local. Some global features are symmetry, circularity and central moments. Local features are associated with size and segment orientation, curvature points, curve angles, etc.

As for image retrieval, we can use the local features as dissimilarity measures for video segmentation. Some works consider edge analysis for event detection [75, 76, 77], this technique is associated with the number of edges that appear and disappear in consecutive frames, but this method is very costly with respect to computational time due to the computation of edges for each frame of the sequence. From this method, it is possible to detect fades, dissolves and wipes.

Motion-based. Motion can be considered the most important attribute of a video. Two approaches can be considered for motion-based video segmentation: blocking matching and optical flow [41]. The motion-based video segmentation is related to the change of the motion model in consecutive shots. From this approach, high-level features that reflect camera motions such as panning, tilting and zooming can be extracted (as we will see in Sec. 3.2).

3.1.1.2 General frameworks for video segmentation

With the aim of increasing the robustness of the results for video segmentation, some techniques can be applied to different approaches, i.e., we can apply the same process to different dissimilarity measures. In the original papers, the presented techniques were applied only to a specific dissimilarity measure, but we can easily extend to other measures. So, we will present these techniques without considering specific measures.

Block-comparison [56, 79]. Instead of computing dissimilarity measures directly from all pixels of the image, the images are divided in blocks and these measures are computed for each pair of blocks in consecutive frames. Afterwards, we perform an analysis of the number of blocks that present a dissimilarity measure greater than a threshold. This technique is more robust to local motions and to noise than pixel-wise comparison.

Post-filtering [22, 31]. This technique is based on the filtering of the dissimilarity measures computed on the video. Firstly, the dissimilarity measure is computed. Secondly, a filter is applied to the $1\mathbb{D}$ image (resulting of the computation of the dissimilarity measure). And finally, a thresholding is applied. This technique can be used to decrease the number of false detections and to detect some patterns in the $1\mathbb{D}$ image.

Twin-comparison [79]. Considering that in gradual transitions the dissimilarity measures between two transition frames are greater than the dissimilarity measures of frames in a same shot, two different thresholds can be considered T_b and T_s. $T_s > T_b$ are set for cut and gradual transitions, respectively. If a dissimilarity measure $d(i, i+1)$ between two consecutive frames satisfies $T_b < d(i, i+1) < T_s$, then potential start frames for gradual transitions are detected. For every potential frame detected, an accumulated comparison $A(i) = \sum d(i, i+1)$ is computed if $A(i) > T_b$ and $d(i, i+1) < T_s$. The end frame of the gradual transition is declared when the condition $A(i) > T_s$ is satisfied.

Plateau-method [74]. With the objective of detecting gradual transitions, instead of calculating dissimilarity measures between two consecutive frames, these are calculated between two frames at time t and $t+k$, where k depends on the duration of the transition to detect.

The choice of a framework and dissimilarity measures, depends on the time constraints and the expected quality of the results. Usually, for cut detection, the methods that use pixel values and statistical measures are the fastest, but the best results are obtained by histogram analysis, where both can consider the block comparison and/or post-filtering. For gradual transitions, the histogram twin-comparison is the most indicated, because the plateau-method is highly dependent on the maximum duration permitted for a transition.

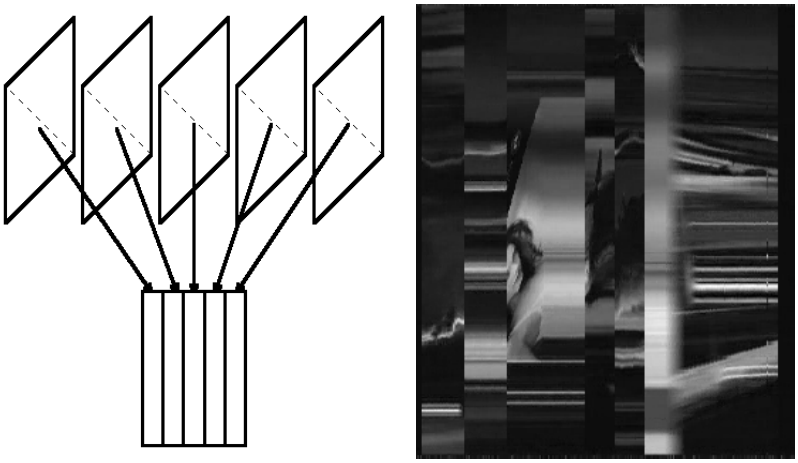
3.1.2 Image-based

When we work directly on the video, we have to cope with two main problems: the high processing time and the choice of a dissimilarity measure. Looking for reducing the processing time and using tools for $2\mathbb{D}$ image segmentation instead of a dissimilarity measure, we can transform the video into a $2\mathbb{D}$ image (as we will see in Sec. 4).

The image-based approach for video segmentation is based on transforming the video V into a $2\mathbb{D}$ image, called VR, and applying image processing methods on VR to extract the specific patterns related to each transition. Informally, each frame is transformed into a vertical line of VR, as illustrated in Fig. 3.2(a). This approach can be found in [71, 17, 57]. [71] defined the X-ray and Y-ray as the result of a video transformation obtained by a linear image transformation in each axis, and an edge analysis was performed to detect cuts. They also cited another video transformation based on the intensity histogram, but it was not well defined and not exploited. [17] defined the visual rhythm and [57] defined the spatio-temporal slice, both related to the same video transformation and a sub-sampling of each frame, like the principal diagonal sub-sampling (illustrated in Fig. 3.2(b)). [17] used statistical measures to detect some patterns, but the number of false detections was very high. [57] used Markov models for shot transition detection, but it fails when the contrast is low between textures of consecutive shots. Statistical measures are used in [17, 23] for cut, wipe, fade and dissolve detection.

3.2 Camera work analysis

Camera operations represent the effects controlled by the camera man with the intention of following objects and/or changing the "concentration" on an object. In this work, we



(a) (b)

Figure 3.2: Video transformation: (a) simplification of the video content by transformation of each frame into a column on VR; (b) a real example of the principal diagonal sub-sampling.

consider only camera operation presence, such as zooming and panning.

Nowadays, the main direction of research on estimating camera operations is the optical-flow-based approach which consists of two main steps [52]: computing the optical flow in a frame and analyzing the optical flow to extract camera operations. According to [52], this approach has two major faults in practice: it is generally time-consuming and it may be generally affected by noise, such as unintentional camera vibration and flashlight presence. With the purpose of eliminating these faults, [52] presented an approach to extract camera operations based on X-ray and Y-Ray images, and texture analysis, where the movement can be estimated by investigating the directivity of the textures by $2\mathbb{D}$ Discrete Fourier Transform. [46] considered the result of the $2\mathbb{D}$ fast Fourier Transform as an input for a multilayer perceptron neural network, which will detect the camera operations.

Another method to extract camera operations is presented in [1]. In this method, called Video Tomography Method (VTM), tomographic techniques are applied to video motion estimation, allowing the visualization of the structure of motion. The VTM is based on spatio-temporal processing of video. Informally, two spatio-temporal images,

the X-ray and Y-ray are computed by horizontal and vertical sub-samplings. Afterwards, an edge filter is applied with the objective of enhancing of the edges, in particular, a first order derivative is used. Finally, the Hough Transform is applied to extract zoom and pan. The purpose of this step is to combine the two main stages of the classical camera operation detection problem, the matching procedure and the estimation process. [42] used also Hough Transform, but differently from [1], the edge detection step was eliminated. According to [42], its results for motion detection are better than those presented in [1].

The utilization of DFT and Hough Transform to detect specific patterns is interesting, but the parameter analysis is difficult and the computational cost is relatively high.

3.3 Quality measures for video analysis

The increase in the number of methods for video parsing makes the choice between them very difficult. In order to find the best method, it is necessary to use the same evaluation criteria for different works. What follows, we describe some of the metrics found in literature that are used to characterize the video segmentation quality.

Generally, the goal of quality measures is to verify the performance of methods in detecting different events in the video. An event can involve transitions like cut, dissolve, wipe and camera operations like zoom, pan and tilt.

In this work, we define two new measures: robustness, which allows one to analyse the sensitivity of a transition detection method relative to a threshold; and gamma measure, which sets a compromise between the number of correct and false detections. We also present a quantitative analysis of the results.

3.3.1 Quantitative analysis

We denote by $\#Events$ the number of events (cuts, fades, flashes, etc.), by $\#Corrects$ the number of events correctly detected, by $\#Falses$ the number of detected frames that do not represent an event and by $\#Misses$ the number of the events that were not detected, defined by $\#Misses = \#Events - \#Corrects$. From these numbers we can define two basic quality measures.

Definition 3.1 (Recall, error and precision rates) *The ratio of number of events correctly detected to the number of events is called the recall rate. The ratio of number of events falsely detected to the number of events is called the error rate. The ratio of number of events correctly detected to the sum of correctly and falsely detected is called the precision rate. These rates are given by*

$$\alpha = \frac{\#Corrects}{\#Events} \qquad (recall) \tag{3.1}$$

$$\beta = \frac{\#Falses}{\#Events} \qquad (error) \tag{3.2}$$

$$\delta = \frac{\#Corrects}{\#Corrects+\#Falses} \qquad (precision) \tag{3.3}$$

3.3.2 Threshold sensitivity

Let τ be the threshold used for transition detection within the range $[0, 1]$. Let $\alpha(\tau)$ and $\beta(\tau)$ be the recall and error, respectively, with respect to a given threshold τ. If we consider that for each threshold τ we obtain different values for α and β, we can represent these relations as functions $\alpha(\tau)$ and $\beta(\tau)$, respectively. A new measure can be created to relate ranges in which α and β are adequate, according to the allowed percentages of misses and false detections.

Definition 3.2 (Robustness) *Let $\alpha(\tau)$ and $\beta(\tau)$ be the functions that relate the threshold to recall and error rates, respectively. Let P_m and P_f be the percentage of miss and false detection that are allowed. The robustness μ is a measure related to the interval where the recall and error rates have values smaller than $(1 - P_m)$ and P_f, respectively. This measure is within the range $[0, 1]$ and is given by*

$$\mu(P_m, P_f) = \begin{cases} \alpha^{-1}(1 - P_m) - \beta^{-1}(P_f) & , \quad if \;\; \alpha^{-1}(1 - P_m) - \beta^{-1}(P_f) > 0 \\ 0 & , \quad otherwise \end{cases} \tag{3.4}$$

where α^{-1} and β^{-1} are the inverses of the functions $\alpha(\tau)$ and $\beta(\tau)$, respectively. Here, the result of the inverse functions is the rate of recall and error that are related to threshold, respectively. In Fig. 3.3, we illustrate the robustness measure obtained from functions $\alpha(\tau)$ and $\beta(\tau)$. Usually, it is expected that the results of the methods present the characteristics

of the graphic illustrated in Fig. 3.3, because it is desired to correctly detect events with a "small" number of false detections.

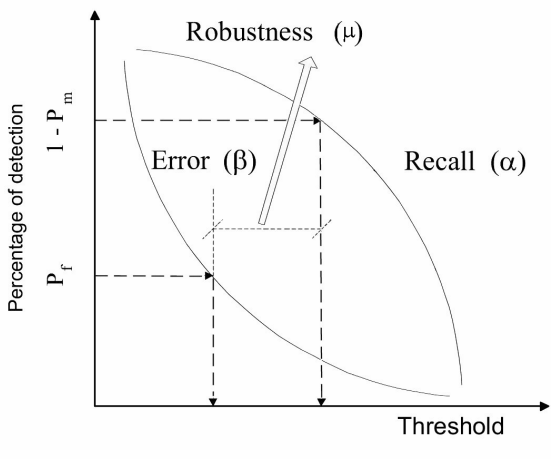


Figure 3.3: Robustness (μ) measure.

Next, we define two other measures, E_m and R_f, that are associated with the absence of miss and false detection, respectively.

Definition 3.3 ("Missless" error) *The missless error E_m is associated with the percentage of false detection when we have results without miss (a "small ratio" of miss P_m can be permitted, like 3%). The missless error is given by*

$$E_m(P_m) = \beta(\alpha^{-1}(1 - P_m)) \tag{3.5}$$

Definition 3.4 ("Falseless" recall) *The falseless recall R_f is associated with the percentage of correct detection when we have results without false detection (a "small number" of false detection P_f can be permitted, like 1%). The falseless recall is given by*

$$R_f(P_f) = \alpha(\beta^{-1}(P_f)) \tag{3.6}$$

When we use methods for video parsing, we expect the recall to be the highest when error rate is the smallest. To find the best compromise between these two requirements, we must define a "reward function" by combining $\alpha(\tau)$ and $\beta(\tau)$. Since high values of α and low values of β should be rewarded, the function $\alpha(\tau) \times (1 - \beta(\tau))$ is a natural choice.

Definition 3.5 (Gamma measure) *The gamma measure γ represents the maximal value of the reward function defined above for all possible values of τ:*

$$\gamma = max\{\alpha(\tau) \times (1 - \beta(\tau))|\tau \in [0, 1]\} \tag{3.7}$$

The gamma measure could also be defined as the area between the two curves, α and β, but this measure is not exploited in this work. The quality of the results is associated with the values of the measures above defined.

To declare that a method A is better than a method B, it is necessary to evaluate the quality measures defined previously, for example, A is better than B if the values of robustness, falseless recall and gamma measure of A are higher than the values of B, and if the values of missless error of A is smaller than B. If these considerations do not occur for all measures, then we need to consider the features more important for the application. To generalize, the highest values of robustness, falseless recall and gamma measure represent the best results of a method. The lowest values of missless error represent the best results of a method.

3.4 Conclusions

In this chapter, we enumerated some video problems and we described how we can cope with these problems. Also, we described some quality measures used in video analysis to compare different methods.

For the video segmentation, we presented two approaches, dissimilarity-measure-based and image-based. Due to the problem of choice a dissimilarity measure, and the possibility of using $2\mathbb{D}$ image segmentation, we will transform the video segmentation problem into a problem of $2\mathbb{D}$ image segmentation without consider dissimilarity measures (as we will see in Chapter 4).

Chapter 4

Video transformation from $2\mathbb{D}+t$ to $1\mathbb{D}+t$

Usually, the shot transition detection is the first step in the process of automatic video segmentation and it is associated with the detection of a cut and gradual transitions between two consecutive shots [39]. In this chapter, we transform the video segmentation problem into a $2\mathbb{D}$ image segmentation problem, taking advantage of the observation that each video event is represented by a specific pattern in this image. We present the main patterns found in this image making a correspondence with the video events. We also discuss about the proposed approaches to solve the problem of image segmentation using morphological and topological tools, without the need of defining a dissimilarity measure between frames. In this way, we can use a simplification of the video content, called visual rhythm, where the video segmentation problem, in the $2\mathbb{D}+t$ domain, is transformed into a problem of pattern detection, in domain $1\mathbb{D}+t$. So, we can apply methods of $2\mathbb{D}$ image processing to identify different patterns on the visual rhythm because each video effect corresponds to a pattern in this image, for example, each cut is shows up as a "vertical line" on the visual rhythm.

4.1 Visual rhythm by sub-sampling

The visual rhythm obtained by sub-sampling (or simply visual rhythm) is a simplification of the video content represented by a $2\mathbb{D}$ image. This simplification can be obtained by a

systematic sampling of points of the video, such as, extraction of points in the diagonal of each frame.

Definition 4.1 (Visual rhythm [17] or spatio-temporal slice [57]) *Let* $\mathrm{V} = (f_t)_{t\in[0,duration-1]}$ *be an arbitrary video sequence, in domain* $2\mathbb{D}+t$*. The visual rhythm, in domain* $1\mathbb{D}+t$*, is a simplification of the video in which each frame* f_t *is transformed into a vertical line of the visual rhythm image VR, defined by*

$$VR(t,z) = f_t(r_x \times z + a, r_y \times z + b)$$

where $z \in \{0,...,\mathrm{H}_{VR}-1\}$ and $t \in \{0,...,duration-1\}$, H_{VR} and *duration* are, respectively, the height and the width of the visual rhythm, r_x and r_y are pixel sampling ratios, a and b are spatial offsets on each frame. Thus, according to these parameters, different pixel samplings could be considered, for example, if $r_x = r_y = 1$ and $a = b = 0$ and $\mathrm{H} = \mathrm{W}$ then we obtain all pixels of the principal diagonal. If $r_x = 1$ and $r_y = 0$ and $a = 0$ and $b = \mathrm{W}/2$ then we obtain all pixels of the central horizontal line. If $r_x = 0$ and $r_y = 1$ and $a = \mathrm{H}/2$ and $b = 0$ then we obtain all pixels of the central vertical line, etc. In Fig. 4.1, we show 4 different examples of the pixel samplings.

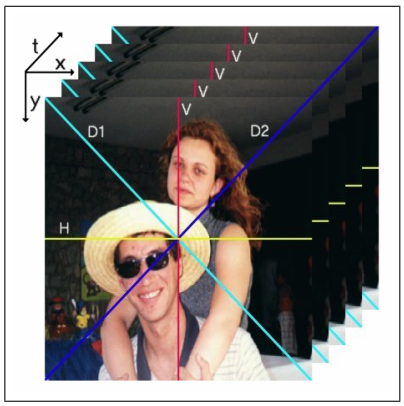
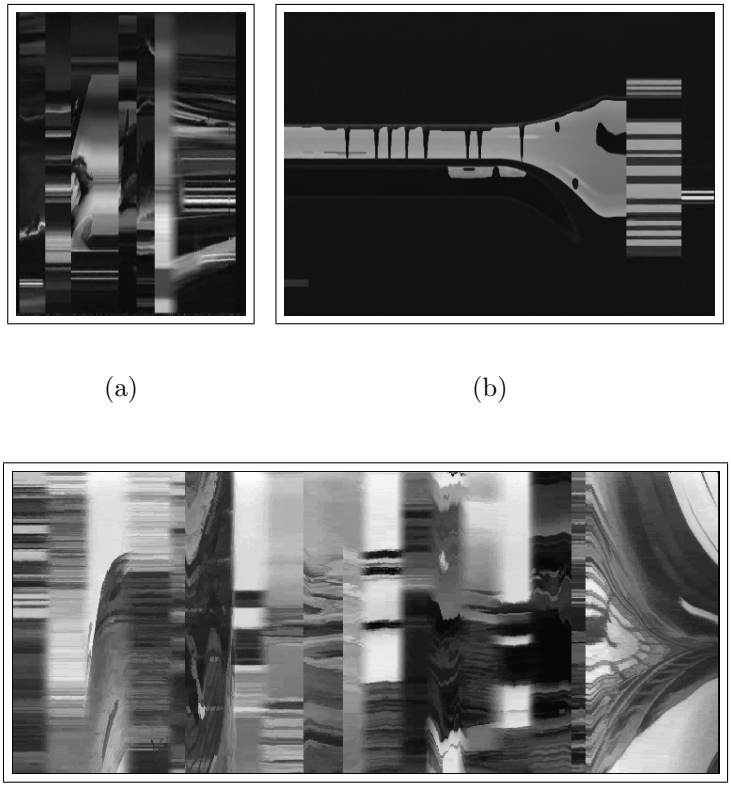


Figure 4.1: Example of pixel samplings: D1 is the principal diagonal, D2 is the secondary diagonal, V is the central vertical line and H is the central horizontal line.

4.1.1 Pattern analysis

The choice of a pixel sampling constitutes an important problem since different samplings produce different visual rhythms in which video events (cuts, fades, flashes, etc) will

appear as different patterns. Unfortunately, depending on the type of visual rhythm, this correspondence is not a one-to-one relation, i.e., a cut transition corresponds to a vertical line, but a vertical line is not necessarily a cut. This problem can be solved by considering visual rhythms obtained at different pixel sampling rates. Afterwards, a simple intersection operation between these results may be used to correctly identify sharp video transitions.



(a) (b)

(c)

Figure 4.2: Examples of visual rhythm by principal diagonal sub-sampling: (a) video "0132.mpg"; (b) video "0599.mpg" and (c) video "0117.mpg".

Fortunately, in general, we need to use only the visual rhythm obtained by principal diagonal sampling because the correspondence problem rarely occurs in practice. Furthermore, the visual rhythm represents the best simplification of the video content, according to [17]. [17] presented different pixel sampling possibilities with their corresponding visual rhythms. The authors mention that the best results are obtained when sampling

is based on a diagonal because it usually contains both horizontal and vertical features. In Fig. 4.2, we show examples of visual rhythms obtained by the principal diagonal sub-sampling. Another example of visual rhythm is shown in Fig. 4.3, where we illustrate two visual rhythms obtained from the same video but with different pixel sampling rates. In these cases we use the principal diagonal (Fig. 4.3-top) and the central vertical axis (Fig. 4.3-bottom) samplings. We can observe that there are "vertical lines" in Fig. 4.3-bottom that do not correspond to a sharp video transition but all sharp video transitions correspond to "vertical lines" on the visual rhythm.

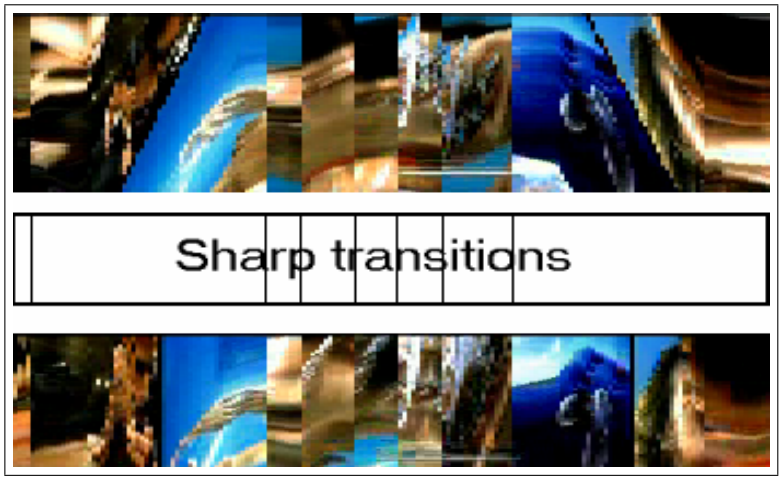


Figure 4.3: Visual rhythm obtained by a real video using different pixel sub-samplings: principal diagonal (top) and central vertical line (bottom). The temporal positions of the cuts are indicated in the middle image.

In Fig. 4.4, we illustrate the correspondence between visual patterns and video events. Next, we analyse the different kinds of patterns on the visual rhythm by sub-sampling.

Sharp image transitions. The notion of sharp transitions on the visual rhythm depends on the human visual perception. For example, in Fig. 4.5, we illustrate vertical sharp transitions and an inclined sharp transition. According to the type of visual rhythm, this pattern can be generally associated with cuts and wipes. On the visual rhythm by principal diagonal sub-sampling, for example, all cuts are represented by vertical transitions, but some vertical transitions can be related to diagonal wipes. In practice the correspondence problem is irrelevant because the number of wipes is typically considerably smaller than the number of cuts. So, the cut detection can be related to detection

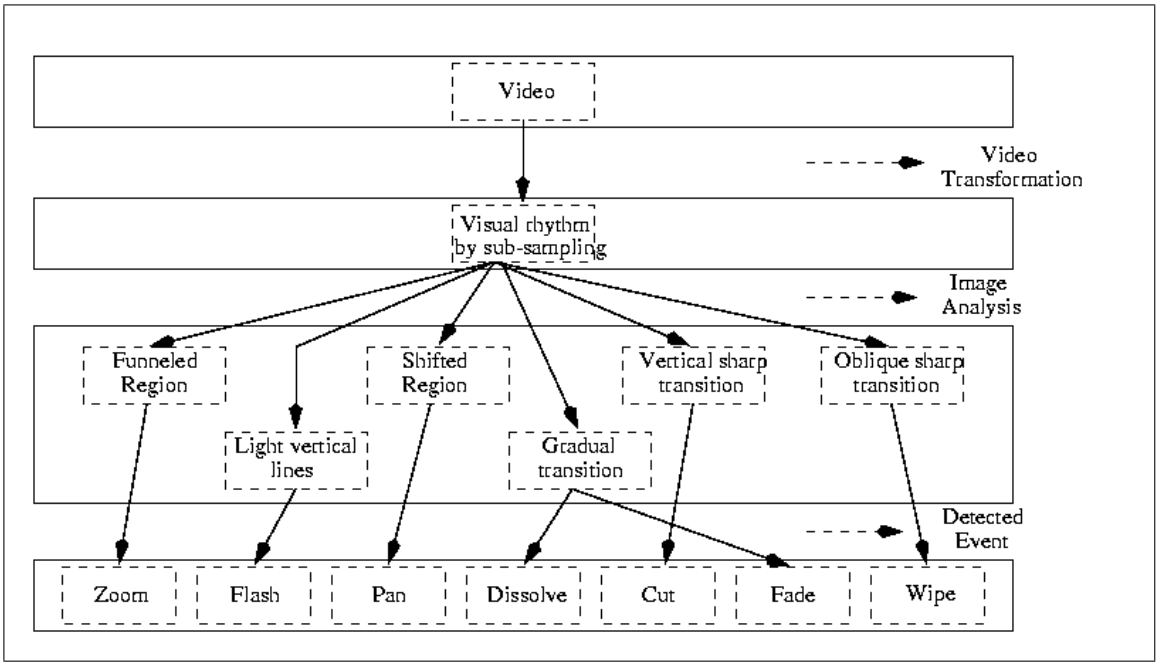
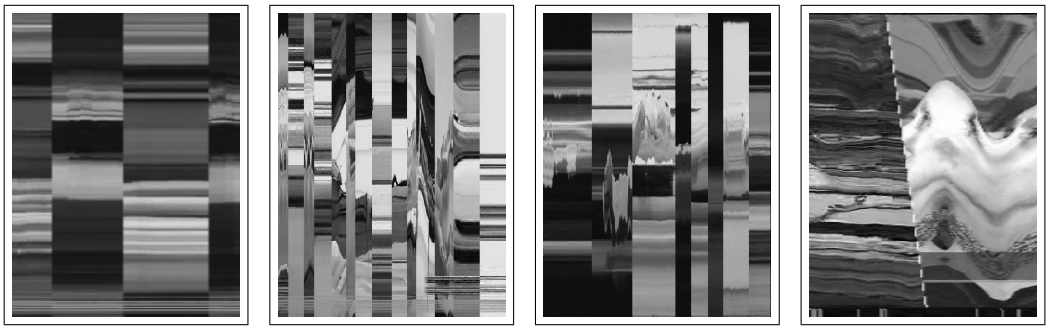


Figure 4.4: Block diagram for video segmentation using visual rhythm by sub-sampling.

of vertical transition lines, and the wipe detection can be associated with the detection of inclined sharp transitions, both considering the visual rhythm by principal diagonal sub-sampling.

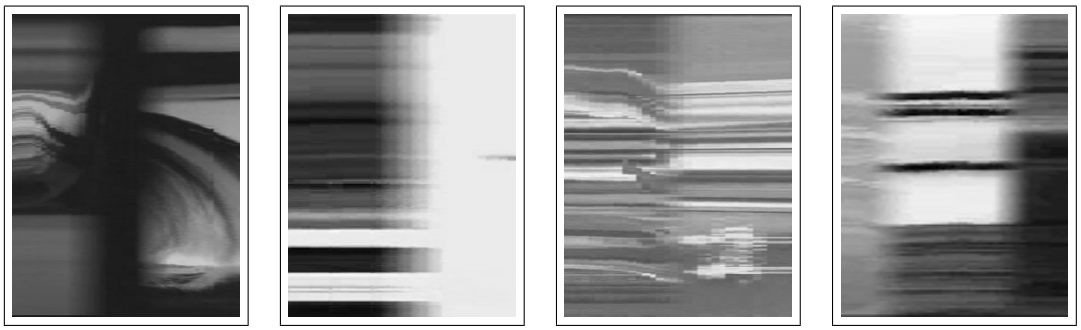
Gradual image transitions. The gradual image transitions can only be found in visual rhythm by sub-sampling (for all types of sub-samplings) and are associated with dissolves and fades. These two types of video events are differently represented on the visual rhythm. While the pattern related to the fade is a gradual image transition between a monochrome region and a non-monochrome region, the dissolve is represented by a transition between two non-monochrome regions. In Fig. 4.6, we show examples of these gradual image transitions.

Light vertical regions. Depending on the change of luminosity in a video, light vertical regions appear on the visual rhythm by sub-sampling. Usually, these regions can represent video effects as the presence of flashes. In Fig. 4.7, we illustrate some examples of light vertical regions.



(a) (b) (c) (d)

Figure 4.5: Examples of sharp transitions in the visual rhythm by sub-sampling (a-c) vertical sharp transitions, and (d) inclined sharp transition.



(a) (b) (c) (d)

Figure 4.6: Examples of gradual transitions present in the visual rhythm.

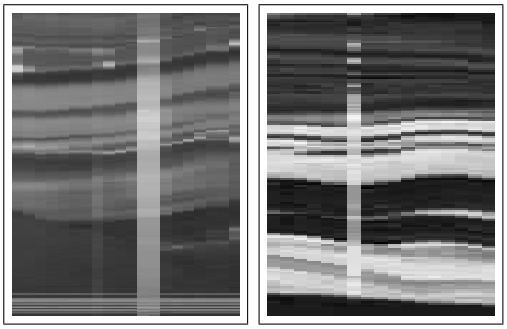
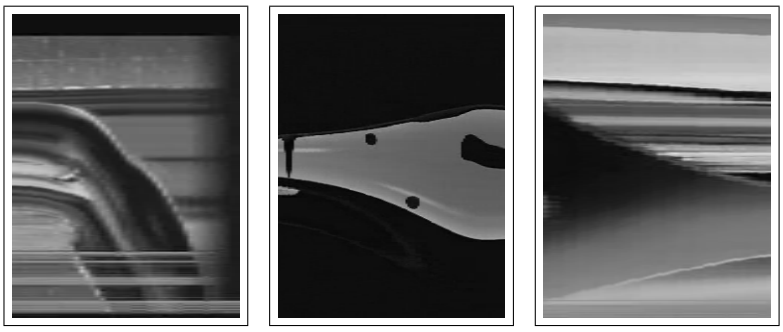


Figure 4.7: Examples of light and thin vertical regions present in the visual rhythm.

Deformed regions. The definition of a deformed region is a high-level concept, and in the visual rhythm by sub-sampling, a deformation is only considered in regard to time axis and can be represented by funneled, expanded and shifted regions. These regions can be related to camera operations like zoom-in, zoom-out and pan, and also to object movements. Some methods which consider these kinds of patterns can be found in [1, 42, 52]. In Fig. 4.8, we illustrate some examples of the deformed regions.



(a) (b) (c)

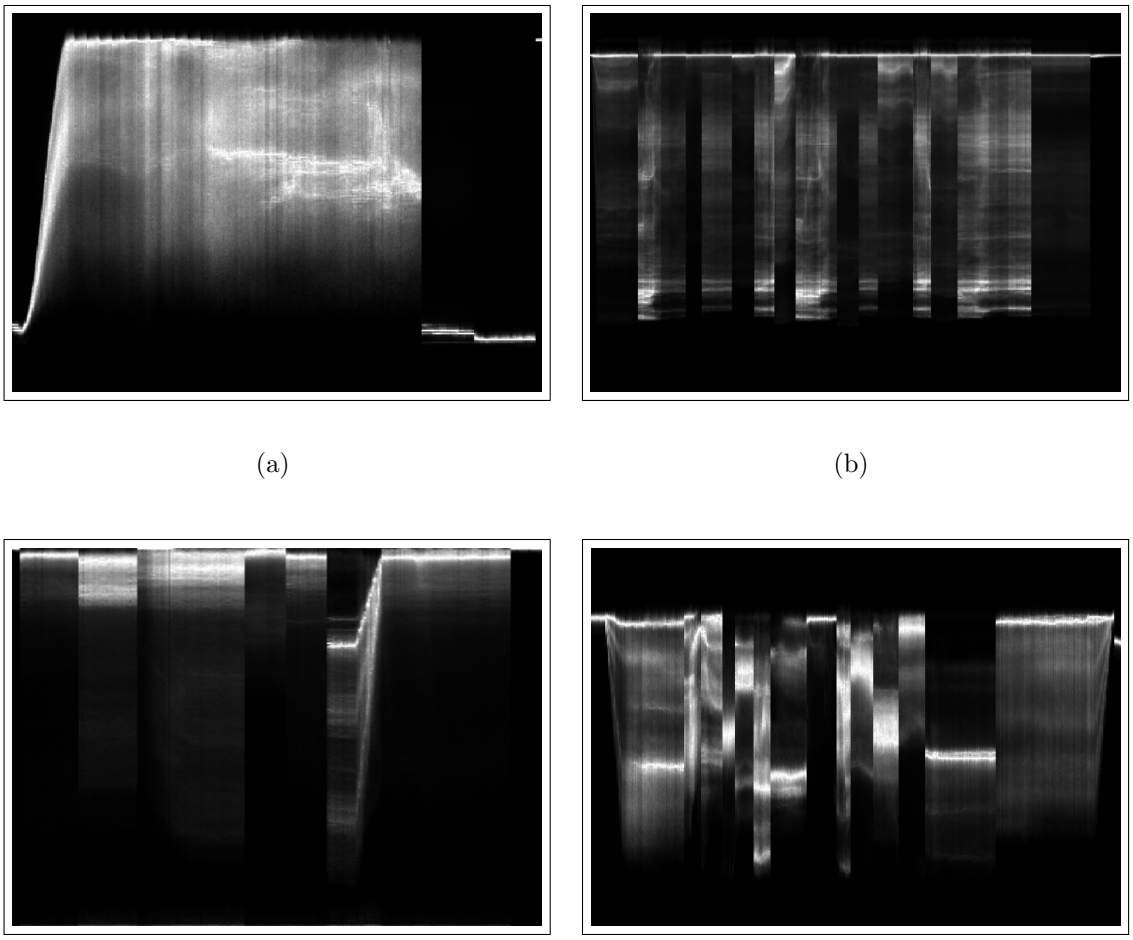
Figure 4.8: Example of deformed regions present in the visual rhythm: (a) shifted region (pan); (b) expanded region (zoom-in); and (c) funneled region (zoom-out).

4.2 Visual rhythm by histogram

To take advantage of the properties of the image histogram, such as global information, invariance to rotation and translation, we define here a new video transformation, called visual rhythm by histogram (VRH), where the video is transformed into a $2\mathbb{D}$ image containing information of the histogram of each frame. Consider a partition of $\mathbb{N}$ in L intervals $I_0 \cdots I_{L-1}$ called bins. A histogram H_{f_t} of an image f_t is given by

$$H_{f_t}(i) = \#\{(x, y) \mid f_t(x, y) \in I_i\}$$

where $\#X$ denotes the cardinality of a set X.



(a) (b)

(c) (d)

Figure 4.9: Real examples of visual rhythm by histogram: (a) video "0094.mpg"; (b) video "0136.mpg"; (c) video "0132.mpg" and (d) video "0131.mpg".

Definition 4.2 (Visual rhythm by histogram (VRH)) *Let* $\mathrm{V} = (f_t)_{t \in [0, duration-1]}$ *be an arbitrary video, in domain* $2\mathbb{D} + t$ *and* $(H_{f_t})_{t \in [0, duration-1]}$ *the sequence of histograms of luminance computed from all frames of* V*. The visual rhythm by histogram* VRH *is a* $2\mathbb{D}$ *representation of all frame histograms where each vertical line represents a frame histogram,* VRH *is defined by*

$$\mathrm{VRH}(t, z) = H_{f_t}(z)$$

where $t \in [0, duration - 1]$ and $z \in [0, L - 1]$, *duration* is the number of frames and L the

number of histogram bins. The main problem of this representation is associated with the transformation of all histogram values into grayscale values. The simplest way for histogram representation is obtained by normalization of each histogram independently. Another possibility is the truncation of the values greater than $G - 1$, where G is the number of grayscales. Due to the loss of information in the representation by truncation, we chose the histogram normalization to represent the visual rhythm by histogram. Furthermore, this normalization produces a filtering effect on the weakest histogram values, which mainly occurs when most pixels are grouped in only few bins. In fact, this kind of filtering is desirable for our application. In Fig. 4.9, we illustrate an example of visual rhythm by histogram where each value of the histogram is in the range $[0, 255]$.

4.2.1 Pattern analysis

In Fig. 4.10 we illustrate the correspondence between the visual patterns and the video events.

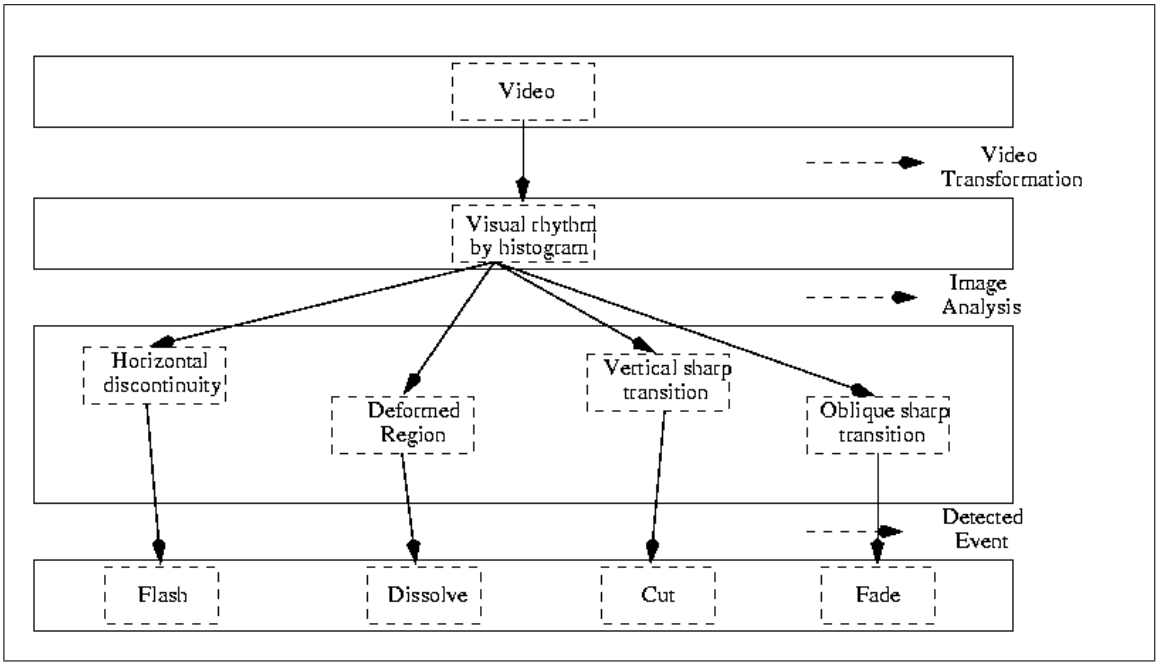


Figure 4.10: Block diagram for video segmentation using visual rhythm by histogram.

Sharp image transitions. Considering the visual rhythm by histogram the correspondence between vertical transitions and cuts is a one-to-one relation. Unfortunately, the

computational time to obtain the visual rhythm by histogram is greater than the visual rhythm by sub-sampling. Also, the fade detection can be associated with the detection of the inclined sharp transitions in the visual rhythm by histogram, but these transitions can also be interpreted as deformed regions. In Fig. 4.11, we illustrate some examples of the sharp image transitions in the visual rhythm by histogram.

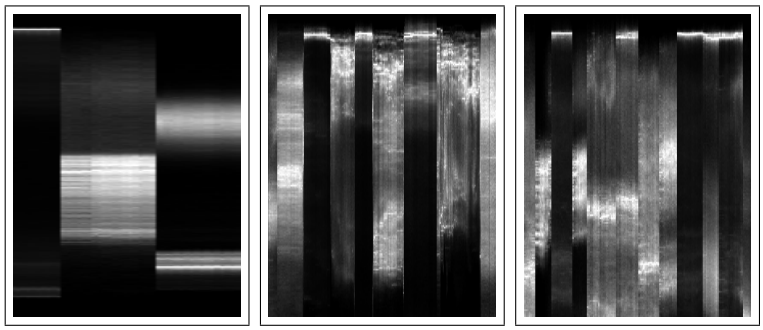


Figure 4.11: Real examples of sharp transitions present in the visual rhythm by histogram.

Orthogonal discontinuities. Usually, the images corresponding to the frames of a same shot have the same characteristics, but the presence of a flash, for example, produces an effect of discontinuity towards the time axis. This effect can be visualized in Fig. 4.12.

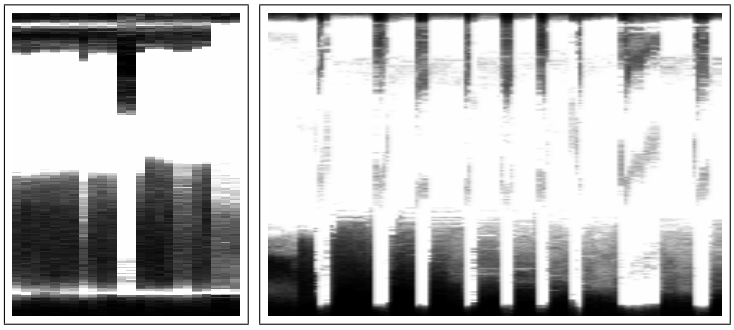
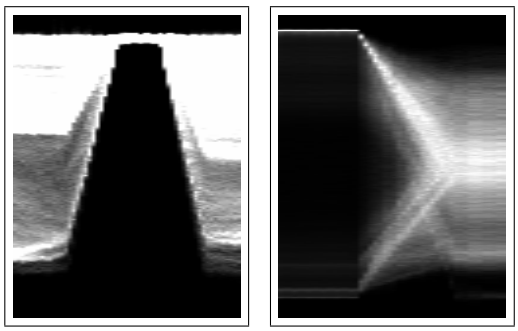


Figure 4.12: Examples of orthogonal discontinuities present in visual rhythm by histogram. Both contain flashes.

Deformed regions. Three types of deformed regions are found: expanded regions, funneled regions and *fuzzy* regions. Differently from the deformed regions in the visual rhythm by sub-sampling which represent camera operations, here they are associated with gradual transitions. The expanded and funneled regions are both related to fade

transitions, and the fuzzy regions correspond to the dissolve transitions. In Fig. 4.13, we illustrate some examples of the deformed regions on the visual rhythm by histogram.



(a) (b)

Figure 4.13: Examples of deformed regions present in the visual rhythm by histogram: (a) corresponds to a fade and (b) to a dissolve.

4.3 Conclusions

According to the video transformation from $2\mathbb{D}+t$ to $1\mathbb{D}+t$, the video events are represented by specific patterns in the image, visual rhythm, obtained by this transformation. In this chapter, we described some of the main patterns present in the visual rhythm. Also, we presented the correspondences between the video events and the different patterns.

Part II

Video transition identification by $2\mathbb{D}$ analysis

Chapter 5

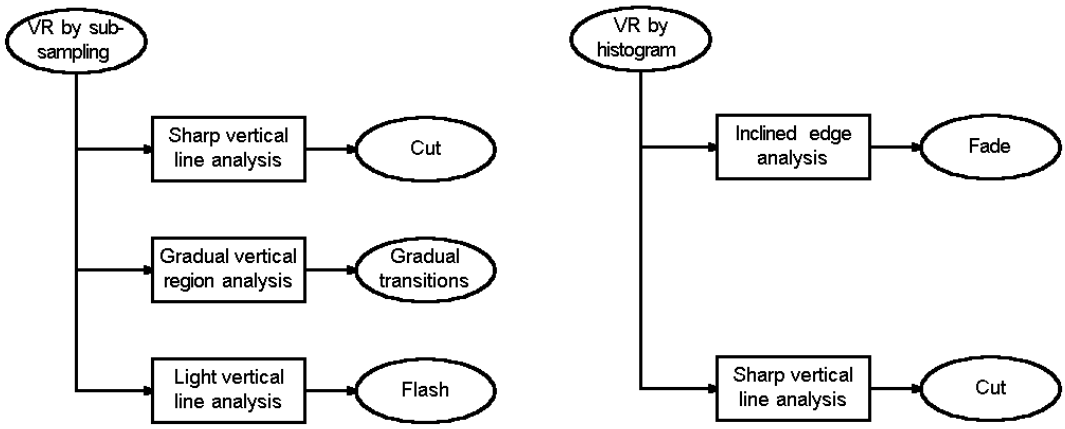
Introduction for video transition identification

The video segmentation problem can be considered a problem of dissimilarity between images (or frames). As described in Chapter 4, an approach to cope with the video segmentation problem is to transform the video into a $2\mathbb{D}$ image, and to apply image processing methods to extract the specific patterns related to each transition (or each video event). In this sense, we summarize the main patterns existing in the visual rhythm by sub-sampling in which we are interested to identify:

- Sharp vertical image transition (SVIT) - the identification of this pattern corresponds to the identification of video cuts;
- Light vertical image transition (LVIT) - the identification of this pattern corresponds to the identification of flashlight (or simply flash) in video;
- Gradual vertical image transition (GVIT) - the identification of this pattern corresponds to the identification of gradual video transitions, like fade and dissolve.

Fig. 5.1(a) illustrates the framework for video segmentation based on visual rhythm by sub-sampling. The identification of SVIT and GVIT is associated with the problem of boundary identification. The boundary identification represents an interesting and difficult problem in image processing, mainly if two flat zones (regions) are separated by a fuzzy region (gradual transition). The most common gradient operators like Sobel and Roberts [28] work well for sharp edges. Unfortunately, the identification of SVIT is not

done by the simple application of these techniques. Thus, we consider a gradient operator to identify sharp transitions as the first step of our methods. Afterwards, some morphological and topological tools are considered to finally identify sharp video transitions. This method is described in Chapter 7. The LVIT disturbs the identification of video transitions. In Chapter 8, we discuss about these patterns proposing two methods to detect them.



(a) (b)

Figure 5.1: Block diagram for event detection from (a) visual rhythm by sub-sampling and (b) by histogram in which we are interested.

The above operators unfortunately fail considerably for gradual transitions. The gradual transitions can be detected by a statistical approach proposed by Canny [13]. Another approach to cope with this problem is through mathematical morphology operators which include, for example, the notion of thick gradient and multi-scale morphological gradient [68]. From this approach, and depending on the size of the transition and its neighboring flat zones, this transition cannot be well detected (as we will see in Chapter 9). To face this problem, we will try to eliminate these gradual transitions by a sharpening process which does not change the number of flat zones of the original image. Thus, in Chapter 9, we propose two different methods to identify gradual video transitions: a method based on a multi-scale analysis and another one based on a sharpening process.

If we consider another video transformation, the visual rhythm by histogram, we can summarize its main patterns:

- Sharp vertical image transition (SVIT) - the identification of this pattern corresponds to the identification of video cuts;
- Inclined lines (IL) - the identification of this pattern corresponds to the identification of video fades.

Fig. 5.1(b) illustrates the framework for video segmentation based on visual rhythm by histogram in which we are interested. To identify the SVIT of this transformation, we can use the same approach used in visual rhythm by sub-sampling. The necessity to identify the IL pattern is that sometimes, we need to classify the gradual video transitions, for that, a method to differentiate a dissolve and a fade is necessary, in this sense, we propose, in Chapter 10, a specific method to identify only fade transitions. Before describing our methods, we introduce some basic image operators in the next chapter.

5.1 Description of the corpora

Nowadays, our video database contains approximately 500 videos with approximately 30 seconds each one, having different video events as zoom, tilt, flash, dissolve, fade, cut, camera motion, object motion, wipe, etc. These videos were downloaded from different *websites*. All videos are stored in MPEG 1 format. For each method, we choose different corpus of videos depending on the video features, this choice was subjective. The number of videos used for each experiment was associated with the quantity of videos with features in which we were interested. This thesis has a CD-ROM containing some of the videos used in the examples.

Chapter 6

Image analysis operators

In this chapter, we describe the basic image operators that are used in this work. These operators are defined in the mathematical morphology, digital topology and discrete geometry domains.

6.1 Mathematical morphology

In this section, we describe some basic morphological operators used in this work (see [68, 63] for more details).

6.1.1 Basic operators

Let B be a flat structuring element (SE) and its homotetic representation ($\lambda B = \{\lambda b \mid b \in B\}, \lambda \geq 0$), such that the size λ represents the radius of the SE.

Definition 6.1 (Morphological gradient [68, 63]) *Let δ_n and ε_n be the dilation and erosion operators with a structuring element B of size n, respectively. The morphological gradient ρ_n of size n is defined by*

$$\rho_n = \delta_n - \varepsilon_n \tag{6.1}$$

The morphological gradient, also called thick gradient, gives the maximum variation of the function in a neighborhood of size n (square or line neighborhood). If n equals the width of the transition between regions of homogeneous grayscale values, the morphological gradient will output the contrast value between these regions. However, the output of this gradient is represented by thick edges.

Another interesting morphological operator that extracts light regions of the image, smaller than a certain structuring element (SE), is the white top-hat, defined next.

Definition 6.2 (White top-hat [68, 63]) *Let γ_n be an opening by a structuring element B of size n ($\gamma_n = \delta_n\varepsilon_n$). The white top-hat WTH_n of size n corresponds to the residue of the opening, and is defined by*

$$WTH_n = Id - \gamma_n \tag{6.2}$$

where Id is the identity operator, "$-$" represents either the binary or grayscale values difference for binary and grayscale images, respectively.

The white top-hat represents the residues of the application of an opening with a specific size of SE. Another operator can be considered if we replace the opening by the erosion in a white top-hat definition. This new operator is called inf top-hat [22].

Definition 6.3 (Inf top-hat [68, 63]) *Let ε_n be an erosion by a structuring element B of size n (ε_n). The inf top-hat ITH_n of size n corresponds to the residue of the erosion, and is defined by*

$$ITH_n = Id - \varepsilon_n \tag{6.3}$$

Another important operator, the ultimate erosion, that will be explored in this work, is associated with the simplification of images. The ultimate erosion indicates the moment in which a connected component disappears. Its definition is given next.

Definition 6.4 (Ultimate erosion [68, 63]) *The ultimate erosion represents the set of all components of an image that disappears from one erosion step to the other, when we consider increasing SE. It is defined for binary and grayscale images as follows.*

$$ULT(X) = \bigcup_n \{{\varepsilon_1}^{(n)}(X) \backslash G_{{\varepsilon_1}^{(n)}(X)}[{\varepsilon_1}^{(n+1)}(X)]\} \tag{6.4}$$

$$ULT(f) = \bigvee_n \{{\varepsilon_1}^{(n)}(f) - G_{{\varepsilon_1}^{(n)}(f)}[{\varepsilon_1}^{(n+1)}(f)]\} \tag{6.5}$$

where G is the morphological reconstruction by dilation operator [68, 63].

If we generalize the concept of white top-hat to a range of SE, then we can define morphological residues, which are described in detail in the next section.

6.1.2 Morphological residues

The morphological residues characterize the information extracted from an image by considering a set of granulometric transformations. The residues are given by the difference between two consecutive granulometric levels, as follows:

Definition 6.5 (Morphological residues [54]) *Let $(\psi_\lambda)_{\lambda\geq 0}$ be a granulometry. The morphological residues, $\mathcal{R}_\lambda$, of residual level λ are given by the difference between the result of two consecutive granulometric levels, that is,*

$$\forall \lambda \geq 1, X \in R^N, \mathcal{R}_\lambda(X) = \psi_{\lambda-1}(X) \backslash \psi_\lambda(X) \tag{6.6}$$

$$\forall \lambda \geq 1, f \in R^N, \mathcal{R}_\lambda(f) = \psi_{\lambda-1}(f) - \psi_\lambda(f) \tag{6.7}$$

where X and f represent binary and grayscale images, respectively. The morphological residues represent the components preserved at level $(\lambda - 1)$ that are eliminated at the granulometric level λ.

According to the transformation ψ, the set of residues corresponding to $(\mathcal{R}_\lambda)_{\lambda\geq 1}$ contains the complete granulometric information and defines a complete hierarchical representation of an image in the sense that the original image can be exactly reconstructed from its residues: for the binary case, we have

$$X = \bigcup_{\lambda\geq 1} \mathcal{R}_\lambda(X) \tag{6.8}$$

and, for the grayscale case,

$$f = \sum_{\lambda\geq 1} \mathcal{R}_\lambda(f) \tag{6.9}$$

An image can be completely decomposed into morphological residues. Here, we consider this decomposition for the morphological gradient image in order to study the behavior of its single domes, that is, we will verify if a single dome in the morphological gradient image corresponds to a n-transition in the original image. To facilitate the analysis of this decomposition, we consider the residue mapping [47].

Definition 6.6 (Residue mapping [47]) *Let g be a $1\mathbb{D}$ image. We denote by $(\mathcal{R}_\lambda)_{\lambda\geq 0}$ the family of morphological residues of g. Let M be a set of regional maxima of $1\mathbb{D}$ image*

g. For all points $p \in M$, we define a residue mapping, $\mathcal{M}$, as follows

$$\mathcal{M}(p) = (\mathcal{M}_1(p), \mathcal{M}_2(p), \cdots, \mathcal{M}_\lambda(p)) \text{ where}$$
$$\mathcal{M}_i(p) = \begin{cases} 1, & \text{if } \mathcal{R}_i(p) \geq 1 \\ 0, & \text{if } \mathcal{R}_i(p) = 0 \end{cases} \tag{6.10}$$

where λ represents the last level in which the morphological residues are greater than zero.

6.1.3 Multi-scale gradient

A gradient model proposed by Soille can be represented by the block diagram in Fig. 6.1. According to this model [68], the problem of thickness introduced by the morphological gradient is avoided by the application of erosions on the thick image. To avoid the merge of boundaries, a white top-hat is applied to the thick gradient before the erosion operation.

Definition 6.7 (Soille's morphological gradient [68]) *The morphological gradient at scale n is given by:*

$$\rho_n^S = \rho_n \times \; L_{[1]}\varepsilon_{(n-1)}\, WTH_n\rho_n \tag{6.11}$$

where ρ_n, $\varepsilon_{(n-1)}$ and WTH_n represent the thick gradient of size n, the erosion of size $n-1$ and the white top-hat of size n, respectively. $L_{[1]}$ represents a thresholding operation. To compute a multi-scale gradient at scale n, it is necessary to calculate the supremum of the gradient values at scale within the range $[1, n]$.

For the application considered here, the problems of this approach are directly related to the choice of the SE family. The first problem we can identify is related to the quality of the detection. For example, let us consider a $1\mathbb{D}$ image (Fig. 6.2(a)) and a homotetic family of SE. If we apply the Soille's gradient on such configuration the transition is not well identified for the corresponding scale (Fig. 6.2(b) and (c)). The second problem concerns the elimination of regions due to the introduction of the erosion operation. Finally, from this approach, it is not possible to classify the detected transitions according to a parameter of size, i.e., we can not identify the transitions of a specific size k because these transitions, identified at a certain scale i, correspond to the transition size smaller than i. Intuitively, this problem can be avoided by the difference between two consecutive levels,

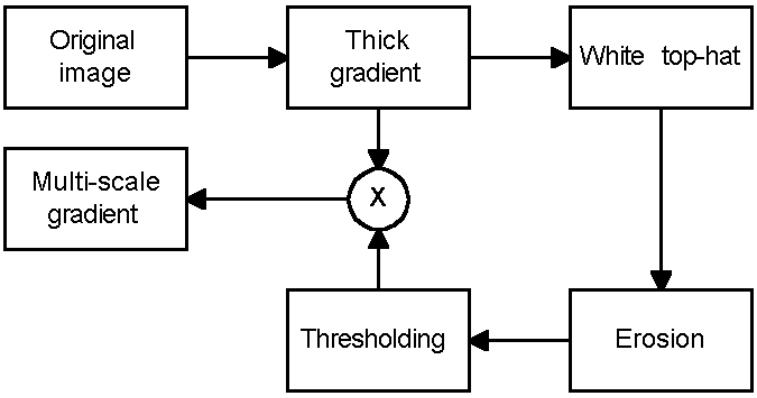


Figure 6.1: Soille's gradient.

but the gradient values depend on the thick gradient that may vary at consecutive scales. In Fig. 6.3(b) we show the result of the Soille's gradient applied to the image in Fig. 6.3(a). Here, the parameter of size is in the range $[1, 7]$, and the final result corresponds to the supremum of the gradient values at all levels.

6.1.3.1 Multi-scale gradient based on ultimate erosion

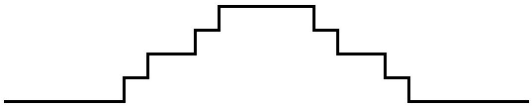
Now, we replace the erosion in Fig. 6.1 by the ultimate erosion to try to eliminate some undesirable components.

Definition 6.8 (Gradient based on ultimate erosion) *Our proposed morphological gradient at scale n is defined by:*

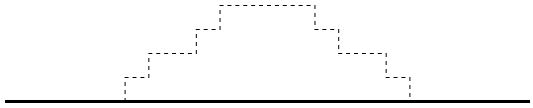
$$\rho_n^U = \rho_n \times \; L_{[1]}\, ULT(\, WTH_n \rho_n) \tag{6.12}$$

To compute a multi-scale gradient at scale n, it is necessary to calculate the supremum of the gradient values at scale within the range $[1, n]$. In Fig. 6.2(d) and (e) we illustrate the application of the multi-scale gradient based on ultimate erosion, considering a certain scale and a range of scales, respectively.

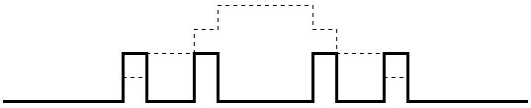
The problem identified here is associated with noise sensitivity, and with the fact that the transitions are not thin. Also, it is not possible to classify the transitions regions according to size criterion. In Fig. 6.3(c), we show the result of the multi-scale gradient based on ultimate erosion of the image in Fig. 6.3(a). The parameter of size is in the range $[1, 7]$, and the final result corresponds to the supremum of the gradient values at all levels.



(a) Original.

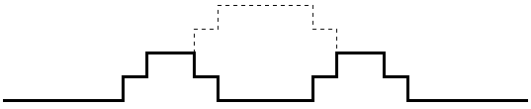


(b) Soille's gradient.



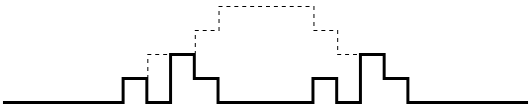
(c) Soille's gradient.

(d) Gradient based on ultimate erosion.



(e) Gradient based on ultimate erosion.

(f) Gradient based on thinning.



(g) Gradient based on thinning.

Figure 6.2: Morphological multi-scale gradient: (a) original $1\mathbb{D}$ image; (b,d,f) correspond to the gradient values at a specific level $n = 4$ and (c,e,g) correspond to the supremum of the gradient values at different levels ($n = [1, 5]$) .

Next, we propose a variant that replaces the ultimate erosion by a thinning operator.

6.1.3.2 Multi-scale gradient based on thinning

While the erosion operation may produce thick edges, the thinning transformation defines one-pixel-thin edges. Another feature of this operator is that it preserves topology in the sense of [5].

Definition 6.9 (Gradient based on thinning) *Our proposed morphological gradient at scale n is defined by:*

$$\rho_n^T = \rho_n \times \ L_{[1]}T(\ WTH_n\rho_n) \tag{6.13}$$

where T represents the thinning operator described in Section 6.2. To compute a multi-scale gradient at scale n, it is necessary to calculate the supremum of the gradient values at scale within the range $[1, n]$.

In Fig. 6.2(f) and (g) we illustrate the application of the multi-scale gradient based on thinning, considering a certain scale and a range of scales, respectively.

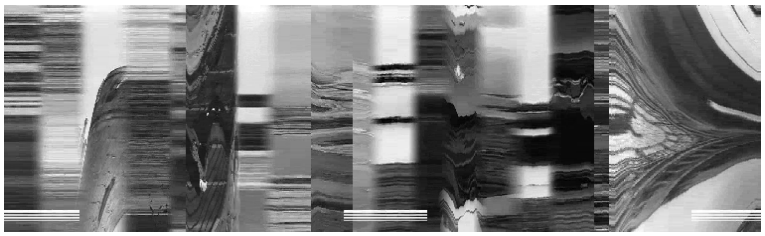
The problem of this approach is the high noise sensitivity. In Fig. 6.3(d), we illustrate the gradient based on thinning applied to the image in Fig. 6.3(a). Here, the parameter of size is in the range $[1, 7]$ and the final result corresponds to the supremum of the gradient values at all levels.

6.2 Thinning

In this section, we describe the basic topological operator used in this work (see [5] for more details). Let us consider a point x in a $1\mathbb{D}$ image (or signal) g. We say that a point x is destructible for g, if one neighbor of x has a value greater than or equal to $g(x)$ and the other neighbor has a value strictly smaller than $g(x)$. The thinning procedure consists in repeating the following steps until stability:

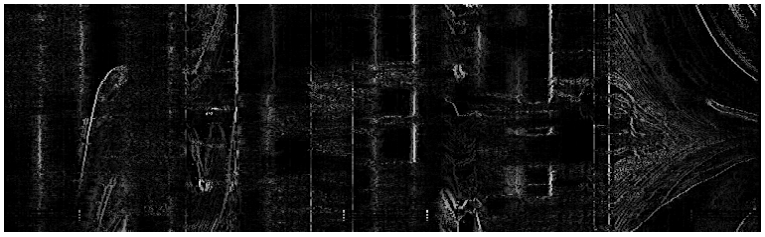
i select a destructible point x;

ii lower the value of x down to the value of its lowest neighbor.



(a) Original image.

(b) Soille's multi-scale gradient.

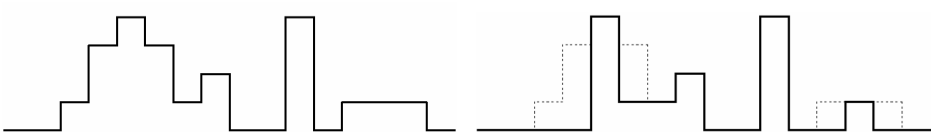


(c) Multi-scale gradient based on ultimate erosion.

(d) Multi-scale gradient based on thinning.

Figure 6.3: Examples of multi-scale gradients where the SE size is in range $[1, 7]$. These results correspond to the supremum of gradient values of all levels.

In Fig. 6.4, we illustrate an artificial example of thinning. Selection of destructible points must be done in increasing order of value, so that each point is modified at most once. Points having the same value are scheduled with a fifo policy which guarantees that, in case of large flat maxima, the thinned image is "well centered" with respect to the original one. This procedure is in fact a particular case, in $1\mathbb{D}$ domain, of a topological operator introduced in [5]. Topological operators aim to simplify the image while preserving the topology. [5] presents operators for image segmentation based on topology which generalizes to $2\mathbb{D}$ grayscale images the notions of binary digital topology [45].



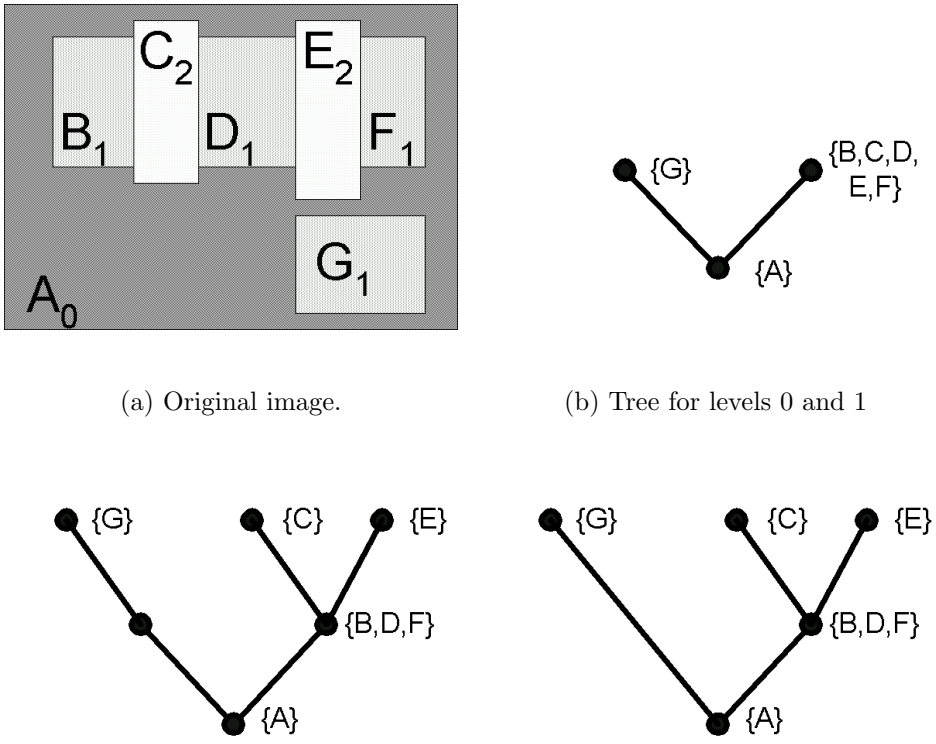
(a) Original (b) Result

Figure 6.4: $1\mathbb{D}$ image thinning example. Dotted line, in (b), represents the original image (a).

6.3 Max-tree

In this section, we describe a structure that deal with the problem of efficiently filtering images. This structure is called *Max-tree* [62]. The idea consists in recursively creating a tree, the nodes of which are the connected components of the sections (thresholds) of the image at each possible level. In Fig. 6.5 we illustrate an example of the max-tree creation process.

The original image (Fig. 6.5(a)) is composed of seven at zones identified by a letter {A, B, C, D, E, F, G}. L_i means that the value of gray level of the region L is i. In this example, the gray level values range from 0 to 2. In the first step, we apply a thresholding operation at level 0, resulting in a binary image, in which the all pixel at level $h = 0$ (pixels of region A) are assigned to the root node of the tree (Fig. 6.5(b)). Furthermore, the pixels of gray level value strictly higher than h = 0 form two connected components that are



(a) Original image.

(b) Tree for levels 0 and 1

(c) Tree for levels 0, 1 and 2

(d) Final tree

Figure 6.5: Process of max-tree creation [62]: (a) original image; (b) first step of the process considering the levels 0 and 1; (c) second step considering the levels 0, 1 and 2; and (d) the final tree.

temporarily assigned to two nodes: {B, C, D, E, F} and {G} (Fig. 6.5(b)). In a second step, the threshold is increased by one ($h = 1$). Each children node of the root is processed separetely (as in first step) (Fig. 6.5(c)). In summary, at each temporary node, a local background is defined by keeping all pixels of gray level value equal to h.

In this procedure, some nodes may become empty. Therefore, at the end of the tree construction, the empty nodes are removed. The final tree is called a *Max-Tree* in the sense that it is a structured representation of the image which is oriented towards the maxima of the image (maxima are simply the leaves of the tree). In our example the final tree is illustrated in Fig. 6.5(d).

6.4 Conclusions

In this chapter, we enumerated some basic image operators that constitute the basis for the development of video transition identification methods presented in this work. To apply the mathematical morphology operators, we consider that the structuring elements (SE) are flat and $1\mathbb{D}$. The direction of the this SE depends on the method and the purpose of its application (as we will see in the next chapters).

Chapter 7

Cut detection

The video segmentation problem is transformed into a problem of pattern detection, where each video effect is transformed into a different pattern on the visual rhythm. To detect sharp video transitions (cuts) we use topological and morphological tools instead of using a dissimilarity measure. We propose a method to detect sharp video transitions between two consecutive shots. We present a comparative analysis of our method with respect to some other methods.

7.1 Introduction

The simplest transition between two consecutive shots is the sharp transition (cut) that is simply a concatenation of these shots. In Fig. 7.1, we illustrate an example of cut. The common approach to cope with the cut detection is based on the use of a dissimilarity measure. [72] and [7] review some of the most popular methods for cut detection, such as pixel-wise comparison and histogram comparison. Unfortunately, the cut detection is complicated by the presence of effects, like gradual transitions, flashes and fast camera and object motions.

Figure 7.1: Cut example.

Another approach to solve the video segmentation problem is to transform the video sequence V into a $2\mathbb{D}$ image VR, called visual rhythm, and to apply image processing methods on VR to extract the different patterns related to each transition. As it was illustrated in Chapter 4, each frame is transformed into a vertical line in VR.

We propose, in this work, a method for video segmentation based on analysis of a $2\mathbb{D}$ image. Here, we consider also the notion of visual rhythm by histogram, and we use it to detect different kinds of transition. On these two variants of visual rhythm, namely visual rhythm by sub-sampling and visual rhythm by histogram, we apply morphological and topological tools to segment the video without the need of defining a dissimilarity measure between frames. The simplicity of implementation, the low processing cost and the high quality of results can be considered as the main qualities of this method for cut detection. Also, we verified that our methods are more robust than other implemented methods with respect to the tuning of threshold values. The fact that two different video events may be represented by the same visual rhythm pattern can be considered as the main drawback of our method. Fortunately, this problem is not frequent in real cases.

This chapter is organized as follows. In Sec. 7.2 we present our methodology for cut detection. In Sec. 7.3 we show a comparative analysis for cut detection involving our method and some other methods, using four different quality measures. According to these measures, we can verify that our method presents generally the best results. Some conclusions are given in Sec. 7.4.

7.2 Our method

With the aim of perfoming a video segmentation without defining a dissimilarity measure, we use a simplification of the video content, the visual rhythm, where the video segmentation problem, in domain $2\mathbb{D} + t$, is transformed into a problem of pattern detection, in domain $1\mathbb{D} + t$. So, we can apply methods of $2\mathbb{D}$ image processing to identify different patterns on the visual rhythm because each video effect corresponds to a pattern in this image, for example, each sharp video transition is transformed into "vertical lines" on the visual rhythm. This correspondence is not a one-to-one relation, i.e., a sharp video transition corresponds to a vertical line, but a vertical line is not necessarily a sharp video transition, as illustrated in Chapter 4. This problem can be solved considering

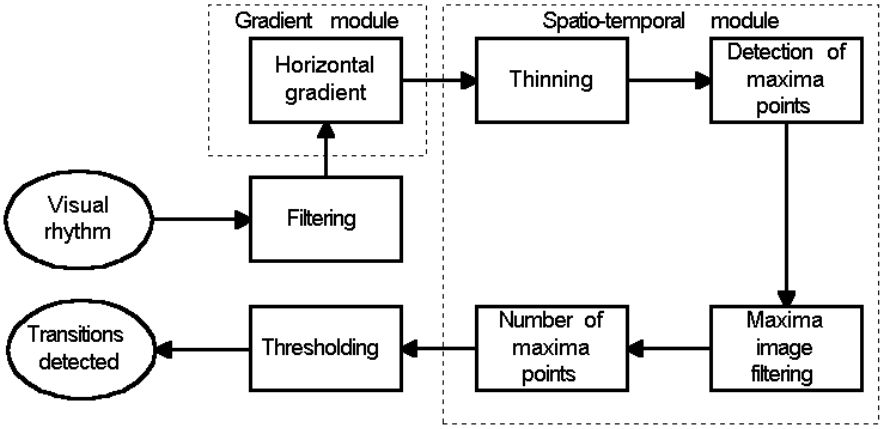


Figure 7.2: Cut detection block diagram.

visual rhythms obtained from different pixel samplings. Afterwards, a simple intersection operation between these results may be used to identify correctly the sharp video transitions.

Fortunately, in general, we can use only a visual rhythm obtained from principal diagonal sampling because this problem rarely occurs in practice. Furthermore, this visual rhythm represents the best simplification of the video content, according to [17]. Next, we will define a method for cut detection based on visual rhythm that is illustrated in Fig.7.2.

Visual rhythm computation. The visual rhythm by sub-sampling is computed from the principal diagonal of each frame. Also, it is possible to use the visual rhythm by histogram to detect the cuts. In Fig. 7.3, we illustrate an example of visual rhythm by principal diagonal sub-sampling.

Filtering. In this step, we reduce noise on VR using mathematical morphology filters (see [64], [68]). The filtered visual rhythm is denoted by VR_F. Here, we apply an opening (closing) by reconstruction to eliminate the small light (black) components. These morphological filters have the interesting property of preserving the sharp contours of the image. In Fig. 7.4, we illustrate the result of this filtering when applied to image illustrated in Fig. 7.3.

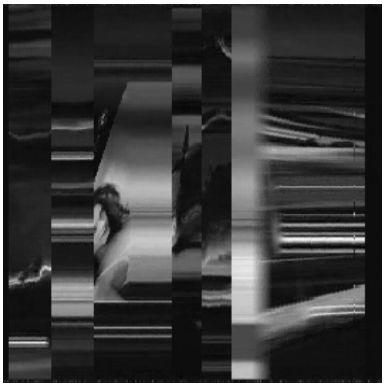


Figure 7.3: Visual rhythm by principal diagonal sub-sampling computed from the video "132.mpg".

Horizontal gradient. The goal of this step is to detect the locations where horizontal grayscale discontinuities occur in the filtered visual rhythm. These locations, when vertically aligned, can represent a cut. So, we calculate the norm of the horizontal gradient ∇_h of the filtered image by

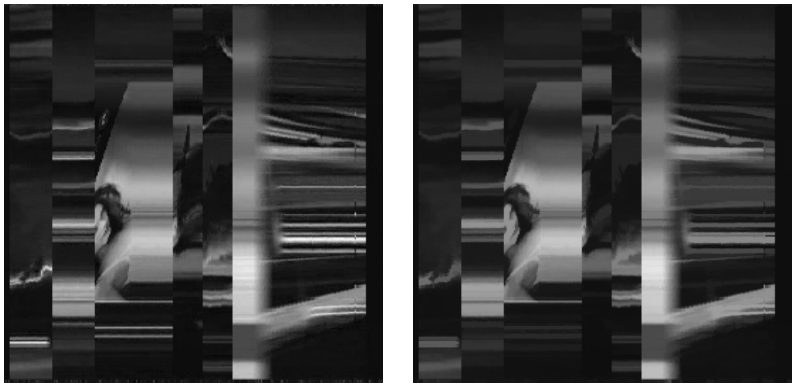
$$|\nabla_h \mathrm{VR}_F(t,z)| = |\mathrm{VR}_F(t,z) - \mathrm{VR}_F(t-1,z)|. \tag{7.1}$$

In Fig. 7.5, we illustrate the result of the computation of the horizontal gradient.

Thinning. This transformation is used here to simplify the peak detections (see [5]). Intuitively, a horizontal transition between two consecutive regions corresponds to a "peak" in the horizontal gradient of each line. In the case of a cut, the maximum of this peak is generally reduced to only one pixel but for gradual video transitions, for example, the maximum of a peak may consist of several neighboring pixels. In such cases, a simple maximum detection would result in multiple responses for a single transition. This is why we introduce the thinning step, with the aim of reducing every peak to a one-pixel-thin maximum.

In Fig. 7.6, we illustrate the result of the thinning when applied to horizontal gradient image illustrated in Fig. 7.5.

Detection of the local maximum points. After the thinning operation, we have a new image $\mathrm{I_T}$ where each horizontal peak is represented by a point, called maximum



(a) (b)

Figure 7.4: Visual rhythm filtered in which the small components are eliminated: (a) the original visual rhythm and (b) the result of the morphological filtering (reconstructive opening followed by a reconstructive closing). The radius size of the horizontal structuring element is 4.

point. A point x in $1\mathbb{D}$ image g_t is maximum if its two neighbors have values strictly smaller than $g_t(x)$. So, we must find all maximum points of the image $\mathrm{I_T}$ to identify the center points of the transitions. The image of maximum points is denoted by M.

In Fig. 7.7, we illustrate the result of the identification of maximum points of the thinned image (Fig. 7.6(b)).

Maxima image filtering. The locations of the cuts appear as "vertical lines" in image M, embedded in irrelevant components (noise) which can be reduced by a morphological filtering. This filtering is an opening by reconstruction [68, 63] with a vertical structuring element of size $\lambda = 7$, defined empirically. The filtered maximum image is denoted by M_F.

In Fig. 7.8, we illustrate the result of the filtering of maximum points.

Detection of cuts. From the filtered maximum image M_F, we create a $1\mathbb{D}$ image P where each point t has a value $\mathrm{P}(t)$ representing the number of maximum points of the vertical line t on M_F. Finally, when the value of the point $\mathrm{P}(t)$ is greater than or equal to a threshold T, then a cut is detected at time t.

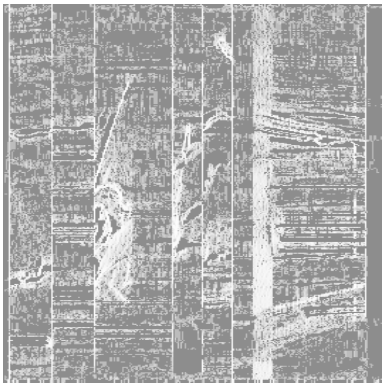


Figure 7.5: Horizontal gradient image. To facilitate the visualization, we apply an operator to equalize the image histogram.

In Fig. 7.9 (Fig. 7.10), we illustrate the results of the main steps of our method when applied to a visual rhythm by sub-sampling (or histogram) obtained from the same real video.

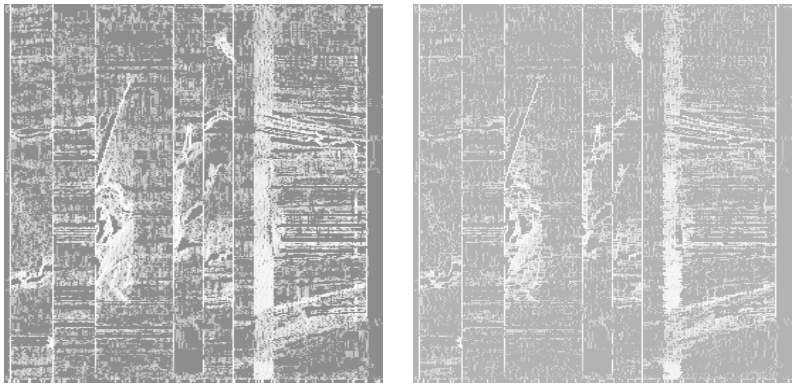
7.3 Experiments

In these experiments, we implemented three methods described in literature: a variant of pixel-wise comparison, a histogram intersection and a statistical technique based on visual rhythm. We chose these methods due to their simplicity and their effectiveness according to [22], [10] and [17], respectively. We also implemented the proposed method with some variations. Next, we describe all experiments applied to Corpus 1, identified in Table 7.1, followed by a global analysis of the results.

	Videos	Cuts	Fades	Flashes	Frames
Corpus 1 (Cut)	32	778	15	14	29933

Table 7.1: Video features selected for experiments.

Experiment 1. This experiment uses the difference between pixels according to [22] as the dissimilarity measure. A $1\mathbb{D}$ image is created from the dissimilarity values calculated in each frame of the video. Then a mathematical morphology operator, called inf top-hat, is applied to this image, and finally, a threshold is used to detect the cuts.



(a) (b)

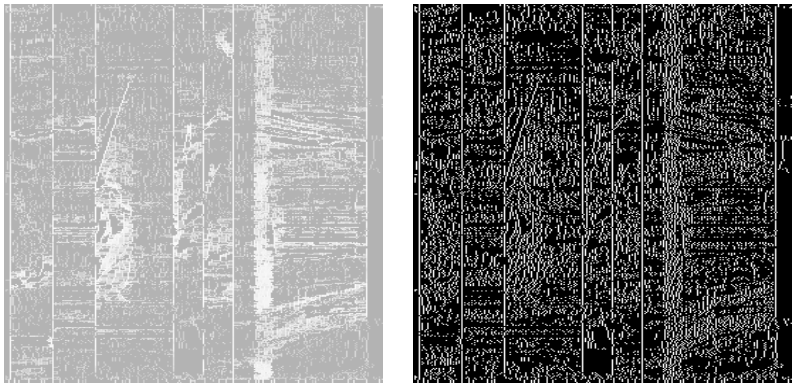
Figure 7.6: Thinning operation: (a) the equalized horizontal gradient image and (b) the result of the thinning. To facilitate the visualization, we apply an operator to equalize the histogram.

Experiment 2. This experiment uses the histogram intersection according to [10] as the dissimilarity measure. If the dissimilarity value is greater than a threshold, then a cut is detected. With the aim of improving the results, a subdivision of each frame is made.

Experiment 3. This experiment uses the visual rhythm for cut detection based on statistical method as described in [17]. Here, the parameters are different from those used in the other mentioned methods, particularly the threshold. While in this method the threshold is locally adaptive and related to a parameter that varies from 1 to 10, in the other methods the threshold is fixed and global.

Experiment 4. In this experiment, that is a variant of our method introduced in Sec. 7.2, we compute a $1\mathbb{D}$ image associated with the mean of the difference between pixels in consecutive frames. We apply the following algorithm to this image: i) apply a white top-hat by reconstruction with a flat structuring element of size 3; ii) apply a thinning; and iii) apply a thresholding.

Experiment 5. In this variant of our method introduced in Sec. 7.2, instead of applying the summation of the number of maximum points to each vertical line, we use the filtered



(a) (b)

Figure 7.7: Detection of maximum points: (a) the equalized thinning image and (b) the maximum points.

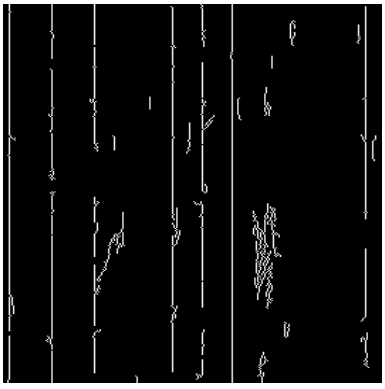


Figure 7.8: Filtering of the maximum points.

maxima M_F image as a mask to select the grayscale values associated with all maximum points. Afterwards, we find the mean of these grayscale values in each vertical line. Then, a thresholding is applied to these results.

Experiments 6 and 7. These experiments are related to the method defined in Sec. 7.2 concerning the analysis of the visual rhythm by sub-sampling and by histogram, respectively.

In Fig. 7.11, we show graphically the experimental results for each experiment previously described. The graphics relate the threshold (rate with respect to the maximum value obtained from each experiment) to recall and error rates. From the curves shown

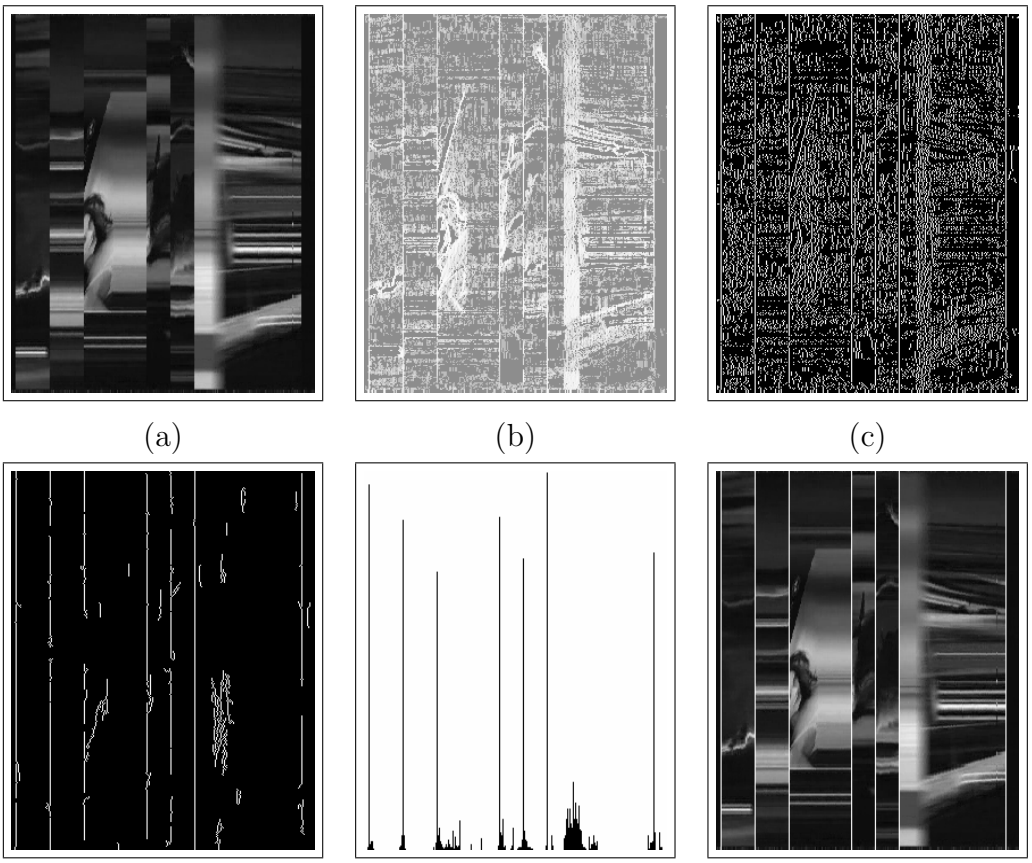


Figure 7.9: Cut detection from a visual rhythm by sub-sampling: (a) visual rhythm; (b) thinning of the horizontal gradient (equalized); (c) maximum points; (d) maxima filtering; (e) normalized number of maximum points in the range $[0, 255]$; (f) detected cuts (white bars) superimposed on the visual rhythm.

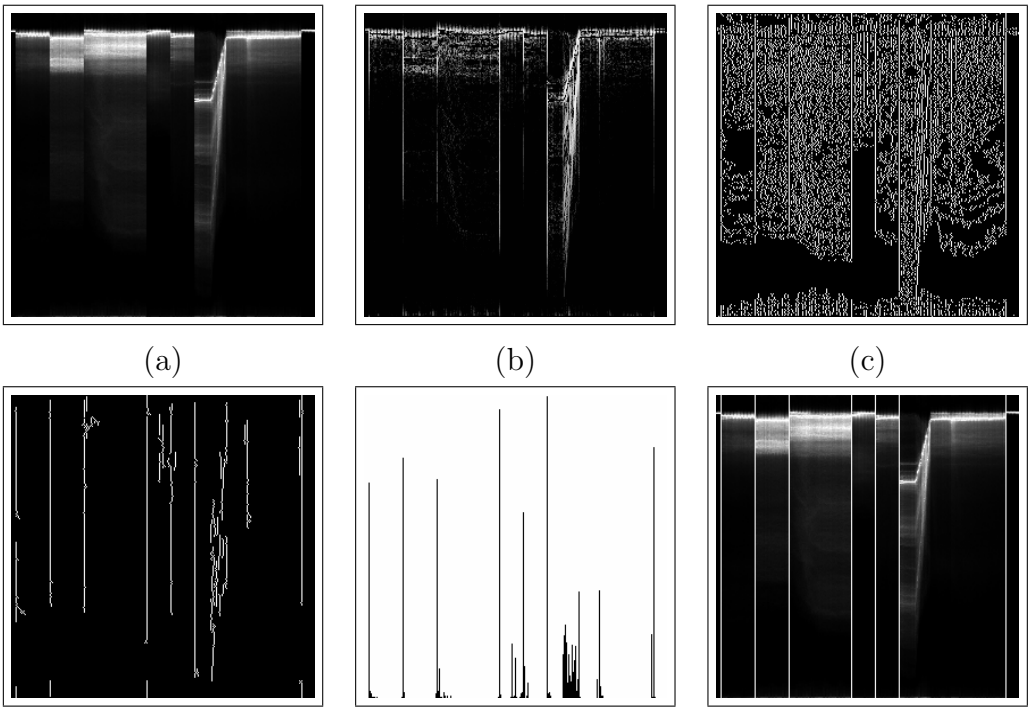


Figure 7.10: Cut detection from a visual rhythm by histogram: (a) visual rhythm; (b) thinning of the horizontal gradient; (c) maximum points; (d) maxima filtering; (e) normalized number of maximum points in the range $[0, 255]$; (f) detected cuts (white bars) superimposed on the visual rhythm.

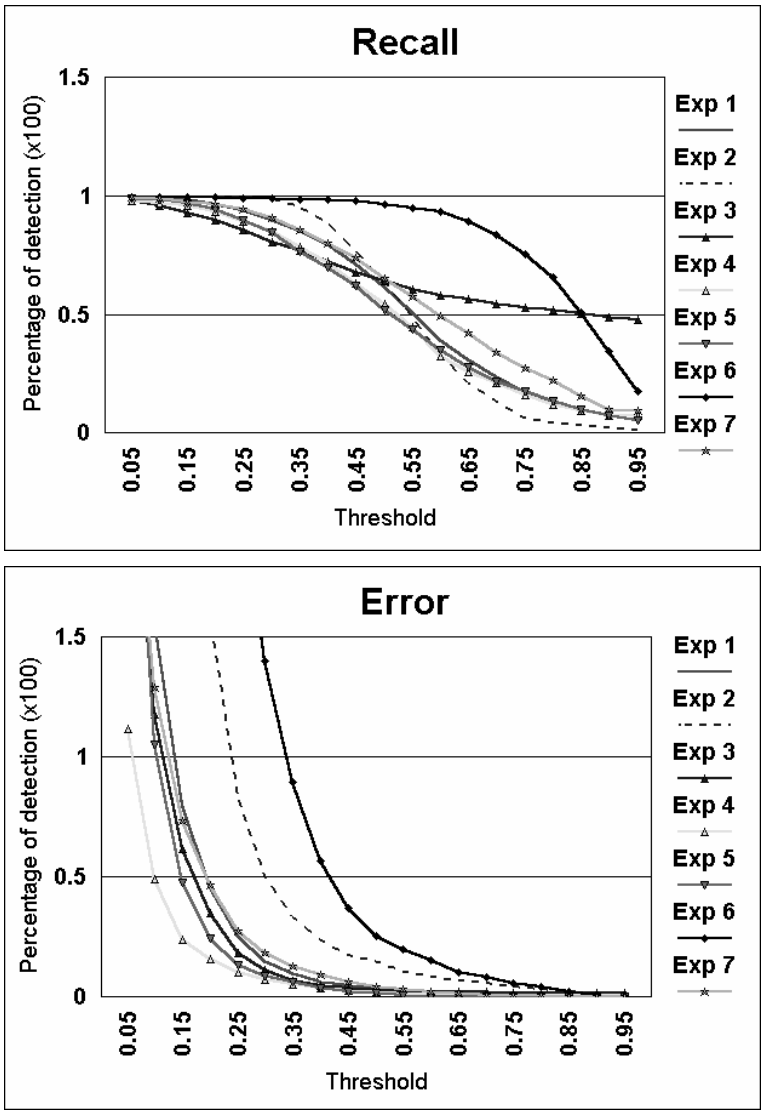


Figure 7.11: Experimental results.

in these graphics, it is possible to find the robustness μ, missless error rate E_m, falseless recall rate R_f and gamma γ measures, that are outlined in Table 7.2. According to Sec. 3.3, we expect high values for μ, γ and R_f, and small values for E_m. In Table 7.3, we illustrate the basic quality measures when we consider the values associated with the gamma measure, i.e., the gamma measure corresponds to the maximal value of a function (Eq. 3.7), this value is related to the recall and error for all values of thresholds, thus we represent in Table 7.3, the values of recall, error and the threshold used to compute the gamma measure. From these experiments, we can verify that the proposed method based on visual rhythm analysis generally produces the best results, mainly according to the robustness and the missless error rate. The good value of the robustness means that the proposed method is not very sensitive to small variations around an "optimal value" of the main parameter. Another interesting aspect of our method concerns the missless error rate since, in general, we want results without miss and with the smallest possible ratio of false detections, so that we can eliminate them posteriorly. Indeed, a post-processing is essential to increase the quality of the results because there are many false detections due to the presence of effects like flash, pan, zoom. Also, we can observe that the processing time for experiments based on visual rhythm by sub-sampling is significantly lower than for the experiments applied directly to the video. We notice that the results of the proposed method based on analysis of the visual rhythm by sub-sampling are better with respect to the visual rhythm by histogram.

	μ	E_m	R_f	γ
Experiment 1	0.01	0.80	0.10	0.77
Experiment 2	0.00	0.51	0.00	0.68
Experiment 3	0.00	1.20	**0.51**	0.72
Experiment 4	0.06	0.49	0.21	**0.80**
Experiment 5	0.01	0.48	0.44	0.78
Experiment 6	**0.11**	**0.37**	0.35	**0.80**
Experiment 7	**0.11**	0.46	0.42	0.75

Table 7.2: Quality measures: robustness ($\mu(0.10, 0.30)$), missless error ($E_m(0.03)$), falseless recall ($R_f(0.01)$) and gamma measure (γ). The best values are highlighted.

Exp	Type	Cut	Detected	False	Recall	Precision	Error	Threshold
1	Commercial	778	662	78	85.1%	89.5%	10%	35%
2	Commercial	778	690	179	88.2%	79.4%	23.8%	40%
3	Commercial	778	598	55	76.8%	52.3%	6.9%	4
4	Commercial	778	697	78	89.6%	90%	10.3%	25%
5	Commercial	778	699	101	89.8%	87.4%	13.6%	25%
6	Commercial	778	697	78	89.6%	90%	10.3%	65%
7	Commercial	778	696	125	89.8%	85%	15.8%	65%

Table 7.3: Basic quality measures related to the gamma measure.

7.4 Conclusions and discussions

A method for cut detection was proposed. The main contribution of this chapter is the application of operators of mathematical morphology and digital topology to solve a problem of video segmentation.

The quality of the results is associated with the choice of good parameters. Two types of parameters can be distinguished: fixed and variable. The fixed parameters, like size of structuring elements, can be pre-determined for all applications. The use of a variable parameter, in our case, the threshold value, is interesting and sometimes necessary, because it plays an important role in the segmentation process where the user can adequate it according to the nature of data and type of application. Also, the tuning of this parameter allows to find a compromise between over-segmentation and under-segmentation.

To do a comparative analysis between different methods for event detection, we used four quality measures: robustness, missless error, falseless recall and the gamma measure. According to these quality measures, we verified that the proposed method for cut detection has the best values of robustness, missless error and gamma measure, when compared experimentally to the other methods.

From this method, we observed that the visual rhythm by sub-sampling and by histogram present an adequate simplification of the video content, which can constitute the basis for other developments such as:

1. identify some other video events, like flash, pan and zoom from the detection of

their correspondent patterns;

2. modify the proposed method for cut detection to detect gradual video transitions, using the multi-scale morphological gradient ([68]) to compute the horizontal gradient (as we will see in Chapter 9).

Chapter 8

Flash detection

The flash presence is very common in digital videos, mainly in television journal videos. When a flash occurs, an increase of the luminosity in few frames is produced. When we calculate a dissimilarity measure, we can see that this measure is very high in the frames affected by a flash. In fact, the presence of flashes often disturbs the cut detection. In this work, we propose two methods for flash detection without taking into account dissimilarity measures. The first one is a variant of the proposed method for cut detection and the second one considers a filtering of the max tree [62] calculated from statistical measures of each frame (or frame sub-samplings).

8.1 Introduction

The identification of flashes can be considered as an essential step to reduce the number of false detections when we try to identify shot boundaries, mainly cuts. Another application concerning the identification of flashes is associated with an important event, like when we have the President of Brazil making a speech, many photos are taken, and consequently, many flashes are dispatched. An interesting approach can be found in [78], in which a flash and a cut are distinguished according to some defined models. In this method, it is computed the average intensity of each frame. A flash is identified if the relation between different frames (sliding window) is verified. This method fails when a flash occurs at the same time of a cut. In Fig. 8.1, we illustrate two different models for a flash: Fig. 8.1(a) represents a flash that occurs in the middle of a shot; and Fig. 8.1(b) represents a flash

that occurs in the boundary of a shot.

(a) (b)

Figure 8.1: Flash model: (a) flash occurrence in the middle of the shot and (b) flash occurrence in the boundary of the shot.

In this work, we propose two different methods to identify flashes, both from visual rhythm by sub-sampling: top-hat based and max-tree based. The first one can be related to the method proposed by [78] in the sense of using a neighborhood, but the principles are different, and we consider the values of the pixels without computing the statistical values. The second method is more sophisticated in the sense of identifying all types of flashes.

This chapter is organized as follows: in Sec. 8.2, we propose a method based on top-hat filtering. In fact, this method is a variant of the cut method. In Sec. 8.3, another method is proposed, in this case, a filtering of the statistical values is considered. In Sec. 8.4, some experiments are shown. Finally, some conclusions are given in Sec. 8.5

8.2 Based on top-hat filtering

On the visual rhythm, we can observe that the video flashes are transformed into thin light vertical lines, as shown in Fig. 8.2. So, we can easily extract these lines from a white top-hat by reconstruction. The white top-hat by reconstruction is a mathematical morphology operator and represents the difference between the original image g and the opening by reconstruction of g [64, 68]. Informally, this operator detects light regions according to the shape and the size specifications of the structuring element. Our first method for flash detection is described below and is illustrated as a block diagram in Fig. 8.3.

Figure 8.2: Flash video detection: (a) some frames of a sequence with the flash presence; (b) visual rhythm by sub-sampling; (c) detected flash.

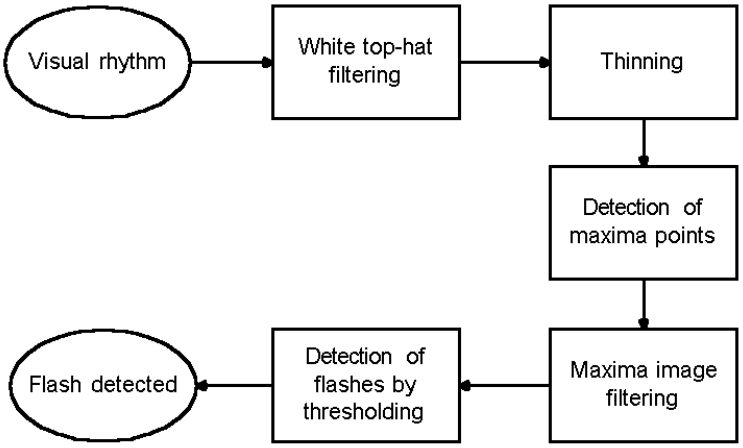


Figure 8.3: Flash detection block diagram using top-hat filtering.

Visual rhythm computation. The visual rhythm by sub-sampling is computed from the principal diagonal of each frame. In Fig. 8.4, we illustrate the visual rhythm by principal diagonal sub-sampling computed between the frames 740 and 800 from the video "0600.mpg".

White top-hat filtering. Applying the white top-hat by reconstruction with a linear horizontal structuring element of size $\lambda = 5$. This size is associated with the potential duration of a flash. In Fig. 8.5, we illustrate the white top-hat and the result of the histogram equalization of the white top-hat to facilitate its visualization.

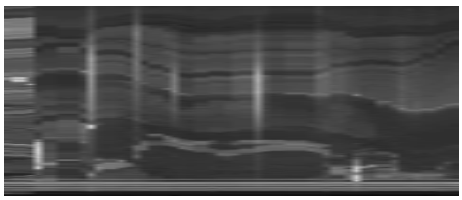
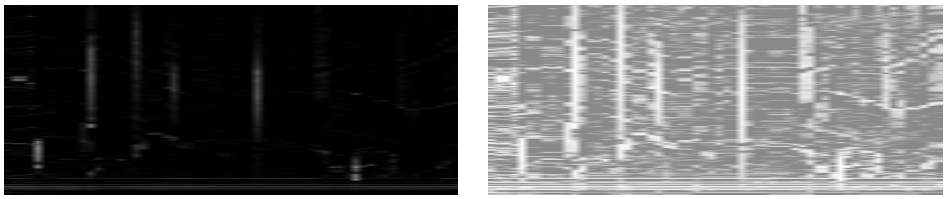


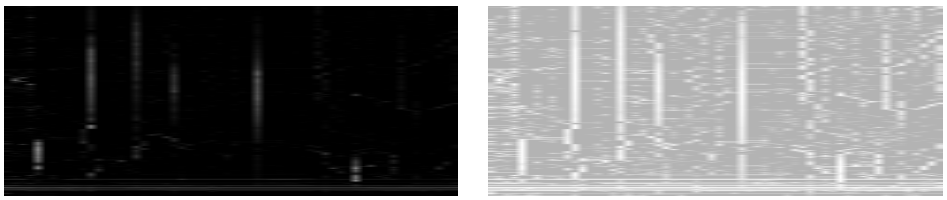
Figure 8.4: Visual rhythm by principal diagonal sub-sampling computed for video "0600.mpg".



(a) (b)

Figure 8.5: White top-hat filtering: (a) result of the white top-hat and (b) result of the histogram equalization of (a).

Thinning. Apply a $1\mathbb{D}$ thinning to each horizontal line to identify the center of each event. In Fig. 8.6, we illustrate the result of the thinning and the result of the histogram equalization of the thinning to facilitate its visualization.



(a) (b)

Figure 8.6: Thinning: (a) result of the thinning and (b) result of the histogram equalization of (a).

Detection of local maximum points. Considering that we have a thinned image, the computation of the maximum points is very trivial. We need to detect these points

to verify if they are vertically aligned. In Fig. 8.7, we illustrate the result of this step in which the maximum points are detected.

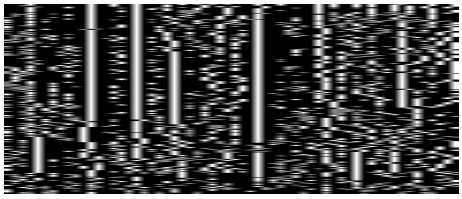


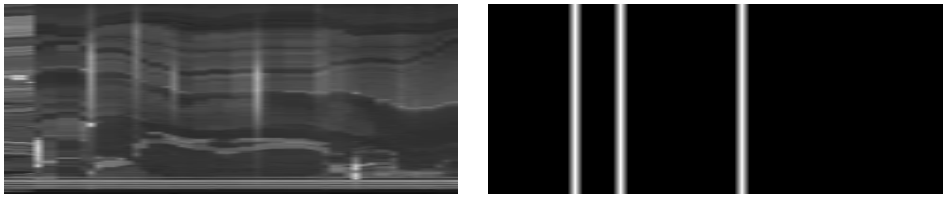
Figure 8.7: Detection of maximum points

Maxima image filtering. Apply an opening by reconstruction with vertical structuring element of size $\lambda = 7$, defined empirically. If the maximum points are not aligned, then they do not represent a flash, and consequently they are eliminated. In Fig. 8.8, we illustrate the result of the maximum points filtering.

Figure 8.8: Maxima image filtering.

Detection of flashes. Depending on the number of maximum points that are present in the maxima image filtering, a flash must be identified. For that, we must calculate the number of maximum points in each vertical line, followed by application of thresholding. In Fig. 8.9, we illustrate the flashes detected, that are represented by white vertical bars.

We can observe that this method is very similar to the proposed method for cut detection (Chapter 7). The difference here is the substitution of the morphological filter and horizontal gradient by the white top-hat by reconstruction. As the method for cut detection, this methodology detects the center of the regions in matter, in this case, regions with peak luminosity. Thus, we can have false detection in regions of high luminosity change that do not represent a flash. Usually, this method produces good results when the flash appears in the middle of the shot. The problem of this method is its noise sensitivity, and due to features of the used operators, we can not apply any filtering.



(a) (b)

Figure 8.9: Detection of flashes in which the flashes are represented by white vertical column bars: (a) original image with 4 flashes and (b) result of the flash detection.

8.3 Based on max tree filtering

Usually, the frames affected by a flash are visually similar to their neighbors but with a higher luminosity. Here, the analysis of flash presence can be given by the computation of some statistical measures like mean or median, because the frames affected by a flash present higher mean and median values in regarding to their neighbors. From the computation of these statistical measures for all frames of the video, we can create a $1\mathbb{D}$ image from the visual rhythm, or preferably from the original video data. From this $1\mathbb{D}$ image, we need to find the "peaks" which the "height" is greater than a certain value h, and a "basis area" less than or equal to a value S that corresponds to the duration of the flash. Fig. 8.1 illustrates the peaks which must be identified jointly with their features. In Fig. 8.10 is illustrated a block diagram for this flash detection method. Next, we describe the main steps of this method.

Visual rhythm computation. The visual rhythm by sub-sampling is computed from the principal diagonal of each frame. In Fig. 8.11, we illustrate the visual rhythm by principal diagonal sub-sampling computed between the frames 740 and 800 from the video "0600.mpg".

Average. For each column of the visual rhythm, we compute its average value producing a $1\mathbb{D}$ signal. This computation is very important because a frame that is affected by a flash has a higher average value than its neighbors. In Fig. 8.12, we illustrate the average computation and the and result of the histogram equalization of the average image to

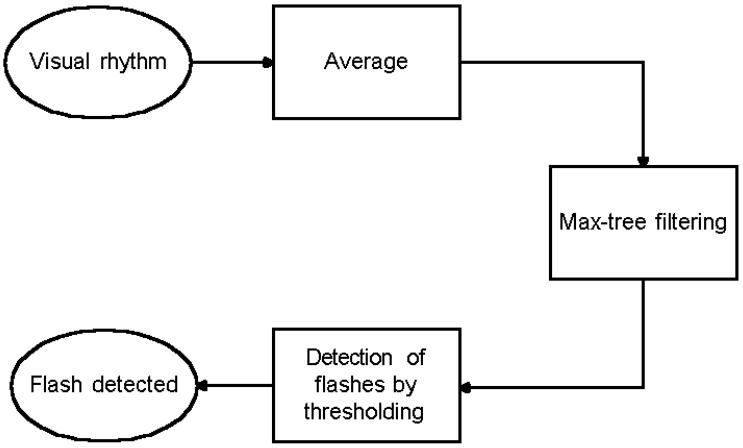


Figure 8.10: Flash detection block diagram using max-tree filtering.

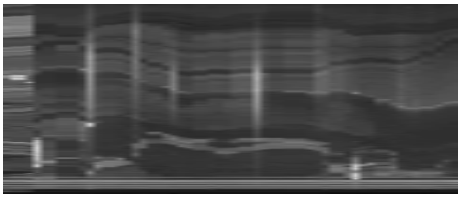


Figure 8.11: Visual rhythm by principal diagonal sub-sampling computed from video "0600.mpg".

facilitate its visualization.

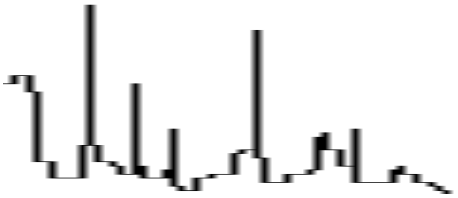


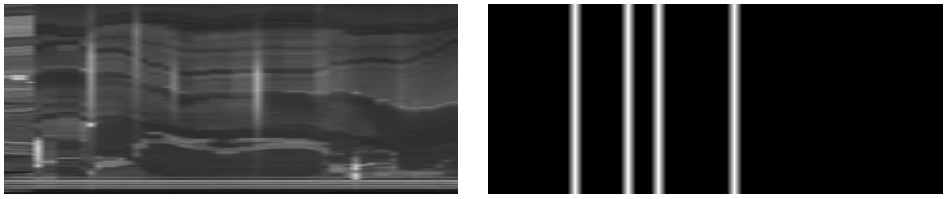
Figure 8.12: Average computation: result of the average of the elements for each column.

Max-tree filtering The notions of peak, height and basis area can be precisely defined thanks to a data structure called *max-tree* [62] (refer to this paper for more details). The parameter h influences the sensitivity of the method and has a role similar to the threshold in Chapter 7. In Fig. 8.13, we illustrate the result of the max-tree filtering.

Detection of flashes In this work, we consider that the maximum flash duration is 5 frames, i.e. $S = 5$. Thus, for each flash candidate we need to verify if the size is smaller

Figure 8.13: Result of the max-tree filtering.

than S. In Fig. 8.14, we illustrate the flashes detected, that are represented by white vertical bars.



(a) (b)

Figure 8.14: Detection of flashes in which the flashes are represented by white vertical column bars: (a) original image with 4 flashes and (b) result of the flash detection.

Differently from the first method, this one does not identify the center, but the problem is quite the same, mainly when there is high luminosity change that does not represent a true flash. Regarding other methods, this one identifies very well all types of flashes thanks to the data structure used to model the peak values.

8.4 Experiments

In these experiments, we apply the methods described in Sec. 8.2 and in Sec. 8.3 to the Corpus 2 illustrated in Table 8.1. In Fig. 8.15, we illustrate some experimental results. The quality measures of the filtering of the max tree and the top-hat filtering are outlined in Table 8.2. In Table 8.3, we illustrate the basic quality measures when we consider the values associated with the gamma measure.

	Videos	Cuts	Fades	Flashes	Frames	Frames/event (mean)
Corpus 2 (Flash)	10	-	-	23	8392	3

Table 8.1: Features of the videos which were selected for the experiments.

	μ	E_m	R_f	γ
Top-hat	0.05	**0.61**	0.26	0.56
Max tree	**0.11**	0.67	**0.43**	**0.69**

Table 8.2: Quality measures for flash detection: robustness $\mu(0.40, 0.30)$, missless $E_m(0.05)$, falseless $R_f(0.01)$ and gamma measure γ.

8.5 Conclusions and discussions

In this chapter, we proposed two methods for flash detection, in which the extraction of peaks by max tree analysis allows a detection of flashes in all positions of the shot. We computed the quality measures for flash detection, and according to these measures, the method based on max tree analysis generally presents the best results.

Exp	Type	Flash	Detected	False	Recall	Precision	Error	Threshold
Top-hat	Commercial	23	21	10	91%	68%	43%	35%
Max-tree	Commercial	23	20	3	87%	87%	13%	18

Table 8.3: Basic quality measures related to the gamma measure.

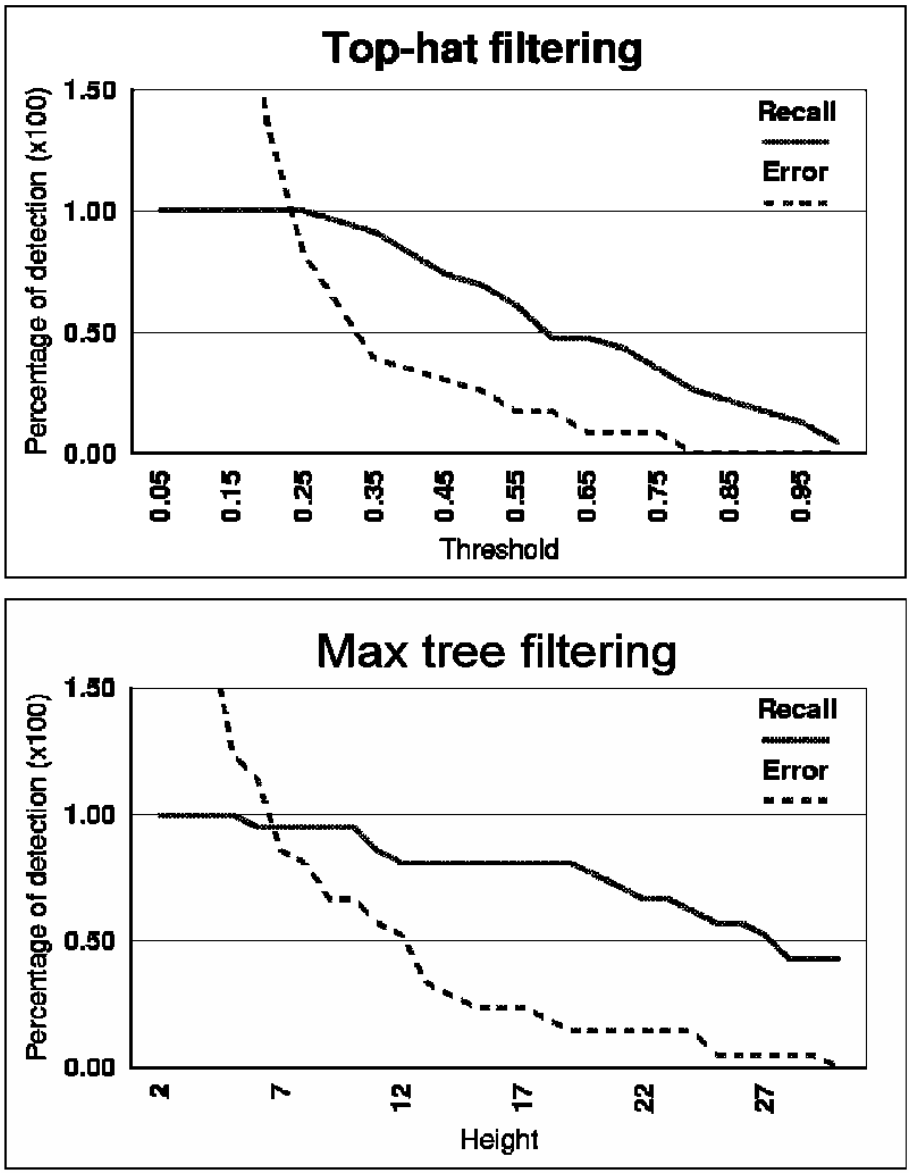


Figure 8.15: Experimental results for flash detection.

Chapter 9

Gradual transition detection

Unlike cut transitions, new frames are inserted between two consecutive shots when we have a gradual transition. Thanks to the correspondence between the gradual video transition and the vertical gradual transition in the visual rhythm, we can transform the problem of video segmentation into a problem of image segmentation. In this way, two approaches may be considered: the multi-scale approach, that considers a transition as the union of many transitions in different scales; and the sharpening approach, that eliminates the gradual transitions according to transformations of transition points. In this chapter, we propose two methods to identify video transitions considering the above approaches.

9.1 Introduction

When there is a gradual transition (as illustrated in Fig. 9.1) between two shots, new frames are created from these shots [39]. Usually, there are specific methods for each kind of transition. Methods for detecting fade and dissolve can be found in [77, 26, 50, 33, 79]. Most of these methods consider dissimilarity measures to characterize the video transition.

Here two different approaches are considered: the multi-scale approach and the sharpening approach. The former, described in Sec. 9.2, is a new approach to detect both cuts and gradual transitions. The idea of this approach is to identify fuzzy and abrupt transitions vertically aligned, in the visual rhythm representation, through multi-scale gradient operators. In this work, we consider three different morphological multi-scale gradient op-

(a) Cut.

(b) Fade-out.

(c) Dissolve.

Figure 9.1: Example of cut and gradual transitions.

erators: the Soille's multi-scale gradient and two other proposed variants of this method based on ultimate erosion and thinning concepts. And also, we use the Canny's filter [13] to compute the gradient values.

The second approach, described in Sec. 9.3, is based on the transformation of gradual transition into sharp transition considering an enlargement of the flat zones, that represent shots.

9.2 Method based on multi-scale gradient

The possibility to detect smooth variations between regions by considering thick gradient operators provides an important tool to identify gradual video transitions. Nevertheless, the result of these operators yields thick edges as result and, also, false edges (merge of the gradual boundaries) that can be produced if the distance between two transitions is smaller than the width of the used structuring element. These problems can be avoided

if we consider the multi-scale gradient proposed by Soille in [68] (described in Sec. 6.1.3). Here, we consider this gradient and we propose two variants of this operator based on the morphological ultimate erosion and thinning transformations. All operators are used to identify video gradual transitions.

9.2.1 Transition detection

The possibility to identify different kinds of transitions, like cuts and dissolves, constitutes an important and interesting aspect of our approach. This is possible mainly due to the use of the morphological multi-scale analysis. In this section, we present our algorithm for video transition identification, followed by a description and analysis of the realized experiments.

9.2.1.1 Algorithm

In Sec. 7 (see also [36]), we proposed an approach for cut transition identification. The method can be subdivided in two main modules: the gradient module and the spatio-temporal module. The former computes the presence of a gradient and the second one verifies if the gradient values are vertically aligned. Here, we propose to use a very similar approach, where the gradient module is replaced by a multi-scale gradient module and the spatio-temporal module is internally modified to adequate the possibility of identifying thick transitions. In Fig. 9.2 we illustrate a block diagram of our proposed method to identify both cut and gradual transitions. Next, we describe each module separately.

Let K be the maximum size of the transition to be detected. This value is related to the size n of the SE, and according to the SE family used in this work, $K = 2n + 1$. The multi-scale gradient module allows the identification of abrupt and fuzzy regions. As stated before, we consider three different types of gradient operators:

- Soille's gradient: this operator produces one-pixel-thin edges, but due to the fixed size of the SE used in the erosion operation, some gradient regions are not detected.
- Ultimate erosion: this operator preserves all gradient regions according to the properties of this erosion, and also one-pixel-thin or two-pixel-thin edges are produced in each level.

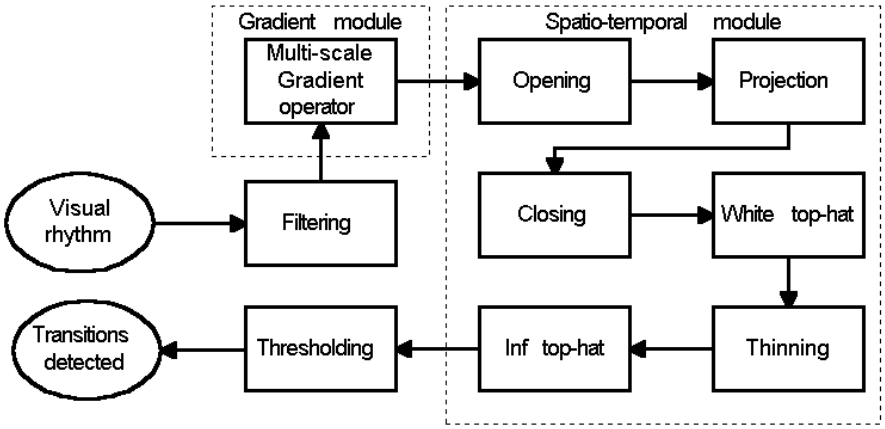


Figure 9.2: Block diagrams for detection of transitions considering multi-scale gradient.

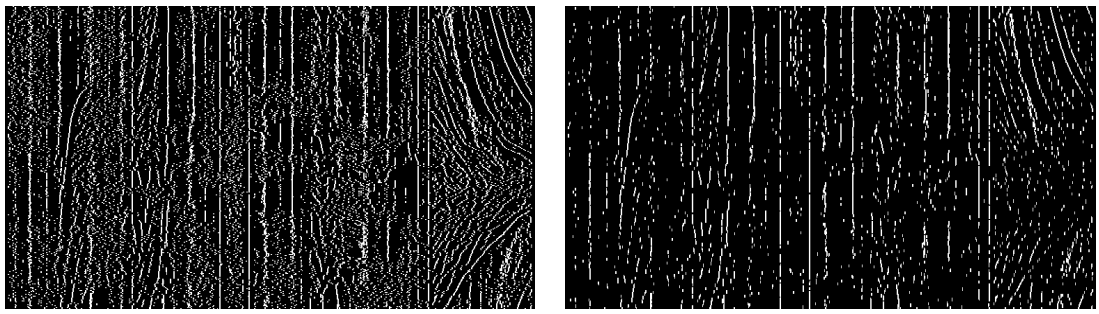
- Thinning: this operator preserves all gradient regions, as well as the topology (in the sense of [5]) of the result obtained from the thick gradient, i.e., the maximum regional is transformed into one-pixel-size region, and non-maximal regions are transformed into flat zones.

These operators are computed at a certain scale n, and to combine the results of various scales, a supremum operation is made. This operation may produce thick edges for all gradient operators. Here, we are interested in the presence of the gradient, without taking into account its values. Thus a thresholding operator is applied to the result of the gradient operator. In Fig. 9.9, we illustrate examples of thresholded multi-scale gradient obtained from the image presented in Fig. 6.3(a) in which we compute the Soille's gradient at scale $n = 7$ (Fig. 9.9(a)), the multi-scale gradient based on ultimate erosion at scale $n = 7$ (Fig. 9.9(b)), and finally, the supremum of the multi-scale gradient based on thinning at scales in the range $n = [1, 7]$ (Fig. 9.9(c)).

The spatio-temporal module executes the identification of vertically aligned transitions. Next, we describe each step of this module according to the block diagram described in Fig. 9.2(b). Here, the input of the spatio-temporal module is a binary image corresponding to the output of the thresholded gradient:

Opening. This operation is used to eliminate small spatial components, i.e., we ignore the time information. So, we consider a vertical SE with size (radius) of 2. In Fig. 9.3,

we illustrate the result of (a) thresholded Soille's gradient and the (b)result of opening operation.



(a) (b)

Figure 9.3: Opening operation: (a) thresholded Soille's gradient in which the SE is in the range $[1, 4]$ and (b) result of the opening operation using a vertical SE with size equals 2.

Projection. This operation is used to compute the number of non-zero gradient values in each column. A $1\mathbb{D}$ image is produced by this step. In Fig. 9.4, we illustrate the number of non-zero gradient values in each column computed from the result of the opening operation.

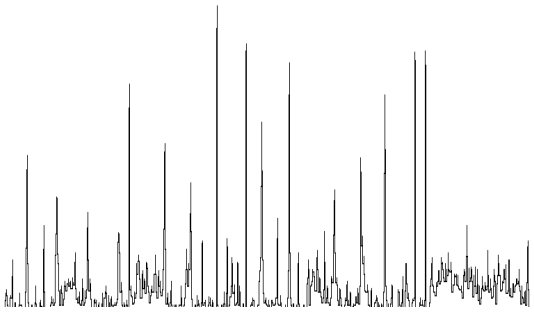


Figure 9.4: Number of non-zero gradient values.

Closing. The purpose of this operation is to eliminate small "holes" on the projected image. Empirically, we verified that a good size of SE for the closing is 3, which is associated with the size of the smallest shot ($k = 2 \times 3 + 1 = 7$). In Fig. 9.5, we illustrate the result of the closing operation.

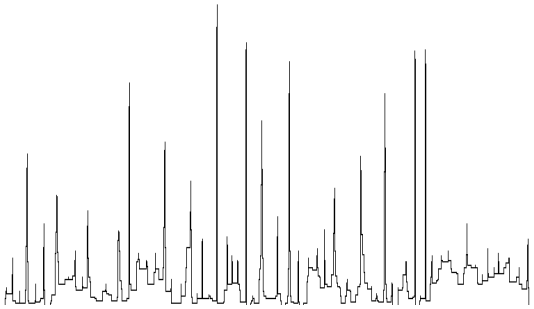


Figure 9.5: Closing operation using a SE of size 3.

White top-hat. The purpose of this operation is to consider only the "dome" that is related to a transition size n . Thus, we compute WTH_n. In Fig. 9.6, we illustrate the result of the white top-hat operation.

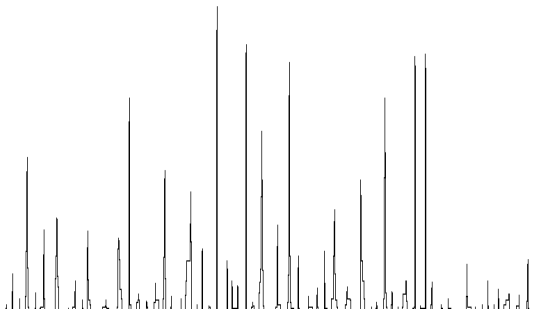


Figure 9.6: White top operation using SE of size 5.

Thinning. The purpose of this operation is to find the center of transitions represented by the regional peaks on the projection image. In Fig. 9.7, we illustrate the result of the thinning.

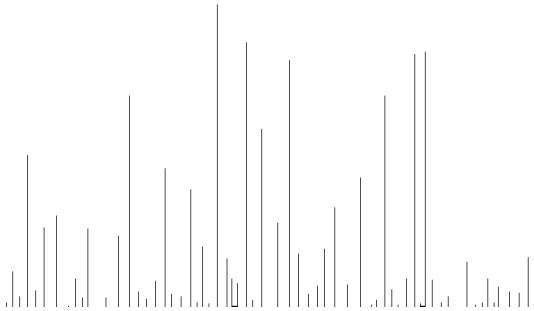


Figure 9.7: Thinning operation.

Inf top-hat. This operation computes the peak values with respect to its neighborhood. As the result of the thinning operation is an image with one-pixel-thin edges, an inf top-hat of level 1 is sufficient to calculate the peak value. This operation constitutes a type of filtering in which the plateau values are not considered. This filtering can eliminate regions that are not vertically aligned but vary according to time.

Detection of transitions. Finally, to find the transitions, we execute a thresholding on the result of the thinning to identify all video transitions. In Fig. 9.8, we illustrate the result of the thresholding ($threshold = 59$). As we can observe in Fig. 9.8, 3 transitions are not detected. This is because we tried to identify, in this example, the transitions of size smaller than 9 frames. Some transitions with size greater than 9 may be detected depending on their configuration, but in general, they are not identified.

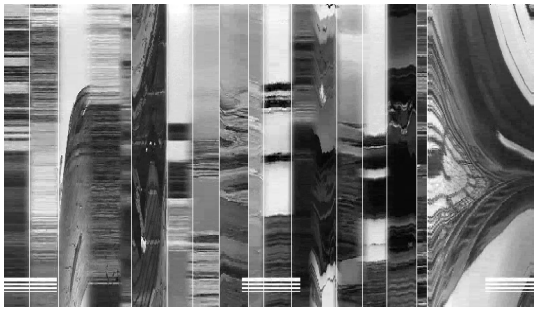


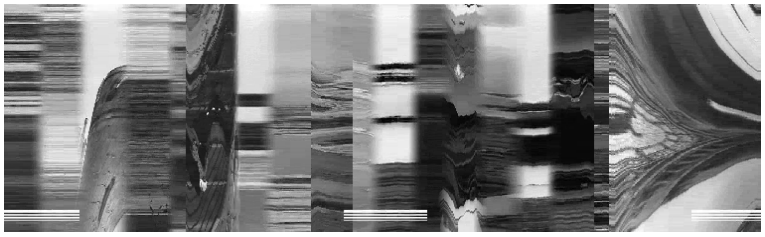
Figure 9.8: Detection of some types of transitions, like cuts, fades and dissolves.

In Fig. 9.9(b), Fig. 9.9(d) and Fig. 9.9(f) we illustrate the $1\mathbb{D}$ images produced by the projection operation, computed by Fig. 9.9(a), Fig. 9.9(c) and Fig. 9.9(e), respectively.

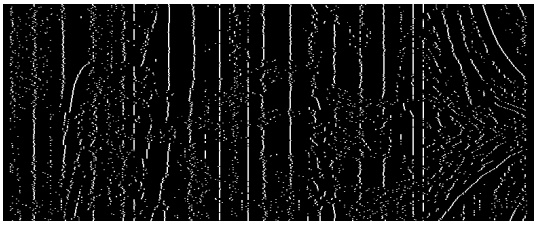
9.2.2 Experiments

Here, we consider only 29 videos that were chosen according to the presence of zoom, tilt, flash, dissolve, fade and cut. In Table 9.1, we outline some features of our video corpora.

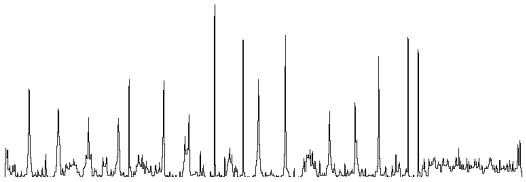
To compare the experiments, we use the quality measures defined in Section 3.3. The experiments done in this work consider the multi-scale gradient by Soille, and its variants based on the ultimate erosion and on the thinning transformation. Also, we make an experiment using Canny's filter [13] to compute the multi-scale gradient instead of the morphological gradient.



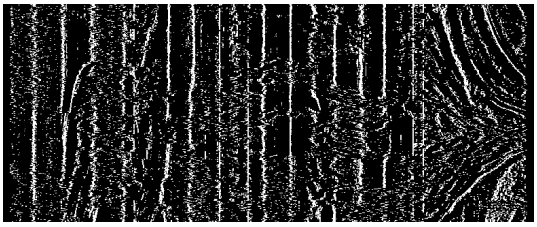
(a) Original image.



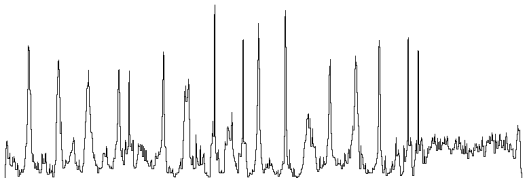
(b) Soille's gradient.



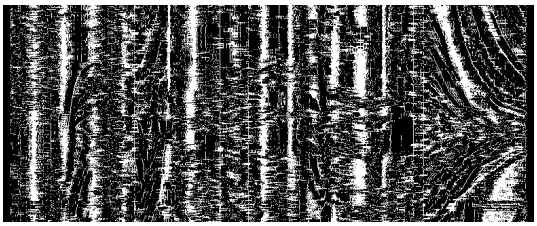
(c) Line profile of the projected image (a).



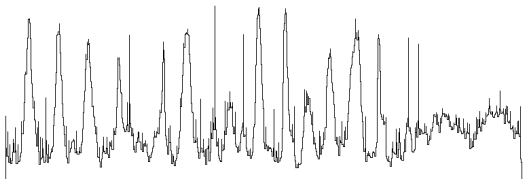
(d) Based on ultimate erosion.



(e) Line profile of the projected image (c).



(f) Based on thinning.



(g) Line profile of the projected image (e).

Figure 9.9: Example of multi-scale gradient analysis for transition detection, where $n = 7$ for the Soille's multi-scale gradient and gradient based on ultimate erosion, and $n = [1, 7]$ for the gradient based on thinning: the gradient computation (a,c,e) and the line profile of the projected image (b,d,f).

	Number of events	Average duration
Cut	250	0
Gradual ($\leq$15 frames)	140	9.00
Gradual (>15 frames)	56	23.2

Table 9.1: Features of the video corpora.

The main feature of the morphological gradient operators presented here is that all transitions smaller than the parameter K are detected considering a SE of size n. In our experiments, the size parameter of SE is set to 7, and consequently we search for detection of transitions of size smaller than $K = 2n + 1 = 15$, this value corresponds to the average size of gradual transitions. According to the features of the multi-scale gradient based on thinning, we run experiments with the size parameters of SE in the range $n = [1, 7]$.

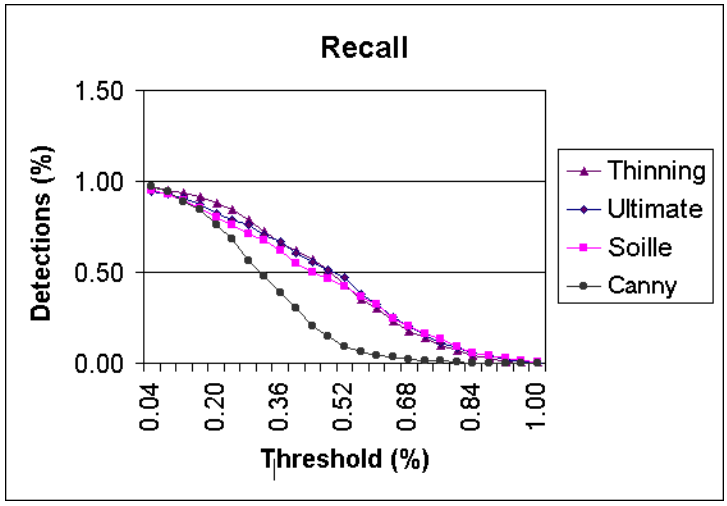
The parameter used of Canny's filter is $\alpha = 0.5$, that is associated with the size of the smallest shot. After the computation of this gradient for each row on the visual rhythm, we run the following sequence of steps: the opening operation, to reduce the noise; the thinning operation, to find the center of each dome; detection of the horizontal maxima; and finally, application of the spatio-temporal analysis as stated above.

In Fig. 9.10, we show the graphics that relate some quality measures to the threshold value. In Table 9.2, we outline the quality measure values obtained from these experiments. In Table 9.3, we illustrate the basic quality measures when we consider the values associated with the gamma measure.

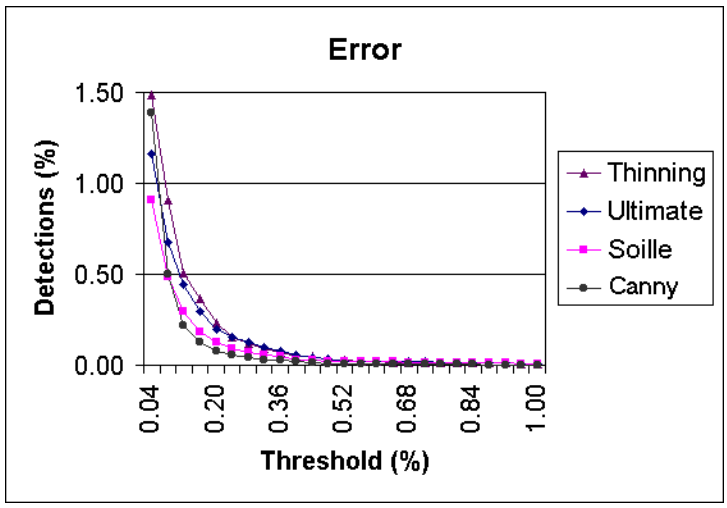
9.2.3 Analysis of results and parameters

As illustrated in Fig. 9.10, the results of our experiments are very similar. Unfortunately, the gradual transitions are difficult, due to this when the value of threshold increase the rate of correct detections decrease absurdly, this is explication of the form of the graphics illustrated in Fig. 9.10.

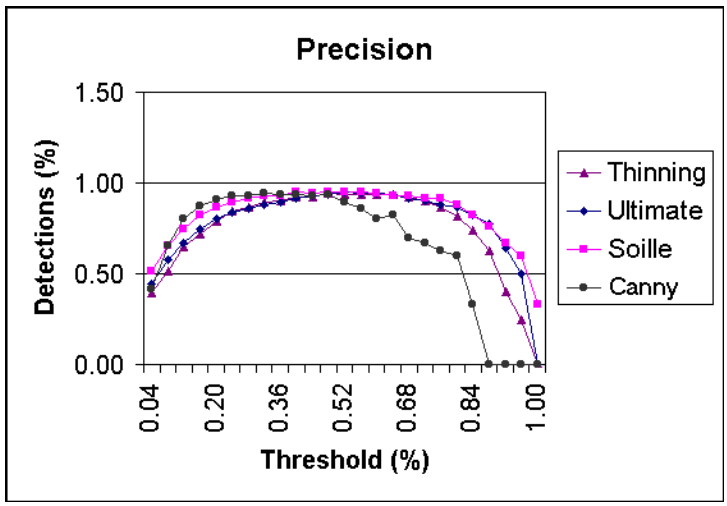
In general, the method based on thinning is more sensitive to low and high values of threshold due to the higher number of regions. On the other hand, the size of regions on the $1\mathbb{D}$ projected image is directly associated with the size of the transitions (as we can



(a)



(b)



(c)

Figure 9.10: Graphics of the quality measures: (a) recall; (b) error and (c) precision.

observe in Fig. 9.9). As we can note in Fig. 9.9, the method based on thinning produces more information and, consequently, a high number of correct detections is identified as well as false detections.

According to all quality measures, the multi-scale gradient proposed by Soille presents slightly better results, in regard to the studied variants, due to the fact that it produces less irrelevant information than the other gradients and, consequently, the reward function between the false detection and correct detection is higher. The problem of Canny's filter is its sensitivity to threshold values, which is, the number of correct detections decreasing very fast when the values of threshold increase, in comparison to the other gradient used here.

The misses in our experiments are mainly due to the quality of the videos and the presence of motion in the gradual transitions and, also, the presence of chained dissolves.

	μ	E_m	R_f	γ
Soille	**0.21**	0.40	**0.47**	0.70
Ultimate	**0.21**	0.46	**0.47**	0.67
Thinning	0.17	**0.36**	0.35	0.72
Canny	0.12	**0.36**	0.30	**0.74**

Table 9.2: Quality measures: robustness $\mu(0.3, 0.3)$, gamma measure γ, missless $E_m(0.08)$ and falseless $R_f(0.02)$

Finally, we can identify two types of parameters used in these experiments: the fixed parameters, which are set only once, associated with the size of the SE and used to filter the intermediate images; and the variable parameters, related to the size of transitions to be detected and the considered threshold. The tuning of these parameters is desirable because it allows the user to find a compromise between over-segmentation and under-segmentation.

The limitations of this approach is associated with the maximum size of the transition to be detected. This size is directly dependent on the size of the structuring element used. In order to reduce this problem we try to eliminate the gradual transitions instead of detecting them.

Exp	Type	Gradual	Detected	False	Recall	Precision	Error	Threshold
Soille	Commercial	393	316	49	80%	87%	12%	25%
Ultimate	Commercial	393	311	60	79%	84%	15%	30%
Thinning	Commercial	393	333	63	85%	84%	15%	30%
Canny	Commercial	393	331	49	84%	87%	12%	20%

Table 9.3: Basic quality measures related to the gamma measure.

9.3 Sharpening by flat zone enlargement

The existence of gradual transitions in the image yields a more difficult problem of edge detection. To deal with this problem, we can consider, for example, multi-scale [34, 68] and sharpen approaches. While the multi-scale approach considers a gradual region as edges of different sizes that can be identified at different scales, the sharpen approach tries to simplify the edges by eliminating (or reducing) the gradual regions. The problem with the first method concerns the definition of the maximum scale to consider because the transition identification is directly associated with this scale. In this work, we will consider the sharpen approach in the identification of gradual transitions of the image. More specifically, we will try to transform the gradual transitions into sharp transitions in a $1\mathbb{D}$ signal by using some operators which enlarge the corresponding neighboring regions.

Next, we give some basic concepts considered in this paper. Let g be a $1\mathbb{D}$ signal represented by a function of $\mathbb{N} \rightarrow \mathbb{N}$. We denote by $\mathrm{N}(p)$ the set of neighbors of a point p. In such a case, $\mathrm{N}(p) = \{p-1, p+1\}$ represents the right and the left neighbors of p.

Definition 9.1 (Flat zone, k-flat zone and k^+-flat zone) *A flat zone of g is a maximal set (in the sense of inclusion) of adjacent points with the same value. A k-flat zone is a flat zone with size equal to k. A k^+-flat zone is a flat zone with size greater than or equal to k.*

Definition 9.2 (Transition) *We denote by F the set of k^+-flat zones of g. A transition T between two k^+-flat zones, F_i and F_j, is the range $[p_0..p_{n-1}]$ such that $p_0 \in F_i$, $p_{n-1} \in F_j$, $\forall 0 < m < n-1,\ p_m \notin F_i \bigcup F_j$, $\forall l \neq i, j\ F_l \not\subset [p_0..p_{n-1}]$ and $\forall 0 \leq i < n-1,\ g(p_i) \leq g(p_{i+1})$ (or $g(p_i) \geq g(p_{i+1})$ - monotonic transition).*

In this work, the analysis of the transition regions is related to the identification and

elimination of the neighboring points of these transitions while the number of k^+-flat zones is preserved, here we set, empirically, $k = 7$. Next, we define two different kinds of transition points, namely, the constructible and destructible points (as illustrated in Fig. 9.11).

Let $\mathrm{D}(p, F)$ be the grayscale difference between p and the flat zone F.

Definition 9.3 (Constructible transition point) *We denote by T the transition between two k^+-flat zones, F_i and F_j. Let $p \in T$, $p-1$ and $p+1$, be a pixel of a $1\mathbb{D}$ signal, g, and its neighbors, respectively. A point p is a constructible transition point iff $g(p) \geq min(g(p-1),\ g(p+1))$, $g(p) \leq max(g(p-1),\ g(p+1))$ and $D(p, F^-) > D(p, F^+)$, where F^- and F^+ denote the flat zone nearest to p with lower and higher grayscales, respectively.*

Definition 9.4 (Destructible transition point) *We denote by T the transition between two k-flat zones, F_i and F_j. Let $p \in T$, $p-1$ and $p+1$, be a pixel of a $1\mathbb{D}$ signal, g, and its neighbors, respectively. A point p is a destructible transition point iff $g(p) \geq min(g(p-1),\ g(p+1))$, $g(p) \leq max(g(p-1),\ g(p+1))$ and $D(p, F^-) < D(p, F^+)$, where F^- and F^+ denote the flat zone nearest to p with lower and higher grayscales, respectively .*

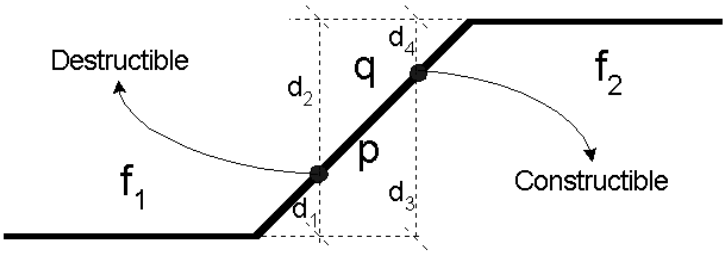


Figure 9.11: Transition points (constructible and destructible).

In Fig. 9.11, we illustrate an example of identification of constructible and destructible points. In such a case, p is a destructible point ($d_1 < d_2$), and q is a constructible point ($d_4 < d_3$). The aim here is to simplify the image without changing the number of its k^+-flat zones. In other words, we want to change only the grayscale values of transition points in the neighborhood of k^+-flat zones, without suppressing or creating new flat zones. The choice of the points to be evaluated is an important parameter of this operator which may affect the final result. Next, we describe the algorithm to eliminate the gradual transitions

of the image by extending the set of its flat zones. The algorithm takes into account the following aspects.

Flat zone identification. This first step identifies all the k^+-flat zones of the image represented by a set F. The application of a filtering method, e.g., a closing followed by an opening operation, may be considered to reduce small irrelevant flat zones of the original image.

Marker identification. The markers, M, correspond to the flat zones of size greater than or equal to k.

Candidate points (C). The candidates points, C, correspond to the neighboring points of the markers: $\mathrm{C} = \{q \mid \exists p \in \mathrm{M}, q \in \mathrm{N}(p) \text{ and } q \notin \mathrm{M}\}$.

Point removal. This operation is given by the following instruction: while $C \neq \emptyset$ do $C = C \backslash \{p\}$ and $C = C \cup q$ (point removal, where p is a constructible or destructible point, and q is the neighbor of p which was not yet modified). The grayscale value of p is modified according to the grayscale value of the neighboring flat zone.

In the above steps, an important aspect concerns the point removal since, depending on the removing order, different results can be obtained. For the purpose of controlling this removal, we use a hierarchical priority queue to maintain an equidistant relation between the removed points and their neighboring flat zones. Thus, the point to be removed presents the smallest priority in this queue, where the priority depends on the criterium used to insert new points in this data structure. Here, we consider gray-scale difference between a k^+-flat zone and its neighboring points as the insertion criterium.

Fig. 9.12 gives an example of flat zone enlargement (or sharpening). The next section illustrates the application of this sharpening operator in the detection of gradual transitions of video images.

9.3.1 Transition detection

The video segmentation problem in the presence of gradual transitions is very difficult, mainly, in the case of dissolves (a video dissolve is the gradual content change of a video

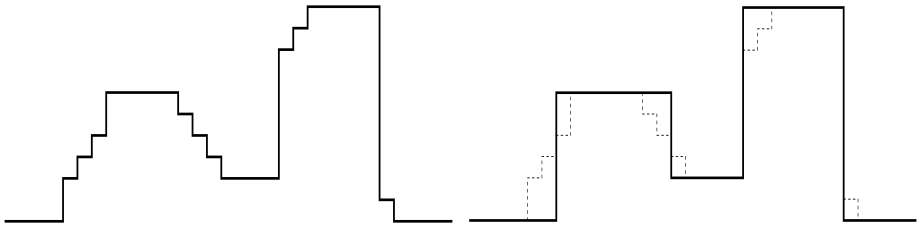


Figure 9.12: Example of enlargement of flat zones: an artificial example.

shot into another one). As described in [57], the gradual transitions are represented by vertically aligned gradual regions on VR. In Fig. 9.13(a), we illustrate a VR of a video that contains 3 cuts (vertically aligned transition), 1 wipe (oblique transition), 2 fades and 1 dissolve. In Fig. 9.13(b) we illustrate the result of our sharpening operator. Fig. 9.13(c) and Fig. 9.13(d) illustrate the line profile of Fig. 9.13(a) and Fig. 9.13(b) of the center horizontal lines, respectively. All lines of the VR present the same features, i.e., in the presence of a gradual video transition, an increasing or a decreasing of the grayscale value in regard to the temporal axis, in a specific time range, can be identified in all lines of the VR.

To detect the gradual transitions of a video we can simplify the VR by considering the sharpening operator described in Sec. 9.3. As we have seen before, this operator preserves the number of k^+-flat zones, and since a k^+-flat zone can be related to a shot, our method preserves the initial number of shots. To reduce noise, we apply an alternated morphological filter with a linear structuring element of size 3 . This value is closely related to the smallest size of a shot. Now, consider that the visual rhythm (VR) was modified according to the sharpening operator described in Sec. 9.3 giving a result called VR^e. The first idea to identify gradual transitions is to apply the sharpening operator to the visual rhythm, that will produce sharp transitions, followed by application of the method for cut detection (described in Chapter 7). Unfortunately, the result of the sharpening operation is not vertically aligned, this is why we do not directly use the method for cut detection. Thus, the basic idea of our method for gradual transition detection is to study the behavior of the points of the VR by taking into account the number of its modified points (points of the gradual transitions) in the sharpened visual rhythm VR^e. From the sharpened image VR^e, we can consider the following steps, summarized in Fig. 9.14, for gradual video transitions identification.

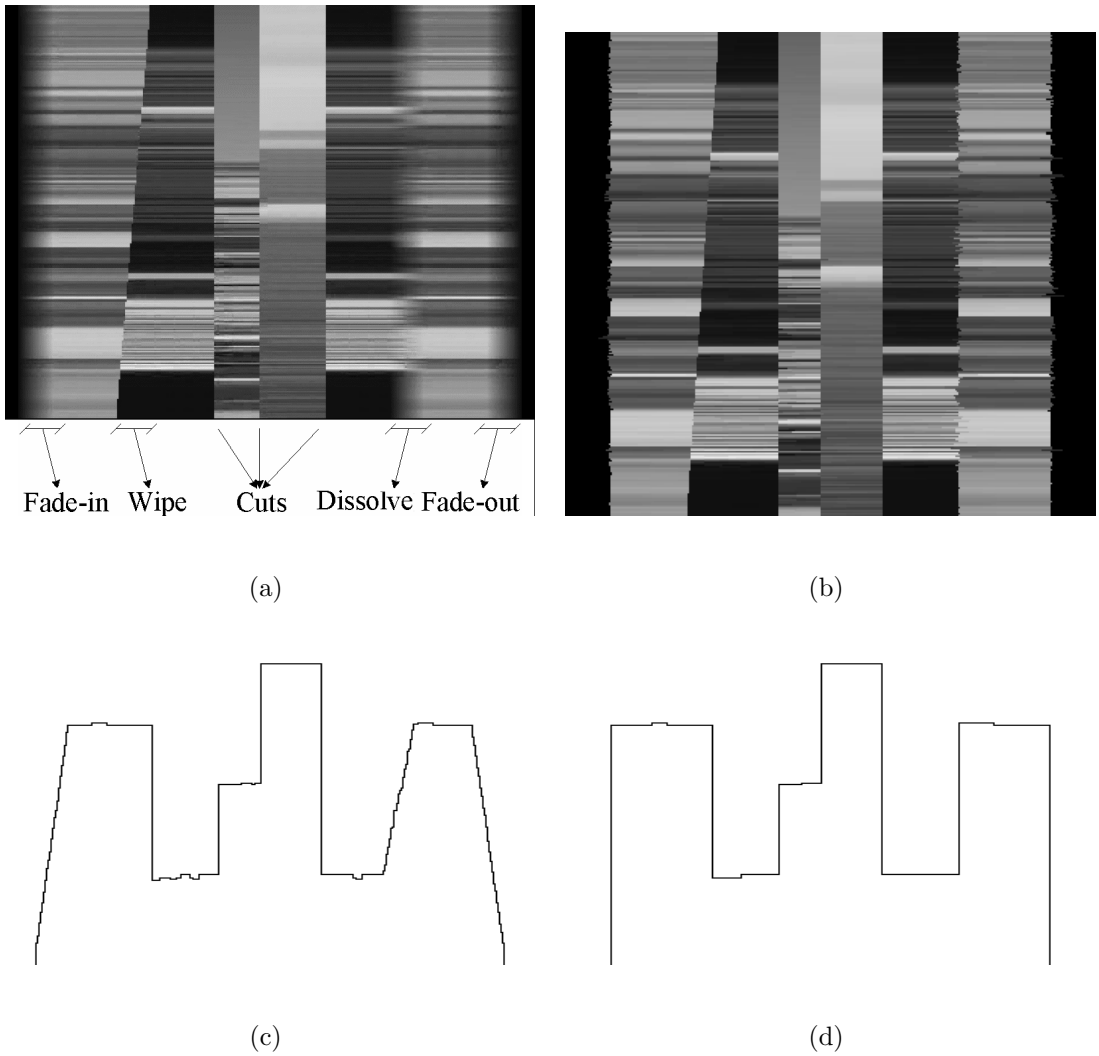


Figure 9.13: An example of visual rhythm with some events (a), and the image obtained by the sharpening process (b). In (c) and (d) is illustrated their respective line profile of the center horizontal line of the image.

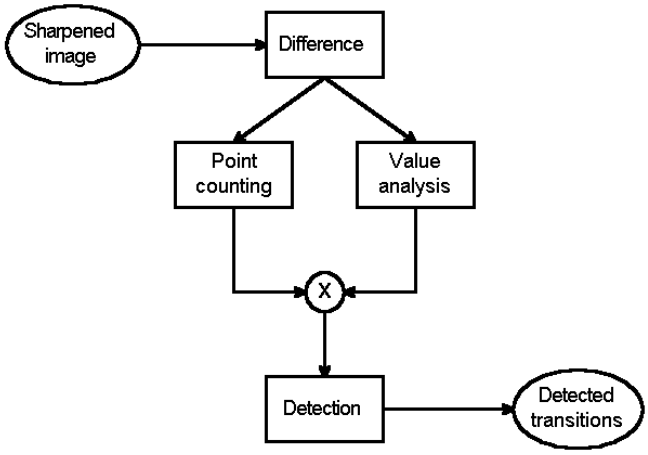


Figure 9.14: Main steps of the proposed gradual transition detection algorithm for video images.

Difference. It computes the difference between the original VR and the sharpened VR^e. Thus, a new image Dif is computed as follows

$$\mathrm{Dif}(x,y) = |\mathrm{VR}(x,y) - \mathrm{VR}^e(x,y)| \tag{9.1}$$

Point counting. It defines, from the above operation, the number of transition points modified by the sharpening process. Thus, a simple operation which counts of non-zero points in each column of the difference image Dif is considered. To reduce the noise and fast motion influence, we apply an opening operation with a vertical structuring element before the counting process given by

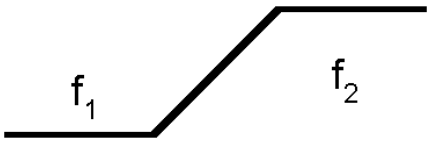
$$\mathrm{M}^p(p) = \sum_{j=0}^{H_{VR}} \begin{cases} 1, & \text{if } \mathrm{Dif}(p,j) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9.2}$$

where H_{VR} mean the height of the VR.

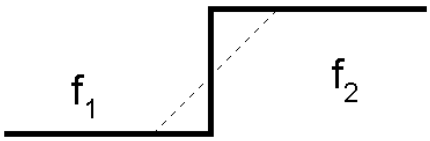
Value analysis. It computes the grayscale mean of the transition points modified by the sharpening process. As illustrated in Fig. 9.15, gradual transitions are represented by single domes (Fig. 9.15(c)) in each horizontal line of the image Dif, the center of the transitions corresponding to their regional maximum. Usually, the first and the last frames of gradual transitions corresponds to the smallest values of these domes. We may observe that the values between the first and the center frames increase, while they decrease between its center and last frames, this is because the transition is monotonic. Before

analyzing the domes of the image Dif, we can compute the mean value of the points for each column of this and analyze the behavior of each frame with respect to its neighbors (on VR, a column corresponds to a frame). In such a case, we create a $1\mathbb{D}$ signal, called M^v, containing the result of this operation, as follows

$$\mathrm{M}^v(p) = \frac{\sum_{y=0}^{\mathrm{H}_{VR}-1}(\mathrm{Dif}(p,y))}{\mathrm{H}_{VR}} \tag{9.3}$$



(a) Original



(b) Sharpened image

(c) Difference image

Figure 9.15: Example of enlargement of flat zones.

To identify the domes configuration, as show in Fig. 9.15(c), we can decompose our $1\mathbb{D}$ signal M^v in morphological residues (as described in [47]) and to study their decomposition behavior. Here, we illustrate a simple case of this decomposition analysis in which we compute the morphological residues between two different levels Inf and Sup. Based on this information, we can verify the number of residual levels containing a point p (that represents a frame f), then

$$\mathrm{M}_{Inf}^{Sup}(p) = \sum_{i=Inf}^{Sup} \begin{cases} 1, & \text{if } \mathrm{R}_i(\mathrm{M}^v(p)) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9.4}$$

where R_i means the residue at level i. We consider that a point p corresponds to a candidate frame for gradual transition if $\mathrm{M}_{Inf}^{Sup}(p)$ is greater than or equal to a threshold,

where the point p considered corresponds to the regional maximum, so

$$\mathrm{C}_{Inf}^{Sup}(p) = \begin{cases} 1, & \text{if } \mathrm{M}_{Inf}^{Sup}(p) > l_1 \\ 0, & \text{otherwise} \end{cases} \tag{9.5}$$

where the values of Inf, Sup and l_1 were empirically defined as 3, 15 and 3, respectively. Note that the configuration of each dome is very important to identify a gradual transition, but it does not represent a sufficient criterium. Here, we also need to verify if the number of points of each candidate frame that was modified by the sharpening process is greater than a specified threshold.

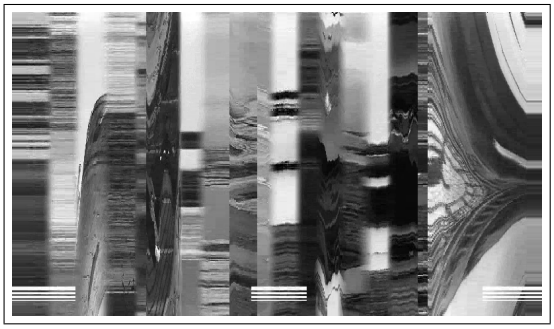
Detection operation. It combines the information obtained from the point counting and the value analysis steps defined previously (a certain number of points with a given grayscale value represents the gradual transitions of the video). By considering that a dome in M^v corresponds to a gradual transition if this dome presents a special configuration (as illustrated in Fig. 9.15(c)), and that the number of modified points is high, we can combine above mentioned steps as follows

$$\mathrm{M}_p^v(p) = \begin{cases} \mathrm{M}^p(p), & \text{if } \mathrm{C}_{Inf}^{Sup}(p) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9.6}$$

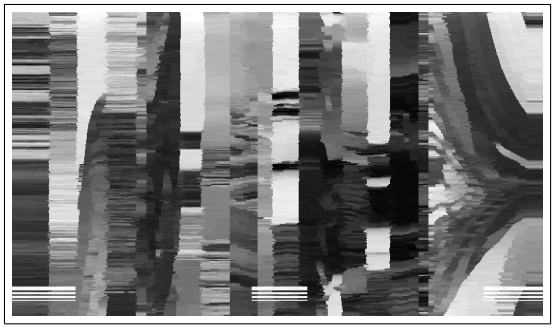
This equation takes into account the candidate frames and their number of points modified by the sharpening process. At this step, we find the gradual transitions using the simple thresholding operation

$$T(p) = \begin{cases} 1, & \text{if } \mathrm{M}_p^v(p) > l_2 \\ 0, & \text{otherwise} \end{cases} \tag{9.7}$$

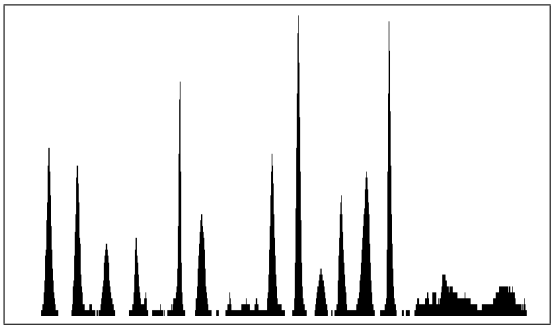
where l_2 is a threshold value. In Fig. 9.16 we illustrate our method for detecting gradual transitions. In this example, we apply the method for each horizontal line. Firstly, we compute the sharpened image as described in Sec. 9.3 in which the gradual transitions are transformed into sharp transitions (Fig. 9.16(middle)). In Fig. 9.16(bottom) we illustrate the result when 25% of the maximal value of M_p^v is considered as a threshold ($l_2 = 25\%$). To evaluate our method, we realized some experiments in next section.



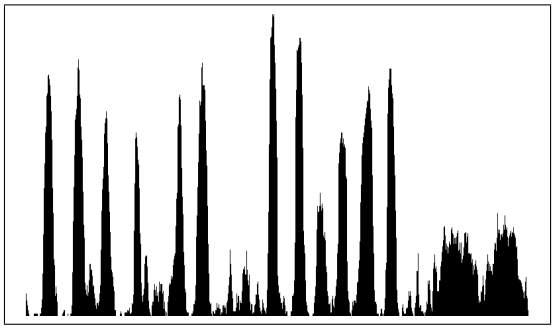
(a) Original image.



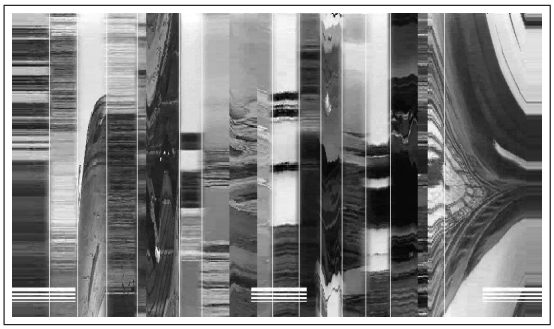
(b) Sharpened image.



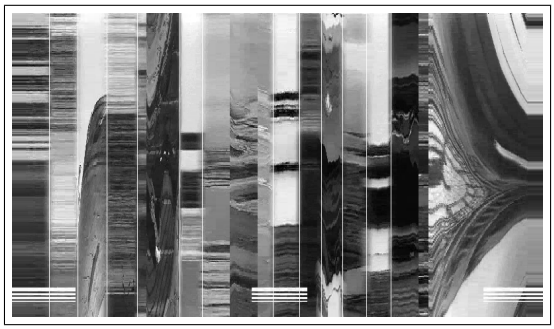
(c) M^v.



(d) Number of modified points.



(e) M^v thresholded.



(f) Result of our proposed method.

Figure 9.16: Gradual transition detection.

9.3.2 Experimental analysis

In this section, we show the experimental results for gradual transition detection. The choice of the sequences was associated with the presence of the different characteristics, such as cut, dissolve, wipe, flashes, zoom-in, zoom-out, pan, tilt, object motion, camera motion, computer effects, and also due to their quality. In Table 9.4, we show some features of the chosen videos. To compare the different methods, we use some quality measures according to Sec. 3 ([36]). Here, we are interested in gradual transitions, so $\#Events$ represents the number of gradual transitions that satisfy our hypotheses: size greater than 11 frames. Following, we describe the experiments realized.

Experiment 1. In this experiment, we consider only the grayscale values of the difference image to identify the gradual transitions. So, a transition T is detected if the $\mathrm{M}^v(T)$ is greater than or equal to a threshold. This value was empirically defined and corresponds to 2% of the maximal possible value (255). Also, T represents a regional maximum because this point is probably the center of the transition.

Experiment 2. In this experiment, we consider only the number of modified points by the sharpening process. If $\mathrm{M}^p(T)$ is greater than or equal to a threshold, then T corresponds to a transition. The point analysis corresponds to the regional maximum of the M^v. The threshold value corresponds to 25% of the VR height.

Experiment 3. This experiment corresponds to our proposed method in which the grayscale values and the modified points are considered.

Experiment 4. In this experiment, we use the twin-comparison approach (that was discussed in Section 3.1.1) to identify gradual transitions based on histogram difference filtered. We observed in many sequences that some transition frames were duplicated, and due to this a very small dissimilarity measure is obtained. So, a closing of the $1\mathbb{D}$ histogram difference value image was applied. In twin-comparison approach, two thresholds are considered, T_s and T_b that represent the candidate frames and transition frame, respectively. Here we consider $T_s = 0.1$ and $T_b = 0.5$. We need to say that all cuts are not considered, so a transition is detected if its frames are candidates.

Our method detects only gradual transitions of size greater than 11 frames. This value can be easily decreased but in this case the number of false detections may increase. The

Type	Videos	Gradual
Commercial	28	77

Table 9.4: Chosen video features for the experiments

Exp	Type	Gradual	Detected	False	Recall	Precision	Error	Threshold
1	Commercial	77	75	46	97.5%	62%	60%	2%
2	Commercial	77	75	52	97.5%	59%	67%	25%
3	Commercial	77	72	10	93.5%	88%	13%	25%
4	Commercial	77	57	53	74%	52%	68%	0.5 and 0.1

Table 9.5: Results of our experiments

choice of this size was empirically defined.

9.3.3 Analysis of the results

According to Table 9.5, we can observe that our proposed method is better when compared to other experimented methods. When we consider only grayscale values, the transitions are very well identified due to their special configuration, but this method (Experiment 1) is more sensitive to difference between two consecutive shots. When we consider only the modified points, transition frames can be confused with special events and fast motions. So, the method (Experiment 2) is more sensitive to noise and fast motion. This is why we consider these two information to identify gradual transitions. Experiment 3 shows that the quality measure of our method is better when compared to others. If we compare our method to twin-comparison approach (Experiment 4), we observe that our results are also better. The main positive aspect of our method is associated with the possibility of identifying transitions between two consecutive shots in which their textures are quite similar. This is not possible with other methods. Some false detections of our method are due to the identification of transitions with duration smaller than 10 frames. This may be possible due to the presence of noise in the neighborhood.

9.4 Conclusions and discussions

In this chapter, we proposed two different methods to identify gradual video transitions. Each method is based on a different approach. The first one is based on a multi-scale approach and the second one considers a sharpening approach.

With regard to the multi-scale approach, we proposed a new approach to identify cut and gradual transitions. Our approach is based on morphological multi-scale gradient operators, more specifically, the Soille's gradient and two other variants that are based on ultimate erosion and thinning. An interesting aspect we remark here is the inclusion relation between the multi-scale ultimate erosion gradient and Soille's gradient, where the former contains the second one. In a general way, the Soille's gradient present the best results, but the gradient based on thinning contains more information concerning, for example, the transition size. In this work, we are interested mainly in identifying the presence of cut and gradual transitions without taking account the size of these transitions. This information will be considered in a future work. From the above experiments of this approach, we observe that the use of the multi-scale gradient can represent an interesting approach to cope with the problems of gradual transitions. According to the features of the gradient based on thinning, a study about the relationship between the transition size and this thinning can be envisaged. Also, a study of the behavior of this method could be done considering its application directly to the video data.

Regarding the sharpening approach, we proposed a new method to transform gradual transitions into sharp ones in a $1\mathbb{D}$ image and its application to gradual video transition detection. The sharpening operator defined here is based on the classification of gradual transitions points as constructible or destructible points. This operator constitutes the first step for the detection of a very common video event known as dissolve. The main feature of our approach is that it does not depend on the transition size, i.e, any dissolve event with different transition time can be detected. Furthermore, the effective cost of our method is much better when compared to others which consider all video information. A drawback of this method concerns its sensitivity to motion. The motion sensitivity could be reduced by a step of motion compensation. On the visual rhythm, the motion compensation is facilitated thanks to the $1\mathbb{D}$ signal representation of this image. We can use, for example, the dynamic programming to cope with this problem of motion

compensation. Interesting extensions of this work concern the analysis of the efficiency of our method, when applied to a long video sequence.

Chapter 10

Specific fade detection

Usually, fades and dissolves can be detected by the same method thanks to their features. Sometimes, these transitions represent different sensations and for that, to differentiate them is very interesting. Nowadays, most of the methods consider the fade as a special case of the dissolve. Here, we propose a new method to detect only fades in which this transition is not considered as a special case of a dissolve.

10.1 Introduction

Here, we introduce the notion of visual rhythm by histogram to deal with the fade detection problem. Also, we propose a method for fade detection based on the analysis of $2\mathbb{D}$ images taking advantage of the fact that each fade is represented by an "inclined line" in this image that, informally, conveys a histogram representation of all frames.

As we described in Chapter 2, the fade process is characterized by a progressive darkening of a shot until the last frame becomes completely black, or inversely. A more general definition is given by [50] where the black frame is replaced by a monochrome frame.

From the fade definition, we can observe that the intermediate frames are directly dependent on the transformation function, $\alpha(t)$, and their color distribution is a combination of the color distribution of the extremal fade frames. To study the behavior of the color distribution we will use image histograms. Consider that a fade with n frames is represented by $(T_{f_0}, T_{f_1}, T_{f_2}, \cdots, T_{f_{n-1}})$ and by a sequence of the histograms computed from theses frames $(H_{T_{f_0}}, H_{T_{f_1}}, H_{T_{f_2}}, \cdots, H_{T_{f_{n-1}}})$. As the fade is related to darkening (lighten-

ing) of a frame series (as illustrated in Fig. 10.1(a)), we can verify that the histograms of the fade frames present an increasing (decreasing) number of different grayscales, so called histogram width.

Definition 10.1 (Histogram width) *Let H_{f_t} be a histogram with L bins. The histogram width, N_H, corresponds to the number of different grayscales (i), where $H_{f_t}(i)$ is greater than or equal to a threshold l, and can be defined by*

$$N_{Hf_t} = \#\{i | H_{f_t}(i) \geq l \text{ where } i \in [0, L-1]\} \tag{10.1}$$

The two histograms, $H(g) = (10, 5, 0, 0, 0, 0, 0, 0)$ and $H(f) = (0, 10, 0, 0, 0, 5, 0, 0)$, present the same histogram width ($N_H = 2$ for $l = 5$) but their spatial arrangement is different. In this case, only $H(g)$ can be associated with a fade due to its spatial arrangement, i.e., there is only one "connected component". If we consider only the quantitative information related to these distributions, we loose the spatial aspect of an image histogram.

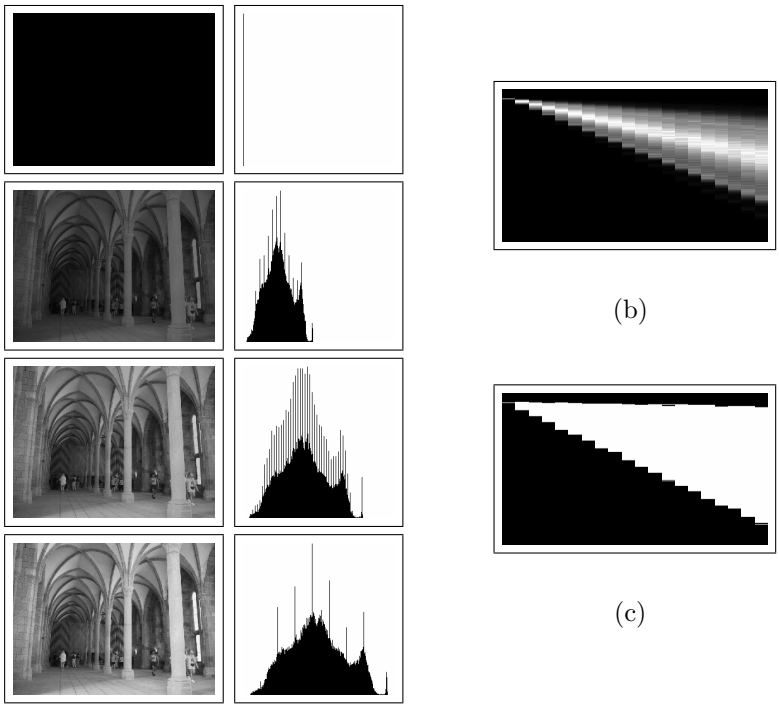
10.2 Video transformation

Taking advantage of the properties of an image histogram, such as global information, invariance to rotation and translation, we consider the video transformation, called visual rhythm by histogram (VRH), where the video is transformed into a $2\mathbb{D}$ image containing frame histogram information (as defined in Chapter 4).

In Fig. 10.2, we illustrate an example of visual rhythm by histogram where each value of the histogram is in the range $[0, 255]$. In this image, we can visually identify some patterns that are related to some transition types like cut, fade, wipe and dissolve (e.g., the fades are represented by inclined transition regions).

10.3 Analysis based on discrete line identification

A natural way for detecting the fades is to study the behavior of luminosity change [50, 26] by modeling the process of fade creation with mathematical equations. Unfortunately,



(a) (b) (c)

Figure 10.1: An example of fade in: ((a)-left) illustrates the frames 0, 14, 24 and 29 of a transition with 30 frames and their respective histograms ((a)-right); (b) visual rhythm by histogram of the fade transition; (c) result of the image segmentation (thresholding) of (b).

this process fails due to noise, when the extremal frames are not completely monochrome, and mainly, when we have short fades. From the fade definition, an interesting aspect to analyse concerns the behavior of the image histograms of the fade sequence. The evolution of these histograms can be studied based on the VRH representation in which the fade is characterized by inclined edges (as illustrated in Fig. 10.1).

We propose, in this chapter, a methodology for fade detection based on analysis of visual rhythm by histogram from inclined edge identification, this methodology is illustrated in Fig. 10.3. Next, we describe each step of our method separately.

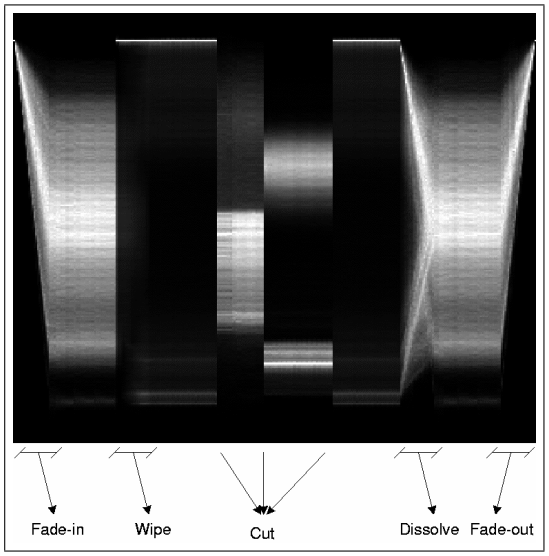


Figure 10.2: Visual rhythm by histogram.

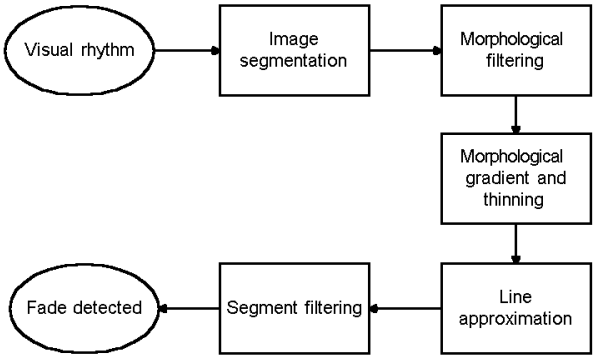


Figure 10.3: Block diagram for the fade detection method

Visual rhythm computation. The visual rhythm by histogram is computed from each frame. In Fig. 10.4, we illustrate an example of this visual rhythm.

Image segmentation. According to the fade definition, the darkening (lightening) process of its frames can be interpreted as a progressive shrinking (expansion) of the histogram width in which this value must be related to one connected component. Considering that the normalization process to create the VRH produces a filtering effect on the histogram values, we can use a thresholding operation ($threshold = 1$) to identify the regions related to the histogram information.

Due to the space shrinking (expansion) and the time coherence of the fade frames, a

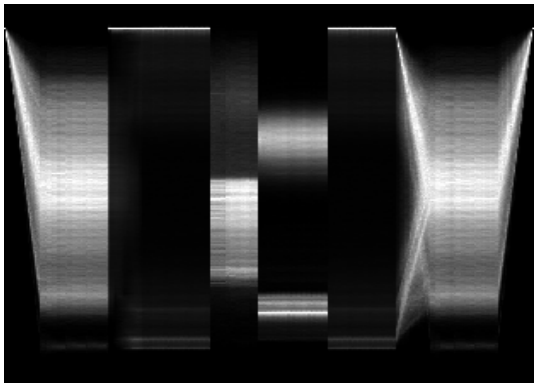


Figure 10.4: Visual rhythm by histogram computed from the video "voyage.mpg".

connected component with an inclined edge, so-called fade edge F_E, regarding the time axis, can be identified (as illustrated in Fig. 10.1(c)). In Fig. 10.5, we illustrate an example of application of a thresholding operator.

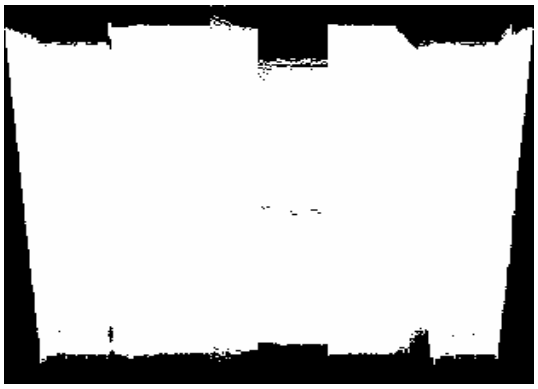


Figure 10.5: Image segmentation using thresholding ($threshold = 1$).

Morphological filtering. In order to reduce noise, we consider a morphological filtering given by an opening followed by a closing. The structuring element used is a vertical line ($\lambda = 13$) to preserve the time information. In Fig. 10.7, we illustrate an example of application of a thresholding operator.

Morphological gradient and thinning. Since the filtered image is a binary image and we are interested in detecting the fade edges, a simple edge detection method is used, the external morphological gradient given by the difference between the dilated image and the original one. This operation defines a new image consisting of connected

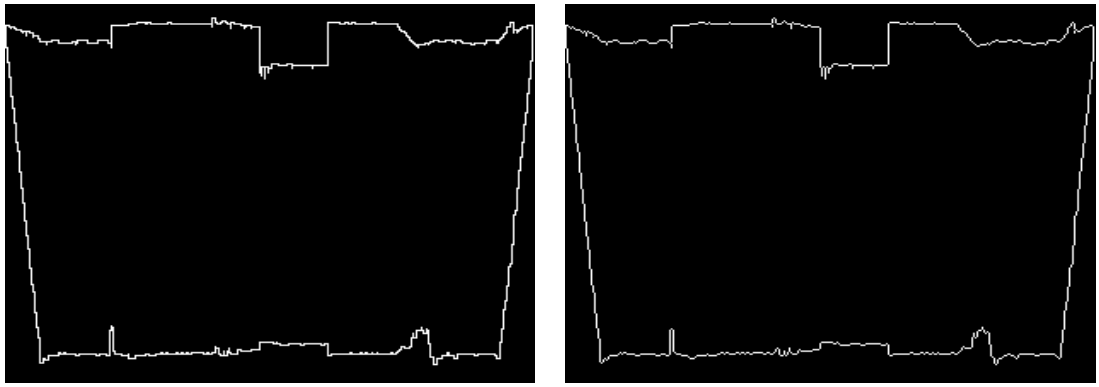
(a) (b)

Figure 10.6: Visual rhythm filtered: (a) the thresholded visual rhythm and (b) the result of the morphological filtering (opening followed by a closing). The vertical size of the structuring element is 13.

components related to the edges E of the filtered image. Unfortunately, the fade edge F_E (as illustrated in Fig. 10.8) is a subset of E with specific size and inclination, that is, $F_E \subseteq E$. This is why we introduce the line approximation step, to segment an edge and to label the different sub-segments of each edge. But to facilitate the computation of the line segments and to eliminate the ambiguity for identifying the line segments, an operation to simplify the components in a one-pixel-thin line is necessary. This is why we introduce the thinning operation [45] before applying the algorithm for curve approximation. In Fig. 10.7, we illustrate an example of gradient computation and the result of the thinning of the gradient result.

Line approximation. The quality of the results is directly associated with the method used to decompose the edge set E into a set of discrete straight line segments. Here, we used an optimization algorithm described by Dunham [25] to transform a curve into a minimal number of segments according to a given error. In Fig. 10.8, we illustrate an example of the line approximation from an edge component.

Segment filtering. The fade edges correspond to the connected components that are preserved by application of an inclination filtering followed by a size filtering. Unfortu-



(a) (b)

Figure 10.7: Morphological gradient and thinning: (a) result of the morphological gradient and (b) result of the thinning applied to the image illustrated in (a).

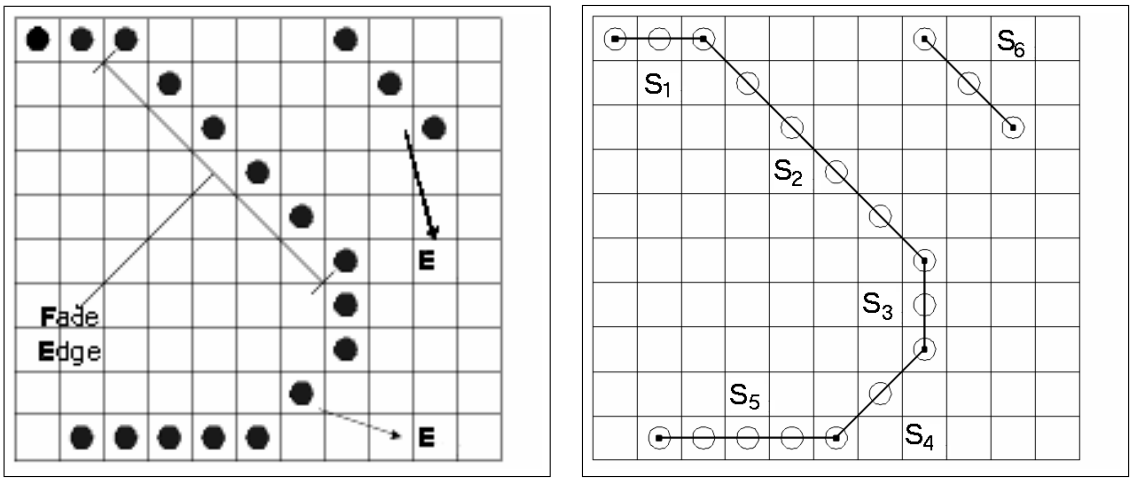
nately, this correspondence is not one-to-one, i.e., all fades are represented by oblique lines, but oblique lines do not always correspond to a fade. In Fig. 10.9, we illustrate an example in which the fade-in (Fig. 10.9(a)) and the fade-out (Fig. 10.9(b)) are identified.

In Fig. 10.10, we illustrate a real example of fade detection considering visual rhythm by histogram, where the permitted inclination is between -44^o and 44^o and the minimal size of the fade edges is 100 pixels. These values have been defined empirically.

10.4 Experiments

Here, we use 46 videos. The choice of videos was associated with the presence of the different events, such as, cut, dissolve, wipe, flash, zoom-in, zoom-out, pan, tilt, object motion, camera motion. For our experiments, the average duration of the fades is 14.2 frames. Following, we describe two experiments realized.

Experiment 1. For analysis of the histogram width, it is necessary for each histogram to compute the histogram width ($l = 0$). So, a $1\mathbb{D}$ image is created from computation of all histogram width values in which we try to detect the regions with a gradient having a specific inclination. The problem of this method is its insensitivity to spatial arrangements, i.e., two histograms completely different but with the same histogram width are



(a) (b)

Figure 10.8: Example of a curve approximation: (a) edge image and (b) segmentation of the edge in sub-segments.

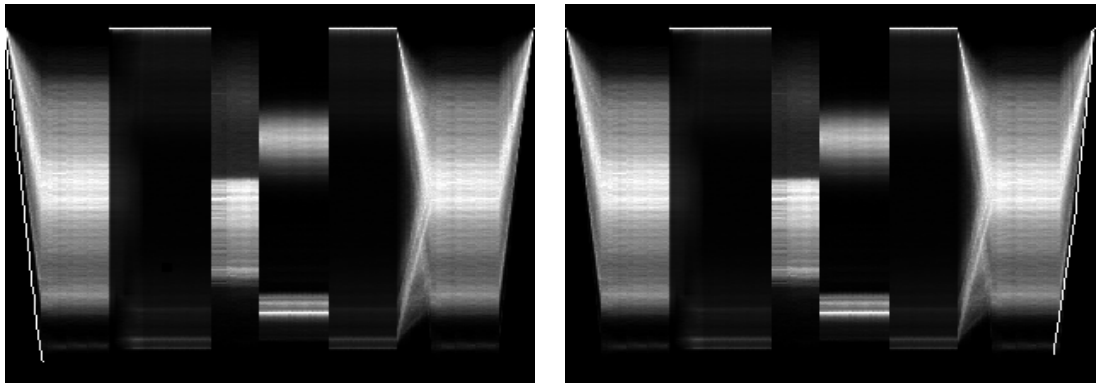
represented by the same value. In Table 10.1, we list the results of this experiment.

Experiment 2. This experiment is associated with the proposed method, where the parameters of size and inclination are 100 and $[-44^o, 44^o]$, respectively. Furthermore, we realize an analysis of the duration of the fade. In these experiments, we distinguished the long fades (duration $\geq$ 15 frames) which are, in general, the easiest ones to detect. In Table 10.1, we list the results of this experiment.

Experiment 3. In [26] was proposed a method to detect fades based on analysis of mean and variance of each frame. This method works very well for identification of exact interval for long fades, but fails for short fades, and also the fades are not identified if their first or last frames are not completely monochrome. The number of false detections may be very high if we try to identify the short fades. In Table 10.1, we list the results of this experiment.

10.5 Conclusions and discussions

In this chapter, a method for fade detection was proposed. An important contribution of this method is the application of operators of mathematical morphology, digital topology



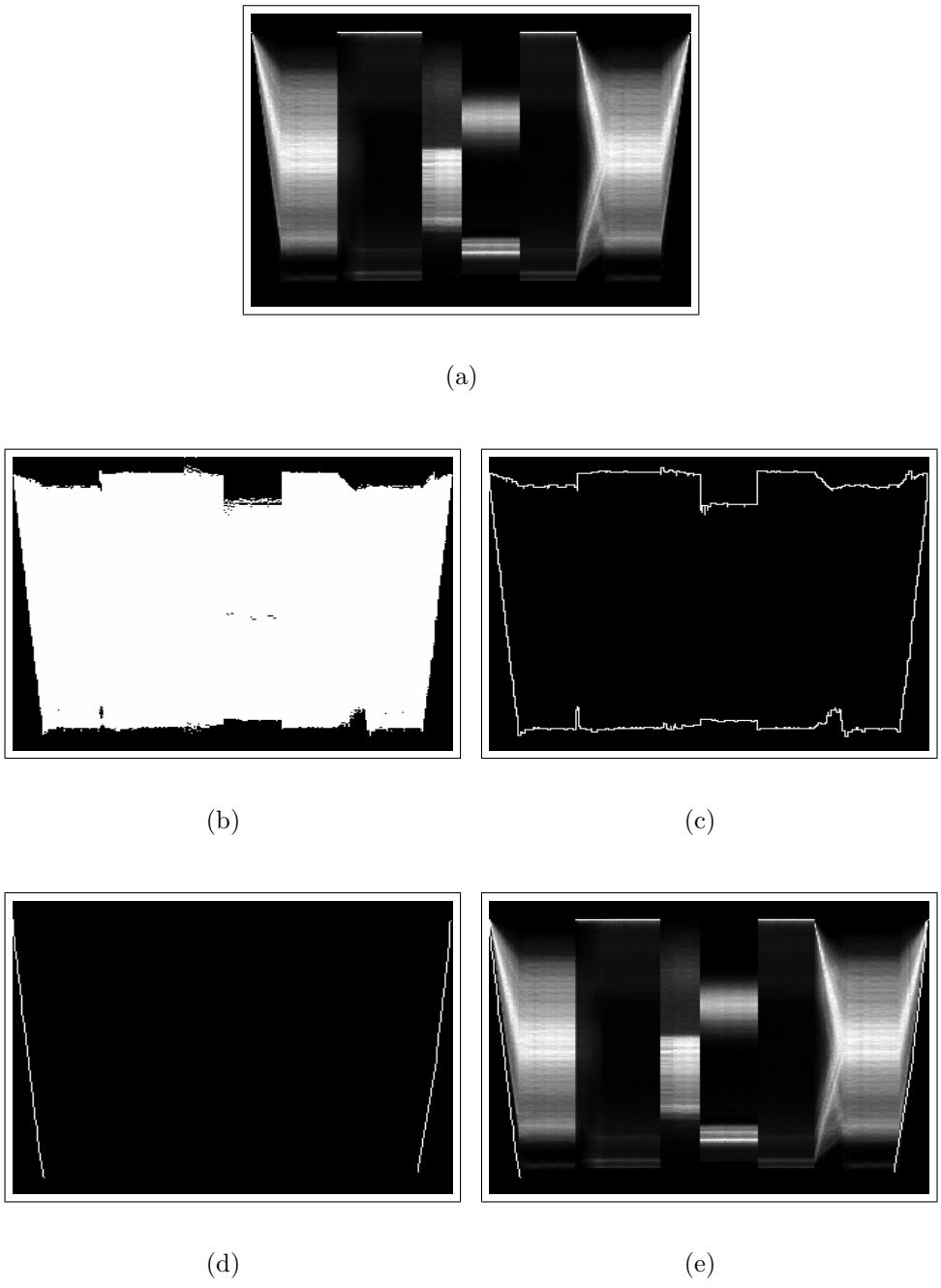
(a) (b)

Figure 10.9: Fade detection: (a) result of the fade-in detection is superimposed on the visual rhythm by histogram and (b) result of the fade-out detection is superimposed on the visual rhythm by histogram.

Exp	Type	Flash	Detected	False	Recall	Precision	Error
1: all fades	Commercial	59	37	16	63%	70%	27%
2: all fades	Commercial	59	56	18	95%	76%	30%
2: long fades	Commercial	20	19	1	95%	95%	5%
3: long fades	Commercial	20	17	10	85%	63%	50%

Table 10.1: Results of the experiments for fade detection.

and discrete line analysis to solve a problem of video segmentation. The use of visual rhythm by histogram (VRH) allows the detection of short fades with a small ratio of false detections. From this work, we observed that the VRH presents an adequate simplification of the video content, which can constitute the basis for future works concerning, for example, other video events such as flashes from detection of their correspondent patterns. In Table 10.1, the quality measures are given. Notice the result corresponding to error and precision rates for long fades, in which our method is much better with respect the other implemented methods.



(a)

(b) (c)

(d) (e)

Figure 10.10: Fade detection process: (a) visual rhythm by histogram (VRH) computed from the video "voyage.mpg"; (b) thresholding; (c) gradient; (d) line filtering result; and (e) result superimposed on the VRH.

Chapter 11

Conclusions and future work

In this work, we transformed the problem of identification video transition into a problem of $2\mathbb{D}$ image segmentation thanks to the correspondence between video events and transformed image patterns. We basically worked with two transformations: the visual rhythm by sub-sampling and the visual rhythm by histogram. The former corresponds to sub-sampling pixels in each frame, and the latter to the histogram information of each frame. Each transformation presents specific patterns that correspond to different video events, thus to identify video transitions we need to identify their correspondent patterns on the visual rhythm. We summarize, below, the main contributions of this work methods for video transition identification are:

Cut detection - identification of sharp vertical line transitions on the visual rhythm (by sub-sampling and by histogram) because these patterns correspond to video cuts. The basic idea behind of this method is to determine the number of non-zero points of the horizontal gradient image that are vertically aligned [31, 36].

Dissolve detection - identification of gradual vertical transition on the visual rhythm by sub-sampling. Two approaches were considered: the multi-scale approach [32, 30] and the sharpening approach [35]. While the former directly identifies the transitions, the latter tries to eliminate them.

Flash detection - identification of light vertical lines on the visual rhythm by sub-sampling. Also, two approaches were considered [36]: the white top-hat filtering and the max-tree filtering. While the former is a variant of the cut detection method,

the latter considers a data structure created from the statistical values of each visual rhythm column.

Fade detection - identification of inclined edges on the visual rhythm by histogram [33]. Due to the fade definition, the number of histogram bins increases or decreases according to the fade type, this effect produces inclined edges on the visual rhythm.

According to the performed experiments, we observe the effectiveness of our approach in which good results in regard to the literature were obtained. Taking into account the local features, as well as the features of the methods, the methods for cut, dissolve and flash detection can be easily extended to the $3\mathbb{D}$ case. In these cases, only the relationship of the neighborhood may be considered.

In order to improve the results of our methods, a study about the threshold process will be very interesting because, here, we consider only a global threshold. In regard to specific methods, some future work can be considered:

Flash detection - a flash is characterized by a sudden increase in intensity in some frames. Therefore, the number of pixels with large grayscale values increases producing a "discontinuity" between frames, before, during and after a flash. This effect is illustrated in Fig. 11.1. Maybe the first step to cope with the identification of this pattern is to apply a temporal opening followed by a temporal closing.

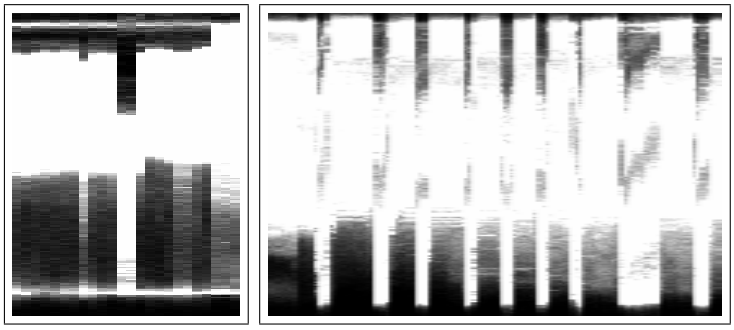


Figure 11.1: Example of orthogonal discontinuities present in visual rhythm by histogram. Both represent flashes.

Fade detection - the method for fade detection proposed here uses a line approximation operator, but this operator presents some problems if we have closed curves. To

avoid this problem, a morphological approach may be used in which a directional opening must be considered.

Dissolve detection - sometimes, the size of the transitions is an important attribute, mainly, to classify the video sequences. Thus, if we need to identify all transitions with a specific size, we must consider an algorithm in which these parameters are taken into account. In [34] it can be found a directional and parametrized algorithm on gradual transition detection.

From this work, we observed also that the visual rhythm presents an adequate simplification of the video content, which might constitute the basis for another future work concerning, for example, matching of video sequences and identification of key-frames.

Bibliography

[1] A. Akutsu and Y. Tonomura. Video tomography: an efficient method for camerawork extraction and motion analysis. In *ACM Multimedia*, pages 349–356, 1994.

[2] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba. Video indexing using motion vectors. In *SPIE Visual Comunications and Image Processing*, volume 1818, pages 1522–1530, 1992.

[3] R. Baeza-Yatez and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] F. Beaver. *Dictionary of Film Terms*. Twayne Publishing, New York, 1994.

[5] G. Bertrand, J.-C. Everat, and M. Couprie. Image segmentation through operators based upon topology. *Journal of Electronic Imaging*, 6:395–405, 1997.

[6] A. D. Bimbo. Special issue on Color and texture analysis. *Pattern Recognition*, 32(10), 1999.

[7] A. D. Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, 1999.

[8] A. D. Bimbo, V. Castelli, S. Chang, and C. Li. (Eds.). Special issue on Document image understanding and retrieval. *Computer Vision and Image Understanding*, 75(1/2), 1999.

[9] A. D. Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 19(2):121 – 132, February 1997.

[10] A. D. Bimbo, P. Pala, and L. Tanganelli. Retrieval of commercials based on dynamics of color flows. *Journal of Visual Languages and Computing*, 11:273–285, 2000.

[11] M. M. Blattner and R. B. Dannenberg. *Multimedia Interface Design*. ACM Press, 1992.

[12] R. Brunelli, O. Mich, and C. M. Modena. A survey on the automatic indexing of video data. *Journal of Visual Communication and Image Representation*, 10(6):78–112, June 1999.

[13] J. Canny. A computational approach to edge detection. *IEEE Trans. on PAMI*, 8(6):679–698, 1986.

[14] R. G. C. Catell. *Object-Oriented and Extended Relational Database-Systems*. Adison-Wesley Publishing Company, Inc, 1994.

[15] T. Chen. (Ed.). Special issue on Multimedia processing - Part 1. *Proceedings of the IEEE*, 86(5), 1998.

[16] S.-C. Cheung and A. Zakhor. Efficient video similarity measurement search. In *Proc. of the IEEE ICIP*, volume 1, pages 85–88, Vancouver, 2000.

[17] M. G. Chung, J. Lee, H. Kim, S. M.-H. Song, and W. M. Kim. Automatic video segmentation based on spatio-temporal features. *Korea Telecom Journal*, 4(1):4–14, 1999.

[18] G. B. Coleman and H. C. andrews. Image segmentation by clustering. *Proceedings of the IEEE*, (5):773–785, May 1979.

[19] A. de A. Araújo. Texture analysis: A review. *Revista Tecnologia e Ciência*, 1(2-3), 1987.

[20] A. de A. Araújo and S. J. F. Guimarães. Recuperação de informação visual com base no conteúdo em imagens e vídeos digitais. *RITA*, 7(2), Dezembro 2000.

[21] G. V. de Wouwer. *Wavelets For Multiscale Texture Analysis*. PhD thesis, Universitaire Instelling Antwerpen, 1998.

[22] C.-H. Demarty. *Segmentation et Structuration d'un Document Vidéo pour la Caractérisation et l'Indexation de son Contenu Sémantique*. PhD thesis, École Nationale Supérieure des Mines de Paris, Janvier 2000.

[23] M. Drew, Z.-N. Li, and X. Zhong. Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences. In *Proc. of the IEEE ICIP*, volume 1, Vancouver, 2000.

[24] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[25] J. G. Dunham. Optimum uniform piecewise linear aproximation of planar curves. *IEEE Trans. on PAMI*, 8(1):67–75, 1986.

[26] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull. Fade and dissolve detection in uncompressed and compressed video sequences. In *Proc. of the IEEE ICIP*, pages 299–303, 1999.

[27] B. Günsel and A. M. Tekalp. Shape similarity matching for query-by-example. *Pattern Recognition*, 31(7):931–944, 1998.

[28] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.

[29] V. N. Gudivada and V. V. Raghavan. (Eds.). Special issue on Content-based image retrieval systems. *IEEE Computer*, 9, 1995.

[30] S. J. F. Guimarães, A. A. Araújo, M. Couprie, and N. J. Leite. An approach to detect video transitions based on mathematical morphology. In *submitted for ICIP*, 2003.

[31] S. J. F. Guimarães, M. Couprie, N. J. Leite, and A. A. Araújo. A method for cut detection based on visual rhythm. In *Proc. of the IEEE SIBGRAPI*, pages 297–304, Brazil, 2001. ISBN 0769513301.

[32] S. J. F. Guimarães, M. Couprie, N. J. Leite, and A. A. Araújo. Identification of video transitions by multi-scale gradient analysis. Technical report, Submitted for Institute of Computing, Brazil, Aug 2002.

[33] S. J. F. Guimarães, M. Couprie, N. J. Leite, and A. A. Araújo. Video fade detection by discrete line identification. In *Proc. of the 16th ICPR*, volume 2, pages 1013–1016, Quebec, Canada, Aug 2002. ISBN 0 7695 1699 8.

[34] S. J. F. Guimarães, N. J. Leite, M. Couprie, and A. A. Araújo. A directional and parametrized algorithm to gradual transition detection. In *Proc. of the IEEE SIBGRAPI*, Brazil, 2002.

[35] S. J. F. Guimarães, N. J. Leite, M. Couprie, and A. A. Araújo. Video transition sharpening based on flat zone analysis. In *NSIP (accepted)*, Italy, June 2003.

[36] S. J. F. Guimarães, M. Couprie, A. A. Araújo, and N. J. Leite. Video segmentation based on 2D image analysis. *Pattern Recognition Letters*, 24:947–957, April 2003.

[37] N. Haering. *A Framework for the Design of Event Detectors*. PhD thesis, B. Sc. University of Edinburgh, Scotland, 1999.

[38] N. Haering and N. da Vitoria Lobo. A framework for designing event detectors. *International Symposium for Computer and Information Systems*, 1999.

[39] A. Hampapur, R. Jain, and T. E. Weymoth. Production model based digital video. *Multimedia Tool and Aplications*, 1:9–46, 1995.

[40] R. M. Haralick. Statistical and structural approaches to textures. *Proceedings of the IEEE*, 67(5), 1979.

[41] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Inteligence*, 17:185–203, 1981.

[42] P. Joly and H.-K. Kim. Efficient automatic analysis of camera work and microsegmentation of video using spatiotemporal images. *Signal Processing: Image Communication*, 8:295–307, 1996.

[43] J. Kanai and H. Baird. (Eds.). Special issue on Document image understanding and retrieval. *Computer Vision and Image Understanding*, 70(3), 1998.

[44] S. Khoshafian and A. B. Baker. *Multimedia and Imaging Databases*. Morgan Kaufmann, San Francisco, 1996.

[45] T. Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *CVGIP*, 48:357–393, 1989.

[46] M.-S. Lee, Y.-M. Yang, and S.-W. Lee. Automatic video parsing using shot boundary detection and camera operation analysis. *Pattern Recognition*, 34:711–719, 2001.

[47] N. J. Leite and S. J. F. Guimarães. Morphological residues and a general framework for image filtering and segmentation. *EURASIP Journal on Applied Signal Processing*, 2001(4):219–229, December 2001.

[48] C. Leung. (Ed.). Special issue on Content-based image indexing and retrieval. *Image and Vision Computing*, 17(7), 1999.

[49] M. S. Lew. *Principles of visual information retrieval*. Springer-Verlag, 2001.

[50] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *SPIE Image and Video Processing VII*, volume 3656, pages 290–301, Jan 1999.

[51] S. Lu and K. Fu. A syntactic approach to texture analysis. *Computer Graphics and Image Processing*, (7):303–330, March 1978.

[52] J. Maeda. Method for extracting camera operations to describe sub-scenes in video sequences. In *IS&T/SPIE*, volume 2187, pages 56–67, 1994.

[53] M. K. Mandal, F. Idris, and S. Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *Image and Vision Computing*, 17(7):513–529, May 1999.

[54] G. Matheron. *Random Sets and Integral Geometry*. John Wiley, 1975.

[55] R. Milanese and M. Cherbuliez. A rotation, translation, and scale-invariant approach to content-based image retrieval. *Journal of Visual Communication and Image Representation*, 10(6):186–196, June 1999.

[56] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Visual Database Systems II*, pages 113–127. Elsevier Science Publishers, 1992.

[57] C. W. Ngo, T. C. Pong, and R. T. Chin. Detection of gradual transitions through temporal slice analysis. In *Proc. of the IEEE CVPR*, pages 36–41, 1999.

[58] S. Panchanathan and B. Liu. (Eds.). Special issue on Indexing, browsing, and retrieval of images and video. *Journal Visual Communication and Image Representation*, 7(4), 1996.

[59] S. Ravela and R. Manmatha. Image retrieval by appearance. In *Proc. of the ACM/SIGIR*, pages 278–303, Philadelphia-PA, USA, 1997.

[60] V. Roth. Content-based retrieval from digital video. *Image and Vision Computing*, 17(7):531–540, May 1999.

[61] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table-of-contents for videos. *ACM Multimedia Systems Journal, Special Issue Multimedia Systems on Video Libraries*, 1999.

[62] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Trans. on Image Processing*, 7(4):555–570, 1998.

[63] J. Serra. *Image Analysis and Mathematical Morphology*, volume 1. Academic Press, 1982.

[64] J. Serra. *Image Analysis and Mathematical Morphology: Theoretical Advances*, volume 2. Academic Press, 1988.

[65] I. K. Sethi and N. Patel. A statistical approach to scene change detection. In *Proc. of the SPIE Storage and Retrieval for Image and Video Database III*, volume 2420, pages 329–338, USA, 1995.

[66] G. Sharma, M. J. Vrhel, and H. J. Trussell. Color indexing for multimedia. 86(6):1088 – 1108, June 1998.

[67] D. Shen, W. him Wong, and H. H. S. Ip. Affine-invariant image retrieval by correspondence matching of shapes. *Image and Vision Computing*, 17(7):489–499, May 1999.

[68] P. Soille. *Morphological Image Analysis*. Springer-Verlag, 1999.

[69] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[70] A. Tekalp. (Ed.). Special issue on Multimedia processing - Part 2. *Proceedings of the IEEE*, 86(6), 1998.

[71] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. Videomap and videospaceicon: Tools for anatomizing video content. In *ACM Interchi*, pages 131–136, 1993.

[72] Y. Wang, Z. Liu, and J.-C. Huang. Multimedia content analysis. *IEEE Signal Processing Magazine*, pages 12–36, 2000.

[73] X. Wen, T. D. Huffmire, H. H. Hu, and A. Finkelstein. Wavelet-based video indexing and querying. *Multimedia Systems*, 7:350–358, 1999.

[74] B.-L. Yeo. *Efficient Processing of Compressed Images and Video*. PhD thesis, Department of Electrical Engineering, Princeton University, January 1996.

[75] R. Zabih, J. Miller, and K. Mai. Feature-based algorithms for detecting and classifying scene breaks. In *ACM ICMCS*, USA, Nov 1995.

[76] R. Zabih, J. Miller, and K. Mai. Video browsing using edges and motion. In *Proc. of the IEEE CVPR*, pages 439–446, 1996.

[77] R. Zabih, J. Miller, and K. Mai. A feature-based algorithms for detecting and classifying scene breaks. *Multimedia Systems*, 7:119–128, 1999.

[78] D. Zhang, W. Qi, and H. J. Zhang. A new shot boundary detection algorithm. *Lectures notes in Computer Science*, 2195:63–70, 2001.

[79] H. Zhang, A. Kanknhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1:10–28, 1993.