

# Abstract

This thesis investigates two different, yet related aspects of video indexing: temporal segmentation and content classification. Temporal segmentation, often performed by detecting transitions between shots, is required in the early stages of video indexing. This is because a shot can be effectively considered as the smallest indexing unit and higher level concepts are often constructed by combining and analyzing the inter and intra-shot relationships. Automatic video classification, on the other hand, enables efficient cataloging and retrieval with large video collections.

We have proposed algorithms for detecting transitions between shots. Hard cuts are detected by recording the peaks in the frame difference curve using an adaptive threshold computed from a local window. Fades and dissolves are detected by inspecting the characteristics of the production models in terms of frame luminance mean and variance. In order to guard against noise and motion which would cause similar effects, constraints derived from the production models are applied. When comparing against two other tools for detecting shot transitions, our algorithms show much better performance.

For video classification, we aim at solving the task of automatic identification of video genres, specifically cartoons, commercials, music, news and sports. We propose a set of computational features originating from our study of editing effects, motion, and color used in videos. These features besides representing human understanding of typical attributes of different video genres, are also inspired by the techniques and rules used by many directors to endow specific characteristics to a genre-program which lead to certain emotional impact on viewers. This research, goes beyond the existing work with a systematic analysis of trends exhibited by each of our features in investigated genres it enables an understanding of the similarities, dissimilarities, and also likely confusion between genres. Classification results from our experiments on several hours of video establish the usefulness of this feature set. We also explore the issue of video clip duration required to achieve reliable genre identification and demonstrate its impact on classification accuracy.

# Acknowledgments

I would like to thank Professor Svetha Venkatesh and Dr Chitra Dorai for their guidance and encouragement that kept me going till the end. Without them, this thesis would have never been possible. I am very grateful for the personal supports from Professor Venkatesh who have turned my desire for completing the honours year at Curtin into a reality. For this matter, thanks are also due to Dr John Bui and Mr Andrew Marriott.

I would like to thank all my friends for making 4 years living away from Vietnam, my home country, a pleasant experience. Special thanks are, however, to Ken and Russell. Ken is simply being there when I need him. Russell, apart from being my study buddy and putting up with me, suggested the distinction of the colour used in cartoons which was eventually formalized in this thesis.

At the technical end, thanks are due to Brett, Kim and Mihai for their occasional instructions and assistance.

Last but not least, I am indebted to my parents back home whose sacrifice, patience and love for their children are impossible to measure. This thesis is dedicated to them.

# Preface

Part of this work has been written into a paper titled “*Automatic Genre Identification for Content-Based Video Categorization*” and submitted to 15<sup>th</sup> International Conference on Pattern Recognition held in Barcelona, Sep 3-9,2000. The paper is included in appendix B.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Content-Based Video Indexing and Retrieval . . . . .	1
1.2	Aims and Approach . . . . .	2
1.3	Contributions . . . . .	3
1.4	Structure of the Thesis . . . . .	4
<b>2</b>	<b>Background and Previous Work</b>	<b>5</b>
2.1	Overview . . . . .	5
2.2	Shot-Based Segmentation of Video Sequence . . . . .	8
2.2.1	Shot Transitions in Video Production . . . . .	8
	Cut . . . . .	8
	Fade . . . . .	9
	Dissolve . . . . .	9
	Wipe . . . . .	9
2.2.2	Full-Pixel Domain Techniques . . . . .	10
	Pixel Wise Change Detection . . . . .	10

---

Color Histogram Comparison . . . . .	11
Motion-Based Approaches . . . . .	12
Production Model-Based Approaches . . . . .	13
Edge-Based Approaches . . . . .	14
Temporal Slice Analysis . . . . .	15
2.2.3 Compressed Domain Techniques . . . . .	16
MPEG Video Compression Standard . . . . .	16
DC Image . . . . .	17
DCT Coefficients . . . . .	18
Bit Rates . . . . .	19
Macro-Block Information/Motion Vectors . . . . .	20
2.2.4 Threshold Selection . . . . .	21
2.3 Hierarchical Video Content Characterization . . . . .	22
2.4 Summary . . . . .	24
<b>3 Shot Transition Detection with Accurate Boundary Identification</b>	<b>25</b>
3.1 Hard Cuts . . . . .	25
3.2 Fades . . . . .	27
3.3 Dissolves . . . . .	30
3.4 Eliminating False Positives Using Color Histograms . . . . .	38
3.5 Experimental Results . . . . .	39

---

3.5.1	The Data . . . . .	39
3.5.2	Performance Parameters . . . . .	40
3.5.3	Results and Discussion . . . . .	43
3.6	Summary . . . . .	45
<b>4</b>	<b>Video Genre Identification</b>	<b>46</b>
4.1	Notations . . . . .	46
4.2	Feature Extraction . . . . .	48
4.2.1	Editing Characteristics . . . . .	48
	Shot Length . . . . .	48
	Transition Types . . . . .	48
4.2.2	Motion Estimation . . . . .	49
	Camera Movement . . . . .	50
	Motion Continuity . . . . .	51
	Dynamic Scenes . . . . .	52
	Static Scenes . . . . .	52
4.2.3	Color Statistics . . . . .	53
	Color Histograms . . . . .	53
	Brightness . . . . .	53
	Saturation . . . . .	55
4.3	Experimental Results . . . . .	56
4.4	Summary . . . . .	58

---

<b>5 Conclusion</b>	<b>60</b>
5.1 Summary . . . . .	60
5.2 Future Work . . . . .	61
<b>A Shot Transition Types</b>	<b>63</b>
<b>B False Positives Identified by the Verification Step</b>	<b>67</b>
<b>C Sample Classification Trees and Their Confusion Matrices</b>	<b>70</b>
C.1 Labels of Tree Nodes . . . . .	70
C.2 Five Categories: Cartoons, Commercials, Music, News and Sports . . . . .	71
C.2.1 For 40-second clips . . . . .	71
C.2.2 For 60-second clips . . . . .	74
C.2.3 For 80-second clips . . . . .	77
C.3 Four Categories: Commercials, Music, News and Sports . . . . .	78
C.3.1 For 40-second clips . . . . .	78
C.3.2 For 60-second clips . . . . .	81
C.3.3 For 80-second clips . . . . .	83
C.4 Four Categories: Cartoons, Music, News and Sports . . . . .	84
C.4.1 For 40-second clips . . . . .	84
C.4.2 For 60-second clips . . . . .	87
C.4.3 For 80-second clips . . . . .	89
C.5 Four Categories: Cartoons, Commercials, News and Sports . . . . .	90

---

C.5.1	For 40-second clips . . . . .	90
C.5.2	For 60-second clips . . . . .	93
C.5.3	For 80-second clips . . . . .	95
C.6	Four Categories: Cartoons, Commercials, Music, and Sports . . . . .	96
C.6.1	For 40-second clips . . . . .	96
C.6.2	For 60-second clips . . . . .	99
C.6.3	For 80-second clips . . . . .	101
C.7	Five Categories: Cartoons, Commercials, Music, and News . . . . .	102
C.7.1	For 40-second clips . . . . .	102
C.7.2	For 60-second clips . . . . .	104
C.7.3	For 80-second clips . . . . .	106
<b>D</b>	<b>The paper submitted to ICPR'2000</b>	<b>108</b>

# List of Figures

2.1	A sample video indexing and retrieval system . . . . .	6
2.2	The layered structure of MPEG encoding . . . . .	16
2.3	Predictive Relationship between I, P and B frames . . . . .	17
2.4	A sample video classification tree . . . . .	23
3.1	Color histograms and its local mean-ratio transform . . . . .	27
3.2	Mean and variance curve during a fade . . . . .	31
3.3	The first derivative of mean and the second derivative of variance during a fade . . . . .	31
3.4	Mean and variance curve during a dissolve . . . . .	35
3.5	The first order difference of mean and variance curve during a dissolve . . . . .	35
3.6	Hints for threshold selection . . . . .	36
3.7	Cover-precision and cover-recall . . . . .	42
4.1	Relation between $\mathcal{V}$ , $\mathcal{T}$ and $\mathcal{S}$ . . . . .	47
4.2	Shot length related characteristics for 50 samples . . . . .	49
4.3	Transition related characteristics for 50 samples . . . . .	50
4.4	$\mathcal{F}_3$ : Average camera motion . . . . .	51

---

4.5	$\mathcal{F}_4$ : Average length of motion runs . . . . .	51
4.6	The amount of dynamic scenes for 50 samples . . . . .	52
4.7	The amount of static scenes for 50 samples . . . . .	53
4.8	Color histogram related characteristics for 50 samples . . . . .	54
4.9	Brightness related characteristics for 50 samples . . . . .	55
4.10	Saturation related features for 50 samples . . . . .	56
4.11	Overview of the video genre classification system . . . . .	57

# List of Tables

2.1	Illusion of edit effects on temporal slice images . . . . .	15
3.1	Ground truth of test data for shot transition detection . . . . .	40
3.2	Shot transition detection results . . . . .	44
4.1	Genre classification results. . . . .	58
A.1	An example of a cut between frame 2 and 3 . . . . .	63
A.2	An example of a fade to/from black . . . . .	64
A.3	An example of a fade to/from white . . . . .	64
A.4	An example of a dissolve . . . . .	65
A.5	An example of a standard wipe . . . . .	65
A.6	An example of a “slide-in” wipe . . . . .	66
A.7	An example of complex wipe . . . . .	66
B.1	An falsely detected cut recognized during the verification step . . . . .	67
B.2	An falsely detected cut recognized during the verification step . . . . .	68
B.3	An falsely detected cut recognized during the verification step . . . . .	68

---

B.4 An falsely detected cut recognized during the verification step . . . . . 69

# Notations

Symbols	Meanings
$\mathbb{N}_m^n$	A set of integers $x$ with $m \leq x \leq n$
$\mathcal{V}$	The video sequence of interest
$X$	The width of video frames
$Y$	The height of video frames
$f_i$	The $i^{th}$ frame of video sequence $V$ indexed from 1. It is a feature vector with $f_i^x$ denoting the value for feature $x$ of this vector for the entire
$\mathcal{D}(f_i, f_{i+1})$	The similarity between frame $f_i$ and $f_{i+1}$ according to some metric.
$f_i(x, y)$	The pixel at position $(x, y)$ of frame $f_i$ . It is a feature vector with $f_i^x$ denoting the value for feature $x$ of this vector for this particular pixel
$t_i$	A transition frame vector which contains all features computed jointly from frame $f_i$ and $f_{i+1}$
$t_i^x$	The value of feature $x$ of frame transition $t_i$
$\mathcal{T}$	The set of all frame transitions $t_i$
$\mathcal{L}$	The set of all transitions label, $\mathcal{L} = \{shot, cut, fade, dissolve\}$
$\Omega^x$	The set of all frame transitions $t_i$ of type $x$ , $x \in \mathcal{L}$
$S_i$	A transition segment which can be either a shot, cut, fade, or dissolve
$\mathcal{S}$	The set of all transition segments $S_i$ in the video sequence
$\Gamma^x$	The set of all transitions segments $S_i$ of type $x$ , $x \in \mathcal{L}$
$\Delta^x$	The set of all frames $f_i$ of type $x$ , $x \in \mathcal{L}$

# Chapter 1

## Introduction

### 1.1 Content-Based Video Indexing and Retrieval

We are witnessing a revolution in the way information is produced, stored, manipulated and communicated. More information is stored in video format and made available to users daily. Managing video information is as important as managing textual information in traditional databases. As a result, video indexing and retrieval has recently emerged as a very active research field. There are two important aspects, among many others, surrounding the development of a video indexing and retrieval systems: temporal segmentation and content classification.

Temporal segmentation, often performed by detecting transitions between shots, is required in the early stages of video indexing. A shot is defined as an image sequence that presents continuous action which is captured from a single operation of single camera. In other words, it is a sequence of images generated by camera from the time it starts recording the action to the time it stops recording the images (Hampapur *et al.*, 1994). Shots are joined together in the editing stage of video production to form the complete sequence. Shots can be effectively considered as the smallest indexing unit where no changes in scene content can be perceived and higher level concepts are often constructed by combining and analyzing the inter and intra-shot relationships. Detecting transitions between shots is also useful in video compression. For example, shot transitions such as fades and dissolves often cause the blockiness artifacts in video sequence compressed using motion compensation, since the motion compensation algorithms fail in fade and dissolve regions and produce inaccurate motion vectors. If these transitions can be detected, their frames would be coded without motion compensation and therefore improve the picture quality.

Classification of videos into categories based on their semantic content to form a classification hierarchy is useful for video search and retrieval, since it mirrors the way humans perceive the content of a video sequence which is done at different abstraction levels. Textual databases have been indexed and classified in this manner and proven to be effective. While humans can quickly interpret the imbedded semantic content from information carried by different modalities such as text, speech, audio, and color pattern, and objects. Computer understanding of a video sequence is still in primitive stage and not incorporated in most video databases reported in the literature. Apart from video indexing and retrieval, the study in automatic content understanding is also useful in areas such as automatic video composition and sequencing.

## 1.2 Aims and Approach

The purpose of this thesis is to investigate the problem of shot transition detection and video classification for video indexing and retrieval. In particular, it is aiming at answering following questions.

- How can transitions between shots in a video sequence can be detected, classified and measured?
- What are the patterns in the visual content of a particular video genre?
- How can these patterns be captured, enabling the automatic recognition of the video genre?

Due to the importance of shot transition detection, many techniques have been proposed in the literature for detecting transitions between video shots. Therefore, rather than investigating new features in which the effect of shot transitions is amplified and detected, we focus on improving existing algorithms. In addition, algorithms should be tested on a large and comprehensive data set. For detecting gradual transitions, we only focus on approaches that can accurately classify and measure the temporal extent of a transition.

Genre identification problem is approached by studying features that can be computed from visual contents of a video sequence. Features can be reflected in individual frames or changes between frames. Apart from relying on human understanding of typical attributes of different video genres and reusing features well established in the literature, we investigate the techniques and rules used by many directors to endow specific characteristics to a genre-program which leads to certain emotional impact on the viewers. In order to confirm the usefulness of the feature set, it should be tested on a large and comprehensive video databases. Our focus in this research is on the feature set itself; therefore, a simple classification tool should be used.

## 1.3 Contributions

There are different novel aspects of our research in both shot transition detection and video genre classification.

- **Shot Transition Detection**

Based on the work by Yeo and Liu (1995b) on shot transition detection, we propose a new adaptive threshold that would reduce the effects of noise, motion and other artifacts on the frame difference curves, and at the same time enhance the peaks caused by real shot cuts. This leads to the improvements in the performance of the hard cut detector.

Based on the previous work on production model based techniques for gradual transition detection by Alattar (1993) and Alattar (1997), we devise two-step algorithms for detecting fades and dissolves. Suspected transitions are recorded in the first step based on the characteristics of frame luminance mean and variance. Different constraints are applied in the second step to eliminate false positives caused by object and camera motion. Instead of selecting thresholds based on traditional “Trial-and-Error” approach, robust adaptive thresholds are derived explicitly from the mathematical models of transitions.

In addition, we also propose a simple, yet effective technique for eliminating false positives from a list of detected transitions. This process is performed after cut, fade and dissolve detectors have been executed.

- **Video Genre Classification**

We have constructed a comprehensive 8 hours video data set composed of five different genres: cartoons, commercials, music, news and sports. This is greater than the largest data set reported in the literature addressing the same problem as us, which only contains 2 hours of video data from 4 genres.

We devise some new computational features which have not been investigated in video classification literature, e.g motion runs, color histograms, brightness and saturation.

This research goes beyond the existing work with a systematic analysis of trends exhibited by each of our features in different genres, and it enables an understanding of similarities, dissimilarities and also likely confusion between genres.

We also explore the issue of video clip duration required to achieve reliable genre identification and demonstrate its impact on classification accuracy.

## 1.4 Structure of the Thesis

The remainder of this thesis is organised as follows.

In Chapter 2, we review related literature. Work to be reviewed include the important components of a video indexing system, especially key-frames extraction and scene construction, basic approaches to shot transition detection in both full pixel domain as well as compressed domain, and existing approaches to video classification which can be addressed at different abstraction levels.

The next two chapters present the main contribution of this thesis.

In Chapter 3, we describe our approach to shot transition detection. Mathematical models for fades and dissolves are detailed. We show how robust thresholds can be established based on these models. On a large and comprehensive video data set, the performance of proposed algorithms are compared against two other existing STD methods in terms of precision and recall in detecting, classifying and measuring the temporal extent of transitions.

In Chapter 4, we present the proposed set of computational features which are grouped into editing effects, motion and colour. The trends of these features for each video genre are described and explained. This chapter also present the experimental results of video genre recognition using the proposed feature set and C4.5 decision tree classification tool.

Finally, in Chapter 5, we conclude and provide some directions for future work. Representative decision trees produced by C4.5 for each testing criterion are provided in Appendix A. Appendix B is the attachment of the paper based upon this work and that has been submitted to 15<sup>th</sup> International Conference on Pattern Recognition.

## Chapter 2

# Background and Previous Work

As set out in the previous chapter, we have two aims: video segmentation and video classification. In this chapter, we review work related to these two areas. We start with an overview of some components of a video database system supporting browsing, search and retrieval. In the second section, we describe some aspects of shot transitions in video production before presenting different approaches for detecting shot transitions. These can be roughly divided into two categories: pixel domain techniques and compressed domain techniques. Finally, we review work related to video classification and content characterization.

### 2.1 Overview

Using video as a primary multimedia data source requires effective mechanisms for browsing and retrieving the video from a database. Ignoring the process of data modeling, organization and management, the important components of a typical multimedia database system that supports video browsing, search and retrieval are shown in figure 2.1. The shaded components are subjects of our research.

The first component, which is essential to most video retrieval systems, is a temporal segmentation process whose task is to break a video sequence into meaningful segments to served as units to be indexed. This is normally achieved by detecting transitions between shots. Due to its ultimate importance, a extensive review of existing techniques for detecting shot transitions is presented in section 2.2.1.

After the video sequence is segmented into shots, various features can be extracted to index these shots. They include camera work information (e.g. pan, tilt, zoom, roll, track), scene activity

estimation based on motion, shot types (e.g. big close-up, medium close up, medium long shot, extreme long shot), editing characteristics (shot length, transition type), color features (e.g. color histograms, dominant colors, color moments, mean brightness, etc.), texture features (e.g. contrast, directionality, coarseness, etc.), shapes of dominant objects, and edge features (Zhang *et al.*, 1998). In addition, it is also possible to extract features without knowing the boundaries between shots.

Figure 2.1: A sample video indexing and retrieval system

Since a video sequence normally contains a large number of frames, it is rarely feasible or desirable to index and/or store all frames for browsing and retrieval purposes. Instead, an abstraction process is required, which may be applied at the individual segment level (Zhang *et al.*, 1998). The abstraction problem can be posed as the problem of best mapping the entire segment to a number of representative images, usually called *key-frames*. Key-frames can be extracted at fixed positions within a shot, e.g. at the beginning and the ending frames of a shot (Rui *et al.*, 1999). However, this approach is ineffective, since it does not take into account the dynamic nature of video shots. Alternatively, one can cluster frames within a shot based on frame differences and select the centroid of each clusters a key-frame (Ferman and Tekalp, 1998) and (Zhuang *et al.*, 1998). However, it is not easy to find more than one cluster within a shot. In Zhang *et al.* (1998), color and motion features are used. The first frame in each shot is chosen as a key-frame, and subsequent frames are compared progressively against it. A similar method is used in Hanjalic *et al.* (1997) where accumulated frame differences are used as a measure of the content variation within a shot. Based on this measure, the appropriate number of key-frames for each shot is automatically determined from the total allocated for the entire sequence, and

representative frames are selected using a minimization process. Xiong *et al.* (1997) propose a Spread-and-Seek algorithm which searches for key-frames sequentially and then extends the representative range of the key-frames as far as possible. While all the above techniques extract key-frames shots are detected, Sun *et al.* (1998) propose a clustering algorithm to extract a fixed number of key-frames which does not require shot boundary detection step.

While shots are the building blocks of a video, scenes, or story units consisting of connected or unconnected shots, convey the semantic meaning of the video to the viewers. Therefore, many researchers argue that the indexing and representation of video at a scene level is more appropriate than at the shot level. While low level image features are adequate for detecting shot boundaries, the detection of scene boundaries is far more difficult. It requires high-level concepts and intra-shot analysis, since there is no physical mark to indicate scene boundaries. Yeung *et al.* (1998) propose a time-constrained clustering algorithm to group similar shots into a scene. In this algorithm, shots are considered similar if they have similar visual content and are close to each other. Rui *et al.* (1998) propose a similar technique called time-adaptive clustering to group similar shots. Hammoud *et al.* (1998) cluster similar shots together and present the video as a *Time-Space graph* (TSG). Scenes are constructed based on the temporal relationship between clusters which can be extracted from TSG. Saraceno and Leonardi (1997), Pleiffer *et al.* (1998) and Gauch *et al.* (1999) combine audio analysis and shot detection tools to construct scenes, as audio breaks normally signal scene changes. Aigrain *et al.* (1998) propose a multimodal rule based approach. They first identify local rules about shot transition, shot repetition, editing rhythm; and then construct scenes, or *macro-segments*, by combining rules. At higher level, Hanjalic *et al.* (1999) propose a method for extract *global story units* (GSU) from a full-length movie, where GSU is defined as collections of temporally consecutive, and semantically inter-related events.

Content characterization and classification are also crucial to a fully automatic video indexing and retrieval system. Based on visual and audio features computed for each shot or the whole sequence, it is possible to recognize the genre of a video sequence, e.g. whether it is a live sport video or a newscast. The classification of a video can be done at different levels of abstraction. Since one major part of our research is to address this problem, we present a detailed review of existing approaches in section 2.3.

Semantic representations of video shots for browsing have become new challenges in content based video retrieval systems. Yeung *et al.* (1998) present an approach to construct a scene transition graph (STG) based on visual similarity and temporal relationships among shots. STG is a directed graph where each node represents a scene and each edge represents a temporal transition. To yield more compact representation, the system prunes “insignificant shots” that are determined by heuristics such as the number of shots in a node and the scene duration.

## 2.2 Shot-Based Segmentation of Video Sequence

### 2.2.1 Shot Transitions in Video Production

Editing is an important aspect of the video production process. At a *practical* level, it allows the creation of smooth-flowing picture development. The director can omit portions that are irrelevant, or distracting or containing errors. Editing also bridges space and time. *Artistically*, editing decisions have a direct influence on how audience responds to the video material; their interpretation; their emotional reactions. According to Millerson (1990), editing is mechanically concerned with:

1. The moment chosen to change from one shot to another (*cutting point*)
2. How the change is made (cut, fade, dissolve, wipe, and other effects), and the speed of this transition.
3. The order of shots (*shot sequencing*) and their duration (*cutting rhythm* )
4. Maintaining good continuity between shots.

In the remaining of this section, we present the descriptions of the four most common types of shot transitions {cut, fade, dissolve, wipe} and the syntactic meanings associated with them. These are essential for designing a shot transition detector and indicate useful features for characterizing the content of a video sequence. It should be noted that there are some other types of effects in video production which sometimes can not be precisely classified. They include split screen effects, super-imposition, chromatic keys, multi-image montages and other computer generated or manipulated effects.

#### Cut

The cut is the simplest and most common transition. It is an instantaneous change from one shot to another and can be seen as the shortest distance between two shots. When used correctly, the cut is the least obvious transition because it occurs quickly and appear natural (Wurtzel and Rosenbaum, 1995). A cut should be motivated by some element in the scene such as action, the beat of the music, dialogue, etc. There are some special situations where cutting is used to achieve special visual effects. One is a *montage* - a rapid succession of shots in sequence. Since the impact of montage is derived from the total effect of all the shots as a whole and not from any single cut, it is useful if a shot transition detection tool can detect montages as well. An example of a cut is included in Appendix A.1.

### **Fade**

There are two types of fades: fade-in and fade-out. A fade-in occurs when the picture information gradually disappears, leaving a blank screen. A fade-out occurs when the picture gradually appears from a black screen. A fade-in to or fade-out from black is the most common; however, it is possible to fade-in to or fade-out from any other colour. Unlike the cut which is sometimes not apparent, the fade is an obvious traditional device that punctuates a video segment as a period does the end of a sentence (Wurtzel and Rosenbaum, 1995). When a fade-out is followed by a fade-in, a pause in flow of action is introduced. Mood and pace vary with their relative speeds and pause time between them. Fade-out/in combinations can be used to connect slow-tempo sequences, where a change in time and space is involved. The fade can also be used to separate different TV program elements such as the main show material from commercial blocks. Examples of fades to/from black and white color are shown in Appendix A.1. A fade can be modeled mathematically and is detailed in section 3.2.

### **Dissolve**

A dissolve occurs when one whole picture fades away while another whole picture is appearing (see Appendix A.1). The dissolve provides a smooth restful transition, with minimum interruption of visual flow. The speed of the dissolve affects the overall mood and flow of the video sequence. A quick dissolve implies the concurrency of action while a slow dissolve suggests the difference in time and place. Dissolves are widely used as softened cuts to provide unobtrusive transitions for slow-tempo occasions where the violence of a cut would be disruptive. Therefore, dissolves are often used in dance and music pieces and in some transitions in drama. It is also used in live sport to separate slow motion replays from the live action. The mathematical model for a dissolve is presented in section 3.3.

### **Wipe**

A wipe occurs when one full-strength image is progressively replaced or pushed away or compressed by another full-strength image. The second image may appear at the side of the screen, the corner, the middle or several places at once. It then takes over the screen by following some geometric pattern. Three different types of wipes are shown in Appendix A.1. In addition, the edge of wipe can be made of soft or ‘move-in’ waves. A video editing kit can produce hundreds of different wipe patterns. The wipe often tells the viewers that a complete new scene will be shown, e.g. slow motion replays in sport videos. However, sometime its purpose is only to

inject some interest or fun into the shot sequence such as in game shows, children’s programs and commercials (Zettl, 1997).

## 2.2.2 Full-Pixel Domain Techniques

### Pixel Wise Change Detection

STD techniques based on pixel-level change detection, first proposed by Nagasaka and Tanaka (1992), are simple and computationally inexpensive. They rely on the following premise.

**Premise 1** *Properties of most pixels at the same image position do not change across frames of the same shot. However, they will change significantly across frames of different shots.*

The comparison of the pixels of two images across the same location can be formulated as

$$\mathcal{D}(f_i, f_{i+1}) = \frac{1}{X \times Y} \sum_{x=1}^X \sum_{y=1}^Y |f_{i+1}(x, y) - f_i(x, y)| \quad (2.1)$$

Instead of computing the sum of absolute pixel intensity differences, Zhang *et al.* (1993) count the number of changed pixels, i.e. those that differ more than a threshold  $\mathfrak{T}$  from their corresponding pixels in the previous image. Thus

$$\mathcal{D}(f_i, f_{i+1}) = \frac{1}{X \times Y} |\Upsilon| \quad (2.2)$$

where

$$\Upsilon = \{(x, y) \in \mathbb{N}_1^X \times \mathbb{N}_1^Y \mid |f_{i+1}(x, y) - f_i(x, y)| > \mathfrak{T}\} \quad (2.3)$$

The main drawback of STD techniques based on pixel level change detection is their sensitivity to noise and motion, especially camera motion, as it causes a global change of object positions in the image. This can be reduced by performing a global motion compensation and aligning images accordingly before computing the difference.

The above techniques for capturing the pixel level difference between two frames can be extended to a block of pixels, and this is more robust to noise and motion. An image is divided into uniform blocks and the number of changed blocks, instead of changed pixels, are counted. Based on the assumption of uniform second order statistic over a region, a *likelihood test* test is performed on each block to determine whether it has changed. Thus

$$\mathcal{D}(f_i, f_{i+1}) = |\Upsilon| \quad \text{with} \quad \Upsilon = \{i \in \mathbb{N}_1^{P \times Q} \mid L(i, i+1) > \mathfrak{T}\} \quad (2.4)$$

$$L(i, i+1) = \frac{\left[ \frac{v_i + v_{i+1}}{2} + \left( \frac{m_i - m_{i+1}}{2} \right)^2 \right]^2}{v_i v_{i+1}} \quad (2.5)$$

where  $L(i, i + 1)$  is the likelihood ratio,  $v$  and  $m$  are the mean and the variance of intensity values for frame  $i$ ,  $P \times Q$  denote the number of regions in the frame and  $T$  is user-defined threshold.

### Color Histogram Comparison

Color histograms have been widely used as a feature for STD algorithms, especially for detecting hard cuts. This is due to the good discriminating capabilities of colour histograms.

**Premise 2** *Frames of the same shot should have similar color histograms, while frames of different shots should have significantly dissimilar colour histograms.*

STD techniques tends to vary in the selection of colour space, the difference metric and thresholding schemes. The most common metrics for comparing two color histograms are histogram difference, histogram intersection, cosine measure, Yakimovsky likelihood ratio test,  $\chi^2$  test, Kolmogorov-Smirnov test, the  $L_1$ -,  $L_2$ - and  $L_\infty$ - measures (Antani *et al.*, 1998) (Patel and Sethi, 1996). The formulation of some of these metrics are as follows:

**Histogram Difference** The histogram difference between 2 histograms  $\mathcal{H}_1$  and  $\mathcal{H}_2$  is defined as the sum of absolute bin-wise difference.

$$\mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \sum_1^K |\mathcal{H}_1[k] - \mathcal{H}_2[k]| \quad (2.6)$$

**Histogram Intersection** The intersection of two color histogram is measured as.

$$\mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \frac{\sum_1^K \min(\mathcal{H}_1[k], \mathcal{H}_2[k])}{K} \quad \text{or} \quad \mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \sum_1^K \frac{\min(\mathcal{H}_1[k], \mathcal{H}_2[k])}{\max(\mathcal{H}_1[k], \mathcal{H}_2[k])} \quad (2.7)$$

**The Chi-Square test** The  $\chi^2$  test is given by

$$\mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \sum_1^K \frac{|\mathcal{H}_1[k] - \mathcal{H}_2[k]|}{\mathcal{H}_2[k]} \quad \text{or} \quad \mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \sum_1^K \frac{|\mathcal{H}_1[k] - \mathcal{H}_2[k]|}{\mathcal{H}_1[k] + \mathcal{H}_2[k]} \quad (2.8)$$

**Kolmogorov-Smirnov test** This test is based on the cumulative distribution of two data sets, i.e. two histograms.

$$\mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \max \left| \sum_{k=1}^j \mathcal{H}_1[k] - \sum_{k=1}^j \mathcal{H}_2[k] \right| \quad (2.9)$$

**Yakimovsky likelihood ratio test** This test was originally proposed to detect the presence of an edge at the boundary of two regions. It is expressed as

$$\mathcal{D}(\mathcal{H}_1, \mathcal{H}_2) = \frac{v_{\mathcal{H}_p}^2}{v_{\mathcal{H}_1} v_{\mathcal{H}_2}} \quad (2.10)$$

where  $v_{\mathcal{H}_1}$  and  $v_{\mathcal{H}_2}$  are the individual variances of the histogram while  $v_{\mathcal{H}_p}$  is the variance of the histogram generated from the pooled data.

Zhang *et al.* (1993) proposed a method called *twin-comparison* to detect gradual transitions using color the histogram difference. This method requires two cut-off thresholds. In the first stage, a higher threshold  $T_h$  is used to detect hard cuts. In the next stage, a lower threshold  $T_l$  is used on the remaining frames. A frame that differs from the previous frame by an amount above this threshold is declared as a potential start of a gradual transition. This frame is then compared to the subsequent frames to get the accumulated difference. During a gradual transition, this accumulated value will gradually increase. The end frame of a gradual transition is detected when the difference between consecutive frames drops below  $T_l$  and the accumulated value has increased to a value that exceeds  $T_h$ . If the difference between consecutive frames drops below  $T_l$  before the accumulated difference exceeds  $T_h$  then the  $f_s$  is dropped and the search starts again for other gradual transitions. Otherwise, a gradual transition is declared. The number of frames between  $f_s$  and  $f_e$  is the duration of the gradual transition. The algorithm does not differentiate the type of the detected gradual transition. In addition, changes due to camera movements such as panning and zooming tend to induce successive difference values of the same order as those of the gradual transitions. Therefore, some global motion estimation needs to be incorporated with the algorithm in order to produce reasonable results.

### Motion-Based Approaches

Motion based approaches to shot transition detection relies on the following premise:

**Premise 3** *While the motion within a shot is smooth, the motion between frames where shot transitions occur tends to be abruptive.*

The algorithm by Shararay (1995) uses block matching and motion estimation to detect shot changes and gradual transitions. Each chosen frame is divided into 12 non-overlapping blocks. Each block is matched to a block in the next chosen frame within  $30 \times 30$  pixel neighborhood. The corresponding motion vector and best best correlation are computed. The correlation values are sorted in ascending order. A similarity measure is computed by taking the average of the first  $s$  values from the sorted list where  $s$  is the number of blocks scaled by a user-specific matching percentage. Shot changes are detected by local maxima in the similarity measure. Gradual increases in frame difference are used to detect gradual transitions.

### Production Model-Based Approaches

Much research in gradual transition detection has been carried out by analyzing the production models of these effects. These models are presented in Chapter 3.

Hampapur *et al.* (1994), Gu *et al.* (1997) and Song *et al.* (1998) analyze the effect of the production model of fades and dissolves on individual pixels. Gu *et al.* (1997) calculate a chromatic image from a pair of consecutive images. Its value at each pixel is the change in intensity between the two images divided by the intensity of the later image. Ideally, the chromatic image should be uniform and non-zero during a fade. Gu *et al.* (1997) exploits the fact that during a fade or dissolve, the first partial derivative of images will be within a range, while Song *et al.* (1998) go further by noting that the first partial derivative of image with respect to time will be constant, and therefore its second derivative will be zero during a fade or dissolve. Allowing for noise, they test for the second image to be less than a fraction of the first partial derivatives. The frames for which this test is satisfied are part of the transitions.

Instead of exploiting the intensity changes of individual pixels, Alattar (1993), Meng *et al.* (1995), Alattar (1997) and Lienhart (1999) investigate the effect of the production model on frame luminance mean and variance. During an ideal dissolve or fade, mean changes in a linear manner, while variance has a parabolic or half parabolic shape. Alattar (1993) detects dissolves by first recording all negative spikes in the second order derivative of frame variance and ensuring the luminance means within a dissolve region do not change sign. Meng *et al.* (1995) record all successive peaks and the valley between them on the variance curve as an indication of a parabolic region caused by a dissolve. Conditions are also applied to ensure the dissolve is wide enough and the valley is depth enough. Alattar (1997) detects fades by recording all negative spikes in the second derivative of variance curve and ensuring that the first derivative of mean curve is relatively constant next to a negative spike. Lienhart (1999) detects fades by fitting a regression line on the frame standard deviation curve. In Chapter 3, we present further extension of work done in this direction, i.e. utilizing mean and variance characteristics.

While mathematical models, presented in Chapter 3, for fades and dissolves are explicit and can be easily formulated, it is not easy to model wipes mathematically due to the variety in wipe patterns. The only research found in the literature proposing a production model for the wipe region is by Alattar (1998). Based on the proposed models, statistical characteristics of the frames in the wipe region, especially the linear change in the means and variances of frames, is derived and used for detecting wipes.

### Edge-Based Approaches

Zabih *et al.* (1999) propose a technique for detecting and classifying shot transitions based on edge images. Their technique is based on the following premise.

**Premise 4** *During a shot transition, the patterns in the appearance of new edges and disappearance of old edges differs from that during a shot transition. In addition, these patterns are different for different types of editing effects.*

Let  $f_i^e$  denote the edge image of frame  $f_i$ . Let  $\pi_{in}$  denote the fraction of edge pixels in  $f_{i+1}^e$  which are more than a fixed distance from the closest edge pixel in  $f_i^e$ , i.e.  $\pi_{in}$  measures the proportion of *entering* edge pixels. Similarly, let  $\pi_{out}$  denote the fraction of edge pixels in  $f_i^e$  which are more than  $r$  pixels away from the closest edge pixel in  $f_{i+1}^e$ , i.e.  $\pi_{out}$  measures the proportion of *exiting* edge pixels.  $\pi = \max(\pi_{in}, \pi_{out})$  is the basic similarity measure. The detection and classification of shot transitions is as follows. A cut will lead to a single isolated high value of  $\pi$ , while gradual transitions will lead to an interval where  $\pi$  is elevated. A fade-in will have  $\pi_{in}$  much higher than  $\pi_{out}$ , since there will be many entering edge pixels, but few exiting edge pixels. Similarly,  $\pi_{out}$  will have a value much higher than  $\pi_{in}$  during a fade-out, since there will be many exiting edge pixels, but few entering edge pixels. On the other hand, a dissolve consists of an overlapping fade-in and fade-out. Therefore,  $\pi_{in}$  will be greater during the first half of the dissolve and it will be smaller during the second half of dissolve. Wipes can be identified by the distribution of entering and exiting edge pixels. A global motion computation is used to reduce the effect of camera and object motion.

Rather than counting the number of entering and exiting edge pixels, Lienhart (1999) proposes a technique called edge-based contrast which originates in following premise.

**Premise 5** *There is loss of contrast and sharpness of the images during a dissolves that generally reaches a maximum in the middle of the dissolve. Hence, if a feature can be computed to capture and emphasize this loss of contrast and sharpness, dissolve detection is enabled.*

After an edge map is computed, e.g. by the Canny edge detector, Lienhart (1999) uses two thresholds to detect strong edges (i.e. high intensity) and weak edges (low intensity). The edge-based contrast feature is defined as:

$$f_i^{ec} = 1 + \frac{s - w - 1}{s + w + 1}$$

where  $s$  and  $w$  is the sum of strengths of strong edges and week edges, respectively. Dissolves are detected by looking for local minima, surrounded by steep flanks in the  $f_i^{ec}$  curve.

### Temporal Slice Analysis

Recently, some STD algorithms based on temporal slice analysis have been proposed. A slice is an 1D image taken from a frame, while a spatio-temporal image is a collection of slices in the sequence at the same position. Vertical, horizontal primary diagonal and subprimary diagonal slices are three frequently used in research. The fundamental idea behind the approaches using temporal-slice images is as follows (Kim *et al.*, 1999).

**Premise 6** *In a temporal-slice image, pixels along a vertical line are sampled from a slice of the same image. The vertical lines of a temporal-slice image will have similar visual features if they come from the same shot. The lines will be different if they come from different shots. Thus, if a shot change occurs, its boundaries will become apparent, as the visual features of the vertical lines will change.*

Table 2.1 shows the patterns of the temporal-slice image for different shot transition types. In general, a camera cut results in vertical boundary lines; a wipe results in a slanted or curved boundary lines; while a dissolve connects two regions slowly and does not have a clear boundary (Ngo *et al.*, 1999). The task of detecting shot transitions is therefore equivalent to the task of segmenting the image into regions.

Slices	Cut	Dissolve	Wipe			
			l-to-r	r-to-l	t-to-b	b-to-t
H						
V						
D						

Table 2.1: Illusion of edit effects on temporal slice images

Ngo *et al.* (1999) proposed a Markov based image segmentation algorithm to locate the color texture discontinuities at region boundaries to detect wipes and cuts. Dissolves are detected by checking that the luminance means of slices are approximately constant and while their variances forms a concave upward parabolic shape. They also utilize MPEG motion vectors and DCT coefficients similar to techniques discussed in sections 2.2.3 and 2.2.3 to eliminate false positives in detection hard cuts. Kim *et al.* (1999) first form the derivative image of a temporal-slice image by taking the absolute difference of two adjacent pixels on the same vertical line. Cuts are detected by looking for peaks in the sums of all columns of the derivative image. Wipes are detected by first recording all peaks in the vertical lines of derivative image and then the connectivity of these peaks are checked.

### 2.2.3 Compressed Domain Techniques

Several video compression schemes such as MPEG, DVI and JPEG have been proposed and standardized to efficiently transmit and store video data. For algorithms proposed for the full-pixel domain to be used with compressed data, the original video data first needs to be fully decompressed into a sequence of image frames. However, this would be computationally expensive. Some researchers have developed STD algorithms for compressed data without the full decompression step. These algorithms either try to approximate features used in full-pixel domain techniques (e.g. DC-image based approaches) or utilize the “prediction by similarity” for motion compensation used by MPEG compression scheme (e.g. techniques based on macro block type) or the reflection of similarity between images on its encoded information (e.g. approaches based on bit rate and DCT coefficients).

#### MPEG Video Compression Standard

The Moving Picture Expert Group (MPEG) was formed in 1988 to generate standards for digital video and audio compression (Gall, 1991). The MPEG standard is arguably the most widely accepted international video compression standard. Most STD algorithms working on the compressed video sequence assume the data is encoded in the MPEG format; therefore, we will briefly describe the fundamental components of MPEG compression algorithm.

Figure 2.2: The layered structure of MPEG encoding

The MPEG encoding algorithm relies on two basic techniques for compressing data: (1) block-

based motion compensation to capture temporal redundancy; and (2) DCT domain compression to capture spatial redundancy. Motion compensations are performed using both the predictive and interpolative approaches. A MPEG stream therefore consists of three different types of frames:

- I (intra-coded) frames are encoded without any reference to other frames. Every macro blocks in an I frame is intra-coded using the JPEG algorithm and can be decompressed without referencing to other frames.
- P (predictive) frames are encoded using motion compensation prediction from the last I or P frame. Macro blocks of P frame may have forward references to its preceding I or P frame when similar patterns are found between them. The macro block is intra-coded when a similar image pattern can not be found in the preceding I/P frame.
- B (bidirectional) frames are encoded with references to both preceding and following I/P frames. The macro block in a B frame can be bidirectional prediction coded, forward-prediction coded or backward-prediction coded.

The predictive relationship between I, P and B frames is illustrated in figure 2.3.

Figure 2.3: Predictive Relationship between I, P and B frames

### DC Image

In I frames, the DCT coefficients in each  $8 \times 8$  DCT-block are related to the luminance/chrominance of  $8 \times 8$  pixels in spatial domain. Therefore, DCT coefficients would be used to detect the difference between the luminance/chrominance signal of all pixels within the  $8 \times 8$  block, and therefore can be used to detect hard cuts. The zero frequency term of the DCT coefficient series is known as DC-term. The set of all DC terms in an I frame forms a DC-image. Since the DC term is the average of the luminance/chrominance of all pixels within the  $8 \times 8$  block, the DC-image can be seen as a spatially reduced version of original image. The extraction of DC-images from I frames is straightforward, while more computation is needed to extract DC-images from B/P frames, since the DC terms for B/P frames available in the MPEG sequence

are not the DC terms of the original frames itself due to motion compensation and prediction involved. Yeo and Liu (1995a) and Shen and Delp (1995) proposed different techniques for approximating DC-images in B/P frames, and the Yeo and Liu (1995a) technique has been extensively used.

**Premise 7** *The DC-image retains most of information about the original image, and therefore, most of features in the original image can be approximated using its corresponding DC-image. With regards to hard cut detection, two frames belong to the same shot should have similar DC image, while frames belong to different shots should have significantly different DC images.*

Although the use of DC-Image eliminates the full decoding of compressed video sequences before the STD algorithm is applied, decoding of DCT component is still required. Approaches discussed in next 2 sections can further reduce the processing time by avoiding the decoding of the video together.

### DCT Coefficients

Strictly speaking, approaches using DC-images can be classified into this category, since DC coefficient is the zero order coefficient of DCT transform; however, in this section we only include approaches that utilize higher order coefficients of DCT transform. Like the DC-Image based approaches, DCT coefficients based approaches can be used with any DCT based compression scheme, e.g. MPEG and JPEG.

**Premise 8** *If two pixel blocks are similar then when transformed into the DCT domain, their DCT-coefficients should be similar, while they are significantly dissimilar if two pixel blocks are not similar.*

It can be seen from the above premise that if we have a function  $\mathcal{D}$  to measure the similarity between compression unit blocks (CUB) based on their DCT-coefficients, then template matching discussed earlier can be used to detect a shot change.  $\mathcal{D}(f_m, f_n, i, j)$  used by (?) is a simple normalized absolute difference of two corresponding blocks. If the difference is larger than a threshold, then block  $(i, j)$  is considered to be a changed block. If the number of changed blocs exceeds a certain threshold, a scene change is declared. Since the method is used on I frames in MPEG video sequence, the exact location of scene changes cannot be located with these detection algorithm.

Arman *et al.* (1993) presented a sub-sampling approach to compare two sets of DCT coefficient series. For each compressed frame  $f_i$ ,  $B$  blocks are first chosen *a priori* from  $R$  connected regions

in  $f_i$  and a set of randomly distributed coefficients is selected from each block. Concatenating the sets of coefficients selected from individual blocks in  $R$  give us a vector  $V_i$ . We can use any vector-distance function to measure the similarity between  $V_i$  of frame  $i$  and  $V_j$  of frame  $j$ . The normalized inner product function is used by Arman *et al.* (1993). A scene change is detected if the difference is greater than a threshold.

### Bit Rates

The use of bit rate information for STD has been investigated by Heng *et al.* (1998), Feng *et al.* (1996) and Deardorff *et al.* (1994). Bit rate information is readily obtained from the video stream by using the size of the block, macroblock or slice of each frame.

**Premise 9** *When variable bit rate encoding is used, frames of similar visual content should have the similar bit rates, while frames of dissimilar visual content should have significantly different bit rates.*

The following observations can be made on the effect of scene change on the on bit-rate of different picture types (Feng *et al.*, 1996).

1. I frame: As I frame is coded independent of other frames, the bit rate of an I frame depends on its visual content and visual activity. When the visual content changes, the bit rate changes too. This observation is used by Deardorff *et al.* (1994) to detect shot changes in motion-JPEG-compressed movies, which contain only I frames.
2. P frame: When a scene change occurs between two consecutive P-frames, it is difficult to make forward predictions. Therefore, most of the macroblocks of the P frame will be intra-coded which leads to a significant increase in the bit rate of the current frame.
3. B frame: Most of macroblocks of this frame will depend on the future frame. Since the prediction can only be done mostly in one direction, the coding efficiency of motion compensation will be lower and it will lead to an increase in the bit-rate when compared to the previous B frame.

Since bit rate information is available for each macroblock, slice and the whole frame, a bit rate vector can be built for each frame. A function, e.g. sum of absolute difference, then can be used to measure the similarity between two vectors. Heng *et al.* (1998) proposed a set of three distinct functions to compare two bit rate vectors: *magnitude function*, *distribution function*, and *shape function*. However, this technique can only be applied to I frames.

### Macro-Block Information/Motion Vectors

Since motion vectors reflect the similarity between frames, some previous work has investigated the use of motion vectors in detecting shot changes.

**Premise 10** *In a MPEG coded video sequences, P frames have references to previous I/P frames, and B frames have references to both previous and following I/P frames. Furthermore, the level of referencing depends on the similarity between the referencing frame and referenced frame. Therefore, a shot change would cause an abrupt change in the referencing pattern of B/P frames.*

Let  $\mathcal{R}_f$  and  $\mathcal{R}_b$  are some functions measuring the prevalence of forward reference and backward reference respectively in an encoded frame. The following observations are fundamental to many STD algorithms using information about macro block types (Kuo *et al.*, 1996). We consider three possibilities for the type of frame where shot change occurs.

1. A shot change occurs at an I frame: I frame does not have any backward or forward references. However, some clues can be found in B frames preceding this I frame. These B frames use this I frame for backward reference. However, as they belong to different shots, this frame does not have many blocks similar to preceding frames, therefore,  $\mathcal{R}_b$  of these B frames must be low.
2. A shot change occurs at a P frame: The behaviour of B frames preceding this P frame is similar to the previous case. In addition, the forward reference of this frame, either a P or I frame, belongs to different shot and is not similar to this frame. Therefore,  $\mathcal{R}_f$  of this P frame must be low.
3. A shot change occurs at a B frame: This frame itself will have low  $\mathcal{R}_f$ . If there exists B frames between this frame and the previous I/P frame, their  $\mathcal{R}_b$  must be low with the similar reason as case 2. If there exists B frames between this B frame and the next I/P frame, their  $\mathcal{R}_f$  must be low too with the similar reason as case 1. In addition, if the first non-B frame in the following sequence is a P frame, the  $\mathcal{R}_f$  of this P frame must be low, since it is predicted from a I/P frame of the previous shot.

Based on these observations, different techniques for STD have been developed, and functions  $\mathcal{R}_f$  and  $\mathcal{R}_b$  are different between techniques. Kuo *et al.* (1996) set up a series of masks for identifying where a scene change occurs. Gamaz *et al.* (1998) compare I frames using their DC-Images and a skipping technique was employed to speed up computation time. In addition

to motion information, Meng *et al.* (1995) examine the peaks in the absolute value of the frame variance difference by a local window to detect a shot change at an I-frame. Sugano *et al.* (1998) use temporal subsampling on B frames and spatial subsampling on slices to reduce the computation. Wipes are detected based on the peak in the variance of inner product of forward and backward vectors in a frame, as the authors argued that they are caused by the large variance in either motion direction or motion vector magnitude caused by a wipe. Sugano *et al.* (1998) also observe patterns in backward prediction and forward prediction useful for flash light detection. Sometimes information about macro block type alone cannot confirm the existence of a shot, Kobla *et al.* (1996) deal with this situation by counting the number of DCT coefficients of corresponding blocks that differ in value by more than a threshold. If this count exceeds another threshold then a shot change is confirmed.

#### 2.2.4 Threshold Selection

While finding appropriate features and metrics for comparison of two consecutive frames is no longer significant problem, the problem of interpreting the difference values, i.e actual selection of certain values to be shot-changes, still remain a major challenge in practice (Hanjalic *et al.*, 1997). The proper selection of difference values is usually done by setting thresholds. Zhang *et al.* (1993) propose a statistical approach for determining the threshold, based on measure mean value  $\mu$  and standard deviation  $\delta$  of frame-to-frame differences. The threshold  $T$  is determined as  $T = \mu + \alpha\delta$ . They also suggest that  $\alpha$  should have values between 5 and 6.

The window-approach in which detection takes place at the center value of the window, used in Yeo and Liu (1995b) Ardebilian *et al.* (1997) and Hanjalic *et al.* (1996), which is also investigated in our work in chapter 3. This can further improve thresholding since it is more appropriate to treat a shot change as a local activity. One requirement with the window-approach is that the window size should be set so that it is unlikely that two shots occur within the window. Therefore, the center value in the window must be the largest frame-to-frame difference in the window. Yeo and Liu (1995b) select the threshold based on the second largest value within the window. Ardebilian *et al.* (1997) divide all points within the window into 2 clusters depending on the distance from each point to the largest point and the lowest point. The threshold is then set based on the distance between two clusters. If the distance is below a threshold, the threshold will be set just above the upper cluster, otherwise, it will be set as half of the distance between the clusters. Hanjalic *et al.* (1996) combines sliding-window approach and general statistical models for the frame-to-frame difference curve to detect hard cuts.

## 2.3 Hierarchical Video Content Characterization

Studies have been reported in the literature addressing the problem of automatic video classification. Hidden Markov Models (HMM) have appeared to be the most widely used classification technique for video analysis, since it can capture temporal structure which is inherent in video programs such as newscasts and sports. Features for constructing classification models have come from three major technologies:

- **Image and image sequence analysis** address the relation between visual characteristics and video semantics. Image characteristics such as color distribution and textures are different across video classes, and allow the recognition of significant objects and actions such as faces, program logos and gestures. Furthermore, image sequence analysis enables camera motion estimation, temporal segmentation, and temporal structure modeling.
- **Audio analysis** investigates the semantics within the audio component of a video sequence. Each video class normally possesses unique audio characteristics. By listening to the audio alone, humans would easily distinguish between a soccer and basketball broadcasts as well as between feature films and children cartoons.
- **Text processing** utilizes spoken words spotted from audio signal analysis as well as close captions and texts extracted from image and image sequence analysis to index a video sequence. It also looks at annotations derived from transcripts to support video content characterization.

The classification of video data depends on specific application requirements and data domain. However, it would be very useful for content-based video indexing and retrieval if automatic video classification would resemble the way humans classify video hierarchically into categories. Figure 2.4 shows a sample hierarchical organization of video data. Data units for classification and recognition at higher levels would be entire clips, while that at lower levels would be events or scenes within a single clip.

At the highest hierarchy level, video can be classified into sports, feature films, news, documentaries and so on. Video classification at this level is discussed in Effelsberg *et al.* (1995) in which the authors propose a three-step approach to video classification: (1) syntactic property analysis, (2) film style attribute abstraction, and (3) mapping and recognition. Effelsberg *et al.* (1995) also discuss a set of features for recognizing tennis, car racing, cartoons, commercials and newscasts. However, the authors do not present the final classification results using their approach. There are also efforts in employing only audio components to classify videos into different categories. Hidden Markov Models and Neural Networks are used in (Liu *et al.*, 1997)

Figure 2.4: A sample video classification tree

and (Liu *et al.*, 1998) respectively to classify television programs into five categories, namely commercials, basketball games, football games, news reports and weather forecasts. Recently, Kobla *et al.* (1999) propose the detection of slow motion replay based on the alternation of static and motion frames to classify videos into sports and non-sports. Unfortunately, the common use of high-speed camera in sports video production would undermine their technique.

At the next level of the hierarchy, sports videos, for example, can be classified into different sub-categories such as car racing, soccer, tennis and so on. Ariki *et al.* (1997) propose a technique for classifying TV sports news using DCT coefficients computed from key frames and multiple subspace method. Similarly, Sahouria and Zakhor (1998) use principal component analysis as the basis to recognize hockey, basketball and volleyball video sequences. Based on motion and shot length, Iyengar and Lippman (1998) characterize feature film trailers as action or character. A similar work is also reported in Vasconcelos and Lippman (1997), in which shot length and average shot activity are used to characterize the level of violence of a movie and to classify movie trailers into action, romance/comedy. Colombo *et al.* (1999) propose an approach to indexing commercials. Based on color characteristics, the distribution of lines, and editing effects to calculate the likelihood score that a commercial belongs to one in four semiotic categories {practical, playful, utopic and critical}.

Video classification was also studied at a higher level of detail in which a video sequence can be segmented, and each segment can then be classified according to its semantic content. For example, a baseball telecast can be classified into scenes according to the game specific actions they imply (Kawashima *et al.*, 1998). Ariki and Matsuura (1999) detect and classify news articles into different topics such as politics, economy, science, etc based on telop character recognition, while Zhang *et al.* (1995) combine audio, video and text information to construct a hierarchy of abstraction of the entire news program.

## 2.4 Summary

In this chapter, we have reviewed existing approaches in the literature that address the two problems that our work aims to solve: video segmentation and video classification. Techniques for detecting scene breaks form two major groups. Techniques from the first group operates on features computed from full pixel frames such as colour histograms, template difference, contour information and so for. Techniques from the second group utilizes information available in MPEG compressed video stream such as bit rate, motion vectors, and macro block types to detect the scene discontinuity resulted from a shot transitions. We have then presented current techniques for video classification and given an overview of the methods used. The details of our work will be presented in the remainder of the thesis. In the next chapter, we present a relatively new approach to detecting cuts, fades, and dissolves in a video sequence.

## Chapter 3

# Shot Transition Detection with Accurate Boundary Identification

This chapter describes our approach to shot transition detection. We employ simple features such as color histograms together with mean and variance of luminance. In tackling gradual transitions, we also aim to accurately classify the type of transitions (fade-in, fade-out, and dissolve) and to precisely locate the boundary of transitions. This makes our work distinguishable from early work in scene change detection which focus on identifying the existence of a transition rather than its temporal extent. Wipe detection is still a challenging problem, and techniques proposed in literature tend to produce many false positives. In addition, from the experiments on our large video data set, wipe transitions are very uncommon, accounting for less than 0.5 percent of all transitions. Therefore, we can safely ignore the problem of wipe detection. The layout of this chapter is as follows. In the first three chapters, we describe algorithms for detecting cuts, fades and dissolves. In section 4.4, we compare the performance of our algorithms against other methods. The results show that our algorithms outperform their counterparts. Section 4.5 summarizes the chapter.

### 3.1 Hard Cuts

Color histogram difference has proved to be the most simple, yet effective method for detecting hard cuts. The variety of algorithms differ in the selection of color space to compute the histograms and the function to compare the similarity between two histograms. Here we propose a simple adaptive thresholding technique for detecting peaks in the histogram difference curve.

The color histogram of a frame is computed on its pixel luminance as:

$$f_i^{\mathcal{H}}[k] = |\mathcal{B}_k| \quad , \quad \text{where} \quad \mathcal{B}_k = \{(x, y) \in \mathbb{N}_1^X \times \mathbb{N}_1^Y \mid f_i(x, y) = k\} \quad (3.1)$$

The color histogram difference between two consecutive frames  $f_i$  and  $f_{i+1}$  is then computed as the sum of absolute bin-wise differences:

$$t_i^{\mathcal{H}} = \sum_{i=1}^K (|f_i^{\mathcal{H}}[i] - f_{i+1}^{\mathcal{H}}[i+1]|) \quad , \quad \text{where} \quad K \text{ is the number of bins.} \quad (3.2)$$

Unlike some previous approach using global thresholds on the difference of color histograms, we employ an adaptive thresholding approach. Since scene change is a *local* activity in the temporal domain, often it is more appropriate to set the threshold based on neighborhood values. For each frame transition  $t_i$ , we consider a sliding window of size  $2w + 1$  in  $\mathcal{T}$  that covers frame transitions  $t_{i-w}, t_{i-w+1}, \dots, t_{i+w-1}, t_{i+w}$ . A hard cut between frame  $f_i$  and  $f_{i+1}$  is declared if all followings condition are satisfied.

1.  $t_i^{\mathcal{H}}$  has the maximum value within the window. This means  $t_i \geq t_j, \forall j \in \mathbb{N}_{i-w}^{i+w}$ .
2.  $t_i^{\mathcal{H}}$  is  $\alpha$  times greater than the mean of all  $t_j^{\mathcal{H}}$  within the window excluding  $t_i^{\mathcal{H}}$  itself. In order to guard against the case when  $f_i$  and  $f_{i+1}$  are surrounded by “freeze” frames that results in the mean being approximately zero, in which case it is difficult to select the appropriate threshold, we add a constant  $c$  to both mean and the value of  $t_i^{\mathcal{H}}$ . Thus

$$t_i^{\mathcal{H}} + c \geq \alpha \frac{\sum_{j=l-w, j \neq i}^{l+w} (t_j^{\mathcal{H}} + c)}{2w} \quad (3.3)$$

There are some possible ways to further enhance the algorithm and reduce the computation time. First, we can use a global threshold to determine all potential cut frames  $t_i$ , and sliding windows are applied only to these frames, and all the rest can be ignored. This global threshold should be small allowing all cuts to be detected. In addition, if  $t_i$  is declared as a cut transition, then it is possible to proceed directly to  $t_{i+w+1}$  without checking  $t_{i+1}, t_{i+1+1}, \dots, t_{i+w-1}, t_{i+w}$ , since the color histogram difference of these frames will be less than  $t_i$  and the first criterion is not satisfied. In fact, if we transform the color histogram difference curve into the local mean-ratio space, then the adaptive threshold presented above becomes a fixed threshold, i.e.  $\alpha$ . The value of the curve at position  $i$  is computed as:

$$t_i^{\mathcal{H}^*} = \frac{t_i^{\mathcal{H}} + c}{\frac{\sum_{j=l-w, j \neq i}^{l+w} (t_j^{\mathcal{H}} + c)}{2w}}$$

Figure 3.1 shows the color histogram difference curve with one global threshold together with the curve of our adaptive thresholds. Figure 3.1 shows the transformed color histogram difference for a window of size  $w = 5$  and threshold  $\alpha = 1.2$ . All cut points, whose histogram different is beyond the computed threshold, are marked by little circles. Under close examination, the

(a) Histogram difference ( $t_i^{\mathcal{H}}$ ) (b) Local mean-ratio ( $t_i^{\mathcal{H}^*}$ )

Figure 3.1: Color histograms and its local mean-ratio transform

global threshold causes a false positive at near frame 200, while the adaptive threshold does not. More importantly, figure 3.1 shows other potentials for false positives due to object and camera motion near frame 200, 300, 500 600, and 800 if a global threshold is used, while these artifacts are significantly reduced in figure 3.1. In figure 3.1 all non-cut frames have value approximate 1 or much less than 1 if they are within the window of frames that contain cuts. In brief, two important aspects of our algorithms are the use of local mean to reduce the effect of motion and the addition of a constant to ease the selection of an appropriate threshold. The proposed hard cut detection algorithm is detailed below:

---

**Algorithm 1** Cut Detection:  $c \leftarrow \text{getCuts}(\text{filename})$

---

**Require:** filename is a valid MPEG-1 filename

```

 $\mathcal{V} \leftarrow \text{initVideo}(\text{filename})$     {initialize the decoder}
 $f_1 \leftarrow \mathcal{V}.\text{getNextFrame}()$     {get the next frame in the video sequence}
 $f_1.\text{compHis}()$     {compute the color histogram}
 $f_2 \leftarrow \mathcal{V}.\text{getNextFrame}()$ 
 $i \leftarrow 1$ 
while  $f_{i+1} \neq \text{NULL}$  do
     $f_{i+1}.\text{compHis}()$     {compute the color histogram}
     $t_i^{\mathcal{H}} \leftarrow f_i.\text{hisDiff}(f_{i+1})$     {compute the color histogram difference}
end while
 $i \leftarrow 1$ 
while  $t_i^{\mathcal{H}}.\text{isLast}(i) \neq \text{NULL}$  do
    if  $t_i^{\mathcal{H}} > \mathfrak{T}_1$  then
         $ws \leftarrow \text{windStart}(i, w, \mathcal{V}.\text{size})$     {compute the start of the window}
         $we \leftarrow \text{windEnd}(i, w, \mathcal{V}.\text{size})$     {compute the end of the window}
         $man \leftarrow t_i^{\mathcal{H}}.\text{getMax}(ws, we)$     {compute the maximum(index) value within the window}
         $\mu \leftarrow t_i^{\mathcal{H}}.\text{getMean}(ws, we, i)$     {compute the mean of all points excluding the center}
        if  $\min \leq t_i^{\mathcal{H}}$  and  $t_i^{\mathcal{H}} \leq \mathfrak{T}_2\mu$  then
             $c.\text{insert}(i)$     {constraints 1 and 2 satisfied, insert this cut into the list}
             $i \leftarrow i + w$     {skip some frames}
            continue
        end if
    end if
     $i \leftarrow i + 1$ 
end while
return  $c$ 

```

---

## 3.2 Fades

For detecting gradual transitions, we start with the production model of these effects. The discussion presented here for fades and in the next section for dissolves, assumes the ideal cases, while, in reality noise, motion and manual operation of these effects would distort the model. However, ideal models still share many characteristics with real effects, and they provide numerous clues for detecting these transitions. Our mathematical modeling of fades is similar to Alattar (1997).

Consider a discrete linear fade-out sequence starting at frame  $f_s$  and ending at frame  $f_e$ . Let  $\bar{f}_i$  denote the frame corresponding to  $f_i$  in the video sequence if the fade operation did not occur between  $f_s$  and  $f_e$ . We are interested only in shots in which fades occur. The fade-out sequence to color  $\mathcal{C}$  can be modeled using the following equation:

$$f_i(x, y) \stackrel{\text{out}}{=} \mathcal{E}_{(s,e)}^{\text{fade-out}}(\bar{f}_i(x, y)) = \begin{cases} \bar{f}_i(x, y), & \text{if } t \leq s; \\ \left(1 - \frac{t-s}{e-s}\right) \bar{f}_i(x, y) + \frac{t-s}{e-s} \mathcal{C}, & \text{if } s < t \leq e. \end{cases} \quad (3.4)$$

where  $x, y$  and  $t$  are variables that represent the horizontal, vertical and temporal dimensions of a video sequence, respectively,  $\mathcal{E}_{(s,e)}^{\text{fade-out}}$  represents the the fade-out operation from frame number  $s$  to  $e$  and the length of fade sequence is  $(e - s)$ . A similar model can also be presented for a fade-in sequence from color  $\mathcal{C}$ :

$$f_i(x, y) \stackrel{\text{in}}{=} \mathcal{E}_{(s,e)}^{\text{fade-in}}(\bar{f}_i(x, y)) = \begin{cases} \frac{i-s}{e-s} \bar{f}_i(x, y) + \left(1 - \frac{i-s}{e-s}\right) \mathcal{C}, & \text{if } s \leq t < e; \\ \bar{f}_i(x, y), & \text{if } t \leq s. \end{cases} \quad (3.5)$$

If we assume that frames  $\bar{f}_i$  remain static, and it has mean  $\bar{\mu}$  and variance  $\bar{v}$  during the fade duration, the mean and variance of  $f_i$  for a fade-out sequence can be computed as:

$$f_i^\mu \stackrel{\text{out}}{=} \begin{cases} \bar{\mu}, & \text{if } t \leq s; \\ \left(1 - \frac{i-s}{e-s}\right) \bar{\mu} + \frac{i-s}{e-s} \mathcal{C}, & \text{if } s < i \leq e, \end{cases} \quad \text{and} \quad f_i^v \stackrel{\text{out}}{=} \begin{cases} \bar{v}, & \text{if } t \leq s; \\ \left(1 - \frac{i-s}{e-s}\right)^2 \bar{v}, & \text{if } s < i \leq e. \end{cases} \quad (3.6)$$

Similarly, for the case of a fade-in, we have:

$$f_i^\mu \stackrel{\text{in}}{=} \begin{cases} \frac{i-s}{e-s} \bar{\mu} + \left(1 - \frac{i-s}{e-s}\right) \mathcal{C}, & \text{if } s \leq i < e; \\ \bar{\mu}(x, y), & \text{if } t \leq s, \end{cases} \quad \text{and} \quad f_i^v \stackrel{\text{in}}{=} \begin{cases} \left(\frac{i-s}{e-s}\right)^2 \bar{v}, & \text{if } s \leq i < e; \\ \bar{v}(x, y), & \text{if } t \leq s. \end{cases} \quad (3.7)$$

As can be seen from the above equations and figure 3.2, mean values move linearly towards  $\mathcal{C}$ , while variance curves of fade-out and fade-in sequences have a half-parabolic shape independent of color  $\mathcal{C}$ . This half-parabolic shape is clear in the figure 3.2 which shows the variance curve for a video segment with 2 pairs of fade-in/fade-out. It can also be shown that  $t_i^\mu$  does not

change during the fade duration while  $t_i^v$  changes in a linear fashion. In the case of a fade-out, we have:

$$t_i^{\mu \text{ out}} \equiv \begin{cases} \frac{\mathcal{C} - \bar{\mu}}{e - s}, & \text{if } s \leq i < e; \\ 0, & \text{if } i < s, \end{cases} \quad \text{and} \quad t_i^{v \text{ out}} \equiv \begin{cases} -\frac{2(e-i)-1}{(e-s)^2} \bar{v}, & \text{if } s \leq i < e; \\ 0, & \text{if } i < s. \end{cases} \quad (3.8)$$

Similar equations can be used for a fade-in sequence.

$$t_i^{\mu \text{ in}} \equiv \begin{cases} \frac{\bar{\mu} - \mathcal{C}}{e - s}, & \text{if } s \leq i < e; \\ 0, & \text{if } i \geq e, \end{cases} \quad \text{and} \quad t_i^{v \text{ in}} \equiv \begin{cases} \frac{2(i-s)+1}{(e-s)^2} \bar{v}, & \text{if } s \leq i < e; \\ 0, & \text{if } i \geq e. \end{cases} \quad (3.9)$$

Therefore, if we take the first order difference of  $t_i^v$  curve of a fade-out, ie. the second order difference of  $f_i^v$  curve, we have:

$$t_i^{v \text{ out}} \equiv t_{i+1}^v - t_i^v \equiv \begin{cases} \frac{2}{(e-s)^2} \bar{v}, & \text{if } s \leq i < e-1; \\ -\frac{2(e-s)-1}{(e-s)^2} \bar{v}, & \text{if } i = e-1; \\ 0, & \text{if } i \leq s-1. \end{cases} \quad (3.10)$$

Similarly, for a fade in, the first order difference of  $t_i^v$  curve can be presented as:

$$t_i^{v \text{ in}} \equiv t_{i+1}^v - t_i^v \equiv \begin{cases} \frac{2}{(e-s)^2} \bar{v}, & \text{if } s \leq i < e-1; \\ -\frac{2(e-s)-1}{(e-s)^2} \bar{v}, & \text{if } i = e-1; \\ 0, & \text{if } i \geq e. \end{cases} \quad (3.11)$$

From mathematical equations presented above and observations of real fades, following remarks are crucial to our fade detection algorithm:

1. Fade-in and fade-out often occur together as a *fade group*. More specifically, a fade group starts with a shot fading out to color  $\mathcal{C}$  that is then followed by a sequence of monochrome frames of color  $\mathcal{C}$ , and it ends with a shot fading in from color  $\mathcal{C}$ . Fade groups formed this way are often referred to later in our thesis as a single fade. Monochrome frames are a very good clue for recalling all potential fades, and they are used in our algorithm as the first step in recognizing the existence of a fade. In a quick fade, the monochrome sequence may last only one frame while in a slower fade it would last up to 100 frames. However, we can apply a smaller constraint (e.g. 2 sec.) on the length of fade-in and fade-out components.
2. It can be seen from equation 3.8 mean difference  $t_i$  remains relatively constant and does not change its sign during a fade-out or a fade-in. Indeed, Figure 3.2 confirms the second statement. Since in real videos, this mean feature would be distorted by motion, some smoothing operation needs be applied to the mean difference curve before examining the constancy of its sign within a potential fade region.

(a) Mean Curve (b) Variance Curve

Figure 3.2: Mean and variance curve during a fade

(a) The first order difference of mean Curve (b) The second order difference of variance Curve

Figure 3.3: The first derivative of mean and the second derivative of variance during a fade

3. Similarly, figure 3.2 confirms that depending on whether it is a fade-in or a fade-out, the variance of fading frames will increase or decrease rapidly. This establishes another constraint for the existence of fade-in for our algorithm.
4. Equation 3.10 and 3.11 reveal that a large negative spike appear near the start of a fade-out and near the end of a fade-in. While Alattar (1997) uses only these negative spikes for detecting dissolves, we observe that motion would cause such spikes. It can be seen from figure 3.2 that two relatively large negative spikes are actually present at the end of the fade-in near frame 420, and only the second spike corresponds to the real boundary. This suggests to us that for robustness we should search for all spikes near a monochrome sequence until either criterion 2 or 3 is not satisfied.
5. We also constrain the variance of the starting frame of a fade-out and the ending frame of a fade-in to be above a threshold to eliminate false positives caused by dark scenes.

The fade detection algorithm is detailed in Algorithm 2, and incorporates the essential features of the above remarks.

---

**Algorithm 2** Fade Detection:  $fades = detectFades(filename)$ 


---

**Require:** filename is a valid MPEG-1 filename

```

 $\mathcal{V} \leftarrow \text{initVideo}(filename)$ 
for  $i = 1$  to  $\mathcal{V}.size$  do
   $f_i^\mu \leftarrow \mathcal{V}.getMean(i)$ 
   $f_i^v \leftarrow \mathcal{V}.getVariance(i)$ 
  if  $i \leq 2$  then
     $t_{i-1}^\mu = f_i^\mu - f_{i-1}^\mu$ 
     $t_{i-1}^v = f_i^v - f_{i-1}^v$ 
  end if
  if  $i \leq 3$  then
     $t'_{i-2} = t_{i-1}^v - t_{i-2}^v$ 
  end if
end for
 $ml \leftarrow f^v.getMonoSequences()$  {get all monochrome sequences}
while  $ml.isLast() \neq \text{NULL}$  do
   $m \leftarrow ml.getCurrItem()$  {get current monochrome sequence from the list}
   $e \leftarrow m.start$ 
   $isFadeOut \leftarrow \text{FALSE}$ 
  repeat
     $s \leftarrow t'^v.getNextClosestPreNegSpike(m.start)$  {get the next closest spike on the left}
    for  $i = s$  to  $e - 1$  do
      if  $f_i^\mu f_{i+1}^\mu > \mathfrak{T}_3$  and  $t_i^v > \mathfrak{T}_2$  and  $(e - s) < threshold_3$  then
         $isFadeOut \leftarrow \text{TRUE}$  {constraints satisfied, a potential fade start}
         $fs \leftarrow s$ 
      else
        break {constraints not satisfied, look for the next spike}
      end if
    end for
  until  $e - s \leq \mathfrak{T}_3$ 
  if  $isFadeOut = \text{TRUE}$  and  $f_{fs}^v > \mathfrak{T}_4$  then
     $fo.insert(fs, e)$  {not start with a dark frame, insert into the list of fade-outs}
  end if
   $s \leftarrow m.end$ 
   $isFadeIn \leftarrow \text{FALSE}$ 
  repeat
     $e \leftarrow t'^v.getNextClosestSucNegSpike(m.end)$  {get the next closest spike on the right}
    for  $i = s$  to  $e - 1$  do
      if  $f_i^\mu f_{i+1}^\mu > \mathfrak{T}_3$  and  $t_i^v > \mathfrak{T}_2$  and  $(e - s) < \mathfrak{T}_3$  then
         $isFadeIn \leftarrow \text{TRUE}$ 
         $fe \leftarrow e$  {constraints satisfied, a potential fade end}
      else
        break
      end if
    end for
  until  $e - s \leq \mathfrak{T}_3$ 
  if  $isFadeIn = \text{TRUE}$  and  $f_{fe}^v > \mathfrak{T}_4$  then
     $fi.insert(s, fe)$  {not start with a dark frame, insert into the list of fade-ins}
  end if
   $m.moveToNext()$  {move to the next monochrome sequence}
end while
 $fades \leftarrow \text{joinFadeGroup}(fo, fi, ml)$  {combine fade-ins and fade-outs to form fade groups}
return  $fades$ 

```

---

### 3.3 Dissolves

Similar to fades, our approach to dissolve detection is based on the production model of an ideal dissolve. As mentioned earlier, a fade can be considered as a special dissolve where one shot is composed of monochrome frames; therefore, some mathematical equations presented below are similar to those presented in the previous section.

Consider a dissolve starting at frame  $s$  and ending at frame  $e$ . Let  $\bar{f}_1$  denote frames of the video sequence if the dissolve is replaced by a cut between frame  $e$  and  $e + 1$ . Similarly,  $\bar{\bar{f}}$  denotes frames of the video sequence if the dissolve is replaced by a cut between frame  $s$  and  $s + 1$ . Basically this mean that the dissolve is produced by blending pure shot frames of  $\bar{f}$  and  $\bar{\bar{f}}$ . We only consider the portion of video sequence containing two shots producing the dissolves. We assume these two shots are static during the dissolve duration with means  $\bar{\mu}_1, \bar{\mu}_2$  and variances  $\bar{v}_1, \bar{v}_2$ , respectively. The production model for a dissolve can be presented as:

$$f_i(x, y) = \begin{cases} \bar{f}_i(x, y), & \text{if } i < s; \\ \left(1 - \frac{i-s}{e-s}\right) \bar{f}_i(x, y) + \frac{i-s}{e-s} \bar{\bar{f}}_i(x, y), & \text{if } s \leq i \leq e; \\ \bar{\bar{f}}_i(x, y), & \text{if } i > e. \end{cases} \quad (3.12)$$

where  $x, y$  and  $t$  are variables that represent the horizontal, vertical and temporal dimensions of a video sequence, respectively. The mean and variance of frames within the dissolve sequence then can be presented as:

$$f_i^\mu = \begin{cases} \bar{\mu}_1, & \text{if } s < i; \\ \left(1 - \frac{i-s}{e-s}\right) \bar{\mu}_1 + \frac{i-s}{e-s} \bar{\mu}_2, & \text{if } s \leq i \leq e; \\ \bar{\mu}_2, & \text{if } i > e, \end{cases} \quad (3.13)$$

and

$$f_i^v = \begin{cases} \bar{v}_1, & \text{if } s < i; \\ \left(1 - \frac{i-s}{e-s}\right)^2 \bar{v}_1 + \left(\frac{i-s}{e-s}\right)^2 \bar{v}_2, & \text{if } s \leq i \leq e; \\ \bar{v}_2, & \text{if } i > e. \end{cases} \quad (3.14)$$

It is clear from equation 3.13 and figure 3.3 that mean curve changed a in linear fashion during a dissolve transition, while equation 3.14 and figure 3.3 indicate that the variance curve has a parabolic shape. This means the first order difference of mean curve should be constant during a dissolve, while that of variance curve should change in a linear fashion. Indeed, taking the first order difference of mean curve and variance curve yields

$$t_i^\mu = \begin{cases} 0, & \text{if } i < s; \\ \frac{\bar{\mu}_2 - \bar{\mu}_1}{e-s}, & \text{if } s \leq i < e; \\ 0, & \text{if } i \leq e, \end{cases} \quad (3.15)$$

and

$$t_i^v = \begin{cases} 0, & \text{if } i < s; \\ \frac{-(2(e-i)-1)\bar{v}_1 + (2(i-s)+1)\bar{v}_2}{(e-s)^2}, & \text{if } s \leq i < e; \\ 0, & \text{if } i \leq e. \end{cases} \quad (3.16)$$

If we take the second order difference of variance curve  $f_i^v$ , two large negative spikes should appear at the start and end of the dissolve similar to the case of fade transitions discussed in the previous section. In fact, this feature is the basis for dissolve detection approach proposed by Alattar (1993). However, our observation on real dissolves suggests that these negative spikes are not obvious during a dissolve compared to fades due to noise and motion. Therefore, we ignore these negative spikes in our algorithms. Instead, we look for other clues that can signal the existence of a dissolve, and various constraints to eliminate false positives. We assume that shots making up a dissolve have luminance variance of at least  $\mathfrak{T}_v$  and that the duration of a dissolve never exceeds  $\mathfrak{T}_l$  frames ( $\mathfrak{T}_l$  depends on frame rate). The first assumption would result in misses, since dissolves to near monochrome frames do exist, although they are uncommon. The second assumption is very reasonable, since it is unlikely to have a dissolve lasting longer than 2 seconds. Based on these assumptions, the following observations form the basis for our dissolve detection algorithm.

1. As it can be seen from Eq.3.16, the first order difference  $t_i^v$  of the variance curve changes linearly from a negative value of  $-\frac{2(e-s)-1}{(e-s)^2}\bar{v}_1$  ( $< -\frac{2\mathfrak{T}_v-1}{\mathfrak{T}_l}$ ) at frame  $s$  to a positive value of  $\frac{2(e-s)-1}{(e-s)^2}$  ( $> \frac{2\mathfrak{T}_v-1}{\mathfrak{T}_l}$ ) at frame  $e-1$ . Therefore, the existence of all dissolves can be triggered by all zero crossing sequences in the  $t_i^v$  curve whose start value is below a negative threshold, which then continuously increases, and then the end value is above a positive threshold. Figure 3.3 illustrates clearly this property. In the actual implementation, to reduce the effect of noise and motion we smooth out the curve before searching for these zero crossing sequences.
2. Due to the smoothing operation, the position of the negative and positive peaks of the  $t_i^v$  curve caused by a dissolve is no longer coincident with its actual position ( $s$  and  $e-1$ ) in the ideal case. We can adjust these positions by moving the position of the negative peak backward until the value of  $t_i^v$  increases beyond a negative threshold. Similarly, the position of the positive peak is moved forward until the value of  $t_i^v$  drops below a positive threshold.
3. As mentioned previously, the variance curve  $f_i^v$  has a parabolic shape during a dissolve. It obtains the minimum value of  $\frac{\bar{v}_1\bar{v}_2}{\bar{v}_1+\bar{v}_2}$  at frame number  $\eta = \frac{e\bar{v}_1+s\bar{v}_2}{\bar{v}_1+\bar{v}_2}$  (we ignore the fact that frame number must be an integer). From this, we have:

$$f_s^v - f_\eta^v = \frac{\bar{v}_1^2}{\bar{v}_1 + \bar{v}_2}, \quad \text{and} \quad f_e^v - f_\eta^v = \frac{\bar{v}_2^2}{\bar{v}_1 + \bar{v}_2} \quad (3.17)$$

(a) Mean curve (b) Variance curve

Figure 3.4: Mean and variance curve during a dissolve

(a) The first order difference of mean curve (b) The first order difference of variance curve

Figure 3.5: The first order difference of mean and variance curve during a dissolve

Fig.3.3 simulates the the plotting of  $\frac{\bar{v}_1^2}{\bar{v}_1 + \bar{v}_2}$  against  $\bar{v}_1$ . Similarly, Fig.3.3 simulates the plotting of  $\frac{\bar{v}_2^2}{\bar{v}_1 + \bar{v}_2}$  against  $\bar{v}_2$ . It can be seen from these two figures that all points lie below the line  $y = x/4 - 200$ ; therefore, the difference between the start frame and the middle frame of a dissolve should be greater than  $\frac{\bar{v}_1}{4} - 200$ . The same condition applies to the difference between the end frame and middle frame of a dissolve. In addition, we have:

$$f_s^v + f_e^v - 2f_\eta^v = \frac{\bar{v}_1^2 + \bar{v}_2^2}{\bar{v}_1 + \bar{v}_2} > \frac{\bar{v}_1 + \bar{v}_2}{2} \quad (3.18)$$

By now we already have a set of constraints on the shape of the parabolic curve to eliminate false positives caused by motion. However, these conditions only guide us to set appropriate thresholds. In order to cope with the effects of noise and motion, in the implementation we use lower thresholds. The overall algorithm for detecting dissolves is presented in Algorithm 3.

(a) Plot of  $\frac{v_1^2}{v_1+v_2}$  against  $v_1$

(b) Plot of  $\frac{v_2^2}{v_1+v_2}$  against  $v_2$

Figure 3.6: Hints for threshold selection

---

**Algorithm 3** Dissolve Detection: `dissolves = getDissolves(filename)`

---

**Require:** filename is a valid MPEG-1 filename

```

 $\mathcal{V} \leftarrow \text{initVideo}(\text{filename})$     {initialize the decoder}
 $f_1^\mu \leftarrow \mathcal{V}.\text{getMean}(1)$     {get mean of the first frame}
 $f_1^v \leftarrow \mathcal{V}.\text{getVariance}(1)$     {get variance of the first frame}
for  $i = 1$  to  $\mathcal{V}.\text{size} - 1$  do
     $f_{i+1}^\mu \leftarrow \mathcal{V}.\text{getMean}(i)$ 
     $f_{i+1}^v \leftarrow \mathcal{V}.\text{getVariance}(i)$ 
     $t_i^\mu \leftarrow f_{i+1}^\mu - f_i^\mu$     {get mean difference}
     $t_i^v \leftarrow f_{i+1}^v - f_i^v$     {get variance difference}
end for
 $st^\mu \leftarrow t^\mu.\text{smooth}()$     {smooth out the mean difference curve}
 $st^v \leftarrow t^v.\text{smooth}()$     {smooth out the variance difference curve}
 $dl \leftarrow st^\mu.\text{getZeroCrossings}()$     {search all zero-crossing sequences}
while  $dl.\text{isLast}() \neq \text{NULL}$  do
     $d \leftarrow dl.\text{getCurrItem}()$     {examine the current zero-crossing sequence}
     $d.\text{adjustEnds}()$     {adjust its ends shifted by smoothing operation, remark 2}
     $m \leftarrow f^\mu.\text{getMin}(d.\text{start}, d.\text{end})$     {find the middle point of the dissolve}
    if  $(f_{d.\text{start}}^v - f_m^v < \tilde{\mathfrak{X}}_1)$  or  $(f_{d.\text{end}}^v - f_m^v < \tilde{\mathfrak{X}}_1)$  or  $(f_{d.\text{start}}^v - 2f_m^v + f_{d.\text{end}}^v < \tilde{\mathfrak{X}}_2)$  then
         $dl.\text{remove}(d)$     {some constraint not satisfied, remove it}
    else
        for  $j = d.\text{start}$  to  $d.\text{end} - 1$  do
            if  $f_j^\mu f_{j+1}^\mu < -\mathfrak{X}_3$  then
                 $dl.\text{remove}(d)$     {go to the next zero-crossing sequence in the list}
                break
            end if
        end for
    end if
     $dl.\text{moveToNext}()$ 
end while
return  $dl$ 

```

---

### 3.4 Eliminating False Positives Using Color Histograms

Excluding complex graphics transition effects, changes in lighting, noise, object and camera motion are major sources for false detections by all shot transition detection algorithms in literature. While those caused by long and fast camera operation can be eliminated by performing “Motion Transition Removal Test” as described in Kobla *et al.* (1999) and Zhang *et al.* (1993), we suggest a simple method based on color histogram difference for eliminating other kind of false positives. The premise is simple: frames belonging to the same shot should be similar if there is no transition in scene space, i.e. camera movements. After the detection of cuts, fades and dissolves are performed, for each declared transition, we examine the shot preceding the transition and the shot succeeding it. This transition is considered a false positive if the difference between an arbitrary frame from the first shot and some arbitrary frame from the second shot is less than an empirically determined threshold. This technique can effectively prevent common effects such as flash lights, close-up objects moving in front of the camera, key-in, and other momentary noise (see Appendix B). For computation efficiency, quantized color histograms can be stored in disk, and therefore the second decoding of video is avoided. The overall algorithm for this verification process is presented below.

---

**Algorithm 4** Transition verification based on color histogram:  $l = \text{verifyTrans}(\text{filename})$

---

**Require:** filename is a valid MPEG-1 filename  
 $\mathcal{V} \leftarrow \text{initVideo}(\text{filename})$  {initialize the decoder}  
 $c \leftarrow \mathcal{V}.\text{detectCuts}()$  {run cut detector}  
 $f \leftarrow \mathcal{V}.\text{detectFades}()$  {run fade detector}  
 $d \leftarrow \mathcal{V}.\text{detectDissolves}()$  {run dissolve detector}  
 $l \leftarrow \text{joinTrans}(c, f, d)$  {merge them into a single transition list}  
**while**  $l.\text{isLast}() \neq \text{NULL}$  **do**  
   $ct \leftarrow l.\text{currTrans}()$   
   $pt \leftarrow l.\text{PrevTrans}()$   
   $nt \leftarrow l.\text{nextTrans}()$   
  **if**  $pt = \text{NULL}$  **then**  
     $m = 1$  {this transition is the first in the list, it has no preceding transition}  
  **else**  
     $m = pt.\text{end}$   
  **end if**  
  **if**  $nt = \text{NULL}$  **then**  
     $n = \mathcal{V}.\text{size}$  {this transition is the last in the list, it has no succeeding transition}  
  **else**  
     $n = nt.\text{start}$   
  **end if**  
  **for**  $i = m$  to  $ct.\text{start}$  **do**  
    **for**  $j = ct.\text{end}$  to  $n$  **do**  
       $h_i \leftarrow \mathcal{V}.\text{getHist}(i)$   
       $h_j \leftarrow \mathcal{V}.\text{getHist}(j)$   
      **if**  $h_i.\text{diff}(h_j) < T$  **then**  
         $l.\text{remove}(ct)$  {histogram difference between 2 frames is low, a false positive declared}  
      **end if**  
    **end for**  
  **end for**  
**end while**  
**return**  $l=l$

---

## 3.5 Experimental Results

### 3.5.1 The Data

This section describes the video data set available for both testing video segmentation algorithms and evaluating the feature set for video genre recognition discussed in the next chapter. We collected around 8 hours of video data from TV programs. To ensure the variety of data, news and commercials from different channels on different days and at different times are used. Some clips were recorded more than 5 years ago. Sports clips come from different sub categories such as soccer, Australian football, rugby, tennis and motors racing. Music clips are extracted from different dance music videos. Analog video sources are digitized into Quicktime format using a *SGI 320* capture card with *Motion JPEG A* compressor. QuickTime videos are then encoded into MPEG format for efficient storage using XingMPEG decoder. Unfortunately, the limitation of the capture card and conversion process resulted in the degradation of video quality for final processing, e.g. motion blurred artifacts and the insertion of extra frames by XingMPEG to make up for the loss of frame rate in capturing phase. This affects our work in the following ways:

- There may be a “transition” frame between two shots joined by a cut. Therefore, we must slightly modify the implementation of our cut detection algorithm to recognize this event.
- Our initial investigation in edge-motion based approach to wipe detection is hindered by difficulties in detecting edges resulting from a wipe transition, since the edge boundary is blurred during the video capture stage.
- It is more difficult to measure the motion discontinuity features in cartoons which is useful video in genre recognition (see chapter 4).
- The accuracy in computing camera parameters and other features for shot transition and video genre recognition decreases (see chapter 4).

While the whole 8-hour data set are used to train and test the genre classification model, only a portion of the set containing a substantial number of fades and dissolves are used to test the proposed gradual transition detection algorithms. While the recall and cover parameters for these algorithms should remain approximately the same, the precision would decrease if the test set had a lower rate of fade and dissolves. Table 3.1 describes the data used for testing shot transition detection algorithms in terms of number of cuts, dissolves, fades and their total duration for each video genre.

Categories	Cuts	Dissolves		Fades	
		number	duration (frm)	number	duration (frm)
Cartoons	334	9	152	17	463
Commercials	336	70	1017	28	761
Music	202	34	147	51	1545
News	239	77	973	5	50
Sports	262	107	852	10	260
Total	1373	297	3141	111	3079

Table 3.1: Ground truth of test data for shot transition detection

### 3.5.2 Performance Parameters

Recall and precision concepts in information retrieval field have been often used in the past to evaluate the performance of cut detectors. Unlike cuts, gradual transitions cover a range of frames; therefore, we need to adapt the concept of recall and precision to include gradual transitions. We consider a declared transition of type  $x$  to be correctly detected if it overlaps with at least one real transition of type  $x$ . The set of transitions we are interested in are {cut, fade, dissolve}. Let  $\Phi$  denote the set of frames covered by high motion graphic scenes and other effects such as wipes, and morphing. We do not consider a declared transition as a false positive if it overlaps with a transition of different type, since in many applications, the detection of a scene change is important while the type of transition is irrelevant. We also do not consider a declared transitions as a false positive if it is caused by special frames of set  $\Phi$ , since none of algorithms to date are designed to deal with special effects. Let  $\bar{\Psi}^x = \{\bar{S}_1^x, \bar{S}_2^x, \dots, \bar{S}_{k_x}^x\}$  denotes the set of all transitions of type  $x$  as declared by the detector, while  $\Psi^x = \{S_1^x, S_2^x, \dots, S_{k_x}^x\}$  denotes the set all real transitions of type  $x$  with  $x \in \{\text{cut, fade, dissolve}\}$ . Recall and precision then can be defined as follows.

**Definition 1** Given the set  $\Psi^x$  of real transitions of type  $x$ , the set  $\Phi$  of special effects frames, the set  $\bar{\Psi}^x$  of transitions of type  $x$  as declared by the detector, the recall level of the detector with respect to video sequence  $\mathcal{V}$  is defined as:

$$\mathcal{R}^x = \frac{N_{correct}^x}{N_{correct}^x + N_{miss}^x} 100\%$$

where

$$N_{correct}^x = |\Upsilon| \quad \text{with} \quad \Upsilon = \{S_i^x \mid i \in \mathbb{N}_1^{k_x} \mid \exists j \in \mathbb{N}_1^{\bar{k}_x} \text{ and } S_i^x \cap \bar{S}_j^x \neq \emptyset\},$$

and

$$N_{miss}^x = |\Upsilon| \quad \text{with} \quad \Upsilon = \{S_i^x \mid i \in \mathbb{N}_1^{k_x} \mid \forall j \in \mathbb{N}_1^{\bar{k}_x} \text{ and } S_i^x \cap \bar{S}_j^x = \emptyset\}.$$

**Definition 2** Given the set  $\Psi^x$  of real transitions of type  $x$ , the set  $\Phi$  of special effects frames, the set  $\bar{\Psi}^x$  of transitions of type  $x$  as declared by the detector, the precision level of the detector with respect to video sequence  $\mathcal{V}$  is defined as:

$$\mathcal{P}^x = \frac{N_{correct}^x}{N_{correct}^x + N_{false}^x} 100\%$$

where

$$N_{correct}^x = |\Upsilon| \quad \text{with} \quad \Upsilon = \{S_i^x \mid i \in \mathbb{N}_1^{k_x} \mid \exists j \in \mathbb{N}_1^{\bar{k}_x} \text{ and } S_i^x \cap \bar{S}_j^x \neq \emptyset\},$$

and

$$\begin{aligned} N_{false}^x = |\Upsilon| \quad \text{with} \quad \Upsilon = & \{\bar{S}_i^x \mid i \in \mathbb{N}_1^{\bar{k}} \mid \bar{S}_i^x \cap \Phi = \emptyset \\ & \text{and } \forall j \in \mathbb{N}_1^{k_{cut}} \bar{S}_i^x \cap S_j^{cut} = \emptyset \\ & \text{and } \forall j \in \mathbb{N}_1^{k_{fade}} \bar{S}_i^x \cap S_j^{fade} = \emptyset \\ & \text{and } \forall j \in \mathbb{N}_1^{k_{dissolve}} \bar{S}_i^x \cap S_j^{dissolve} = \emptyset\}. \end{aligned}$$

---

The degree of importance of precision against recall depends on specific applications. The high precision would be more crucial in video indexing than video compression, since false positives would have to be removed from indexing process while they may only slightly increase the size of compressed sequence.

Obviously, recall and precision parameters as defined above, while adequate for evaluating cut detection, do not take into consideration the detector accuracy in determining gradual transition boundaries which is important in video indexing, video content characterization, a director's style study and video compression. The boundaries of detected gradual transitions do not always coincide with the real boundaries. Figure 3.7 illustrates such a situation. We reuse cover-recall and cover-precision parameters proposed by (Lupatini *et al.*, 1998) to evaluate the detector ability to accurately locate the start and end of a fade or dissolve. While (Lupatini *et al.*, 1998) include both miss and false detections in the coverage evaluation, we believe that it is more appropriate to evaluate these parameters only on correctly detected transitions as defined previously. For each correctly detected transition  $S_i^x$ ,  $\bar{S}_{\alpha_i}$  denote the corresponding segment in the set of declared transitions  $\bar{\Psi}^x$ . Since a video sequence may contain more than one dissolves or fades, we take the average of cover-recall and cover-precision of all correctly detected transitions to measure the cover-recall and cover-precision of the detector with respect to the whole video sequence. Thus we can define cover-recall and cover-precision precisely as follows.

**Definition 3** The cover-recall of a transition detector with respect to video sequence  $\mathcal{V}$  is defined

Figure 3.7: Cover-precision and cover-recall

as:

$$\mathcal{R}_{cov}^x = \frac{\sum_{i=1}^{k_x} \theta_i}{|\Upsilon|}$$

with

$$\Upsilon = \{S_i^x \mid i \in \mathbb{N}_1^{k_x} \mid \exists j \in \mathbb{N}_1^{\bar{k}_x} \text{ and } S_i^x \cap \bar{S}_j^x \neq \emptyset\},$$

and

$$\theta_i = \begin{cases} 0, & \text{if } S_i^x \notin \Upsilon \\ \frac{|S_i^x \cap \bar{S}_{\alpha_i}|}{|S_i^x|} 100\% & \text{otherwise.} \end{cases}$$

**Definition 4** The cover-precision of a transition detector with respect to video sequence  $\mathcal{V}$  is defined as:

$$\mathcal{P}_{cov}^x = \frac{\sum_{i=1}^{k_x} \theta_i}{|\Upsilon|}$$

with

$$\Upsilon = \{S_i^x \mid i \in \mathbb{N}_1^{k_x} \mid \exists j \in \mathbb{N}_1^{\bar{k}_x} \text{ and } S_i^x \cap \bar{S}_j^x \neq \emptyset\},$$

and

$$\theta_i = \begin{cases} 0, & \text{if } S_i^x \notin \Upsilon \\ \frac{|S_i^x \cap \bar{S}_{\alpha_i}|}{|\bar{S}_{\alpha_i}|} 100\% & \text{otherwise.} \end{cases}$$

These four parameters allow a better evaluation of shot transition detection algorithms than previous work which assess the performance of shot transition detectors with respect to edit detection in general, but not with their ability to accurately classify and locate the temporal extent of transitions.

Since video data for this research is stored in MPEG compress format, the evaluation of computation time highly depends on the speed of the decoder. In addition, instead of implementing our own decoder, we use the MPEG library developed at Wayne State University Li and Sethi (1999). Therefore, it is impractical to perform an evaluation of computation time of the system. However, all proposed features are relatively simple and can be computed easily from a decoded luminance frame. Therefore, excluding the decoding time, the speed of the system should be comparable to other approaches proposed in the literature.

### 3.5.3 Results and Discussion

We compare the performance of the proposed algorithms with two other methods for detecting scene changes. The first one is part of WebFlix, a commercial tool for analyzing and editing MPEG videos. The algorithm for detecting cuts and dissolves by WebFlix is not known. WebFlix detects only one frame within a dissolve segment instead of its start and end; therefore, we will not evaluate cover-recall and cover-precision for WebFlix. The second one is a simple version of twin-comparison technique proposed by Zhang *et al.* (1993) described in chapter 2. We do not perform any sophisticated fine tuning for this algorithm. Instead we use 5 pairs of thresholds, and for each test sequence and for each type of transition the best detection result is recorded.

The shot transition detection results measured with respect to performance parameters described in the previous section are presented in table 3.2. The first impression is that our algorithms out-perform twin-comparison and WebFlix in almost all aspects. As for hard cut detection, our adaptive thresholding technique decreases the number of false positives and obtains a precision of 98 %, while still maintaining a very good level of recall of 98 % . Most flashlight effects in music videos are detected by the histogram verification step and removed; therefore, the performance of our algorithm on music sequences is still very high. More false positives occur in cartoon videos due to the discontinuous nature of motion in cartoons where histogram differences between consecutive frames often interleave between high and small values. While twin-comparison obtains a reasonable good level of recall, its precision is much lower than our algorithm, since it fails to deal adequately with artifacts caused by object and camera motion. In contrast, while precision of WebFlix algorithm is alright, two many cuts are missed by the algorithm. Most noticeably, it detects less than 50% of all cuts of the music sequences. Under close examination, we discover that WebFlix has problems with detecting changes in very dark scenes. Overall, in cut detection, WebFlix does not perform as well as the the twin-comparison approach, and far below our adaptive thresholding approach.

Our algorithm for detecting dissolves also performs better than WebFlix and the twin-comparison

Categories	Cuts		Dissolves				Fades			
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}_{\text{cov}}$	$\mathcal{R}_{\text{cov}}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}_{\text{cov}}$	$\mathcal{R}_{\text{cov}}$
Proposed Alg.	96.0	99.4	53.8	77.8	97.8	88.2	93.3	82.4	96.0	94.2
WebFlix	90.7	66.8	24.2	88.9	-	-	-	-	-	-
Twin-Comp.	91.2	96.4	12.7	77.8	91.7	87.5	-	-	-	-
(a) Results for cartoon sequences										
Proposed Alg.	97.0	97.0	79.2	87.1	97.2	80.8	89.7	92.9	94.9	99.6
WebFlix	90.6	80.1	54.3	62.9	-	-	-	-	-	-
Twin-Comp.	84.4	83.6	41.2	70.0	96.7	66.6	-	-	-	-
(b) Results for commercial sequences										
Proposed Alg.	98.7	96.7	82.4	79.2	96.0	84.0	62.5	100.0	100.0	100.0
WebFlix	97.4	78.2	20.1	39.0	-	-	-	-	-	-
Twin-Comp.	90.0	93.7	67.0	87.0	88.4	65.2	-	-	-	-
(c) Results for music sequences										
Proposed Alg.	98.5	98.5	64.7	64.7	93.9	52.4	92.5	96.1	98.5	99.2
WebFlix	86.2	46.5	26.7	70.6	-	-	-	-	-	-
Twin-Comp.	90.9	94.1	36.5	67.6	70.8	11.6	-	-	-	-
(d) Results for news sequences										
Proposed Alg.	98.1	97.7	73.2	86.9	94.7	84.6	90.0	90.0	91.5	98.8
WebFlix	93.4	86.3	16.0	48.6	-	-	-	-	-	-
Twin-Comp.	87.6	94.3	44.4	73.8	86.2	68.3	-	-	-	-
(e) Results for sports sequences										
Proposed Alg.	97.5	97.9	75.1	82.2	96.0	81.9	89.6	92.8	96.6	98.5
WebFlix	92.0	72.8	23.3	53.2	-	-	-	-	-	-
Twin-Comp.	88.6	92.1	43.7	75.8	90.4	65.0	-	-	-	-
(f) Overall results										

Table 3.2: Shot transition detection results

approach. It obtains a reasonable good level of recall of around 82%, while that for twin-comparison is 76%. This suggests that our algorithm would be able to detect dissolves whose color histogram differences between two consecutive frames is small, and therefore these dissolves are not detected by twin-comparison. The accuracy of our algorithm is also much better than twin-comparison, since most of false positives are eliminated by different thresholds set on mean and variance curve (see section 3.3). Since, almost all dissolves are detected after the zero-crossing detection step of our algorithm, we believe that these thresholds can be further refined to improve the recall while still maintaining approximately the same level of precision. The performance of WebFlix in detecting dissolves is very poor, as it misses near 50% of dissolves while only around 25% of declared dissolves are correct. The twin-comparison approach obtains a good level of cover-precision, since it is uncommon for fast motions to occur at boundary of a dissolve so that a correctly detected dissolve by twin-comparison would include false frames. However, the cover-recall by twin-comparison is rather low (65%), since the inter-frame differences at the start and end of a dissolve are very small and can not be detected by threshold  $T_g$ .

Apart from a slight improvement in cover-precision, our algorithm offers a much better level of cover-recall of 82%. The performance of our dissolve detection algorithm on cartoon and music sequences is not as good as on other categories. The reason for this is dissolves in cartoons are relatively uncommon (see table 3.1), and therefore, false positives start to dominate, while dissolves in dark music videos are missed by our algorithm. Among dissolves missed by our algorithms, a few are not cross-dissolves as presented in our model, but additive-dissolves which are relatively uncommon. In additive dissolves, the intensity of component shots do not scale at the same time, but one happens before another (Lienhart, 1999).

The performance of our fade detection algorithm is very good. Overall, it can detect 93% of fades, and 90% of declared fades are correct. In addition, the algorithm obtains a very high score of 97% and 99% in cover-precision and cover-recall, respectively. The lowest performance of our fade detection algorithm is for news sequences, and it obtains the precision level of only 63 %. However, this only slightly affects the overall results, since fades are uncommon in newscasts (see table 3.1 and chapter 4). The precision of fade detection would be further improved, if frames are divided into equal regions, e.g.  $4 \times 4$ , and fade characteristics described in section 3.2 are examined for each sub-region, since this approach will take into account the global nature of a fade operation.

## 3.6 Summary

In this chapter we have presented our algorithms for detecting different types of shot transition effects such as cut, fade, and dissolve. We improve conventional cut detection methods using color histogram difference by utilizing an adaptive threshold computed from a local window on color histogram difference curve. Based on the mathematical model for producing ideal fades and dissolves, different clues (e.g. monochrome frames) for discovering the existence of these effects are used, while other constraints are applied to eliminate the false positives caused by camera and object motion. We evaluate our algorithms against two other methods for shot transition detection, WebFlix and twin-comparison, on a relatively large data set and with respect to four parameters, recall, precision, cover-recall and cover-precision. The initial results are promising and can be further improved especially by choosing appropriate threshold settings.

## Chapter 4

# Video Genre Identification

In this chapter we describe a set of computational features originating from our study of editing effects, motion and color used in videos, for the task of automatic video categorization. These features besides representing human understanding of typical attributes of different video genres, are also inspired by the techniques and rules used by many directors to endow specific to a genre-program which lead to certain emotional impact on the viewers. We propose new features whilst also employing traditionally used ones for classification. This research, goes beyond the existing work with a systematic analysis of trends exhibited by each of our features in genres such as cartoons, commercials, music, news and sports, and it enables an understanding of similarities, dissimilarities between genres. This chapter is organized as follows. In the first section, we introduce some notations essential for describing the proposed features. The details of our feature set are presented in the following section, while the last section is devoted to describing the classification results using our feature set which is tested on several hours of video. The results establish the usefulness of this feature set. In addition, we explore the issue of video clip duration required to achieve reliable genre identification and demonstrate its impact on classification accuracy. The novelty of the chapter lies in (a) proposing new features; (b) analyzing trends of genres; (c) presenting the classification results for 8 hours of video data; (d) analyzing video clip duration and its impact on classification.

### 4.1 Notations

Consider a logical video clip,  $\mathcal{V}$ , a contiguous sequence of  $n + 1$  frames,  $\mathcal{V} = \{f_1, f_2, \dots, f_{n+1}\}$ . Here a video segment within a larger clip that is physically stored in disk is referred to as a *logical clip*. This means a physical clip may contain multiple logical clips, each of which can

be treated as a video data item for processing. A vector  $T = \{t_1, t_2, \dots, t_n\}$  is generated from  $V$ , where  $t_i$ , a *frame transition*, is a feature set computed jointly from frames  $f_i$  and  $f_{i+1}$ . For example:

$$\begin{aligned} t_i^\mu &= |f_{i+1}^\mu - f_i^\mu| \\ t_i^v &= |f_{i+1}^v - f_i^v| \end{aligned}$$

where  $f_i^\mu$  and  $f_i^v$  denote the mean and variance of pixel luminance of frame  $f_i$ , respectively. The concept of frame transition helps to define features involve inter-frame computation, because we can make all frames affected by shot transitions visible in vector  $\mathcal{T}$ . Since features such as camera work information need to use two consecutive frames for computation, we present them in vector transition space  $\mathcal{T}$ , rather frame space  $\mathcal{V}$ . For example,  $t_i^{\text{tilt}}$  is the amount of camera tilt between frame  $f_i$  and  $f_{i+1}$ .

After the video sequence is segmented into shots using methods proposed in the previous chapter, each member  $t_i$  of  $\mathcal{T}$  receives a label from the set,  $\mathcal{L} = \{\text{shot}(S), \text{cut}(C), \text{fade}(F), \text{dissolve}(D)\}$  depending on whether  $f_i$  and  $f_{i+1}$  are part of a pure shot, cut, fade or dissolve transition respectively. The transition vector  $\mathcal{T}$  is further segmented into  $k$  segments,  $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$ , where  $S_i$  is a set of consecutive elements  $t_{m_i}, t_{m_i+1}, \dots, t_{m_{(i+1)}-1}$  of  $\mathcal{T}$  having the same label, with  $m_1 = 1$  and  $m_{k-1} = n$ . Each  $S_i$  is also assigned a label from  $\mathcal{L}$  according to the type of frames it is generated from. Figure 4.1 illustrates the relation between  $\mathcal{V}$ ,  $\mathcal{T}$  and  $\mathcal{S}$  for a video sequence of 22 frames. This sequence has a cut between frame 4 and 5, a dissolve between frame 7 and 11, and a fade between frame 15 and 18. It can be seen that while a cut is not “visible” in  $\mathcal{V}$ , it can be seen in  $\mathcal{T}$  and  $\mathcal{S}$ .

Figure 4.1: Relation between  $\mathcal{V}$ ,  $\mathcal{T}$  and  $\mathcal{S}$

Let  $\Gamma^x$  and  $\Omega^x$  denote the set of segments  $S_i$  and  $t_i$  of type  $x$ , respectively. Let  $\Delta^{\text{shot}}$  denotes the set of pure shot frames in the video sequence, noting that generally  $|\Delta^{\text{shot}}| = |\Gamma^{\text{shot}}| + |\Omega^{\text{shot}}|$ . For the sample video sequence in figure 4.1, we have:

$$\Gamma^{\text{shot}} = \{\langle 1 \dots 3 \rangle, \langle 5 \dots 6 \rangle, \langle 11 \dots 12 \rangle, \langle 14 \dots 15 \rangle, \langle 18 \dots 21 \rangle\} \quad \text{and} \quad |\Gamma^{\text{shot}}| = 5$$

$$\Omega^{\text{shot}} = \{1, 2, 3, 5, 6, 11, 12, 14, 15, 18, 19, 20, 21\} \quad \text{and} \quad |\Omega^{\text{shot}}| = 13$$

$$\Delta^{\text{shot}} = \{1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22\} \quad \text{and} \quad |\Delta^{\text{shot}}| = 18.$$

## 4.2 Feature Extraction

In this section, we describe our feature sets which is grouped into editing, motion and color. Accompanying each feature is a plot of the values of the feature computed for 50 video samples (60sec each) randomly selected from each genre, after sorting values in ascending order.

### 4.2.1 Editing Characteristics

#### Shot Length

Shot length is a useful feature in video characterization, since it is fundamental to our perception of pace and content. The frequency of changing shots depend on visual and semantic continuity as well as the amount of motion, change and action the shot contains. It is also governed by the amount of time allocated for a video sequence and the amount of information to be assimilated by the audience. Therefore, short-duration shots are often used in commercials and music videos with fast music. In contrast, longer shots are used in sports to maintain the continuity of actions (see Figure 4.2.1). Shot length is measured as the number of frames between the last frame of the preceding transition and the first frame of the succeeding transition. So if a shot is corresponding to the segment  $S_i$  then its length would be  $|S_i| + 1$ . The average shot length of the whole logical clip,  $\mathcal{V}$ , is used as a classification feature:

$$\mathcal{F}_1 = \frac{\sum_{i=1}^k \theta_i}{|\Gamma^{\text{shot}}|}, \text{ where } \theta_i = \begin{cases} |S_i| + 1 & \text{if } S_i \in \Gamma^{\text{shot}} \\ 0 & \text{otherwise.} \end{cases}$$

We also attempted to capture the *cutting rhythm* by computing the standard deviation of shot length. Figure 4.2.1 show that overall, the trend is similar to average shot length, i.e. high for news news and sports, low for music and commercials.

#### Transition Types

Percentage of each type of transition used for editing can also identify a video genre. As we mentioned before, the shot transition method contributes in an unique way to the mood of the new shot and it has certain semantic implications regarding the nature of shots it links. We observe that while *fade* transitions are common in commercials and sometimes in music, they are rarely used in sports and news (see Figure 4.2.1). Fades are used in commercials, since it can effectively notify the viewer that a significant change in content, e.g. a new product, is about to occur. It can also be used simply to introduce a new commercial spot or exit the current one. Dissolves can be used in sports, especially sport highlights, to separate the main game

(a)  $\mathcal{F}_1$ : Average shot length

(b) Shot length standard deviation

Figure 4.2: Shot length related characteristics for 50 samples

action to other events such as the crowd scenes and slow motion replays. Children cartoons are supposed to be vivid, delightful and exciting. Slow relaxing mood and tempo rarely need to be conveyed; therefore, cuts are frequently used, instead of fades and dissolves. As can be seen from figure 4.2.1, the pattern of cut rate complements that of fade rate. Commercials have the lowest cut rate, while gradual transition appears to be uncommon in cartoons. We compute the percentage of each type of transitions  $x$ ,  $x \in \{cut, fade, dissolve\}$  as:

$$\mathcal{F}_2^x = \frac{|\Gamma^x|}{|\Gamma^{cut}| + |\Gamma^{fade}| + |\Gamma^{dissolve}|}$$

(a)  $\mathcal{F}_2^{fade}$ : Fade percentage

(b)  $\mathcal{F}_2^{cut}$ : Cut percentage

Figure 4.3: Transition related characteristics for 50 samples

### 4.2.2 Motion Estimation

Excluding noise in the video signal, changes in visual content between two consecutive frames can be caused either by object or camera motion. While camera movement may implies some semantics, its main use is to effectively capture the subject of interest. Motion characteristics therefore are largely determined by scene content.

#### Camera Movement

Camera movement influences the narration of scene content. In sports such as soccer and rugby fixed cameras are positioned around the field, and since the ball changes its position

continuously, a lot of camera movement is needed to track the ball continuously. In contrast, in newscasts, the object of interest such as an anchor person or a reporter remains relatively static (see figure 4.2.2). Camera motion magnitude between two consecutive frames is extracted using method proposed in Srinivasan *et al.* (1997), and the overall amount of camera movement of a video segment is captured using tilt and pan:

$$\mathcal{F}_3 = \frac{\sum_{i=1}^n \theta_i}{|\Omega^{\text{shot}}|} \quad , \text{ where } \quad \theta_i = \begin{cases} |t_i^{\text{tilt}}| + |t_i^{\text{pan}}| & \text{if } t_i \in \Omega^{\text{shot}} \\ 0 & \text{otherwise.} \end{cases}$$

Figure 4.4:  $\mathcal{F}_3$ : Average camera motion

### Motion Continuity

Another characteristic of motion is the average length of *motion runs*. A motion run  $R_i$  is an unbroken sequence of those frames,  $f_i$  whose sum of absolute pixel-wise luminance difference between  $f_i$  and  $f_{i+1}$  exceeds a certain threshold,  $T_4$ . Let  $|R_i|$  be the length of this run. Let  $R$  denote the set of all motion runs in the video clip. Then

$$\mathcal{F}_4 = \frac{\sum_i |R_i|}{|R|}$$

Figure 4.2.2 shows  $\mathcal{F}_6$  for the five genres.  $\mathcal{F}_6$  is consistently high for sports when compared against cartoons. The main reason for this is that motion in sports tends to occur continuously in time, whilst normally in the production process of a cartoon a single drawing may be exposed a number of times resulting in a lower pixel-wise difference between consecutive frames.

### Dynamic Scenes

We introduce two motion-related features to capture distribution of motion by measuring the prevalence of scenes with large change in content and those with little change in content between two frames. They are named “dynamic scenes” and “static scenes” respectively.

Figure 4.5:  $\mathcal{F}_4$ : Average length of motion runs

In music videos, there are often special effects such as quick changes of lighting, flash lights and very fast object/camera movements. These effects cause a large change in the variance of pixel luminance between two consecutive frames. We measure the prevalence of these effects as number of frame transitions whose pixel luminance variance is above a certain threshold:

$$\mathcal{F}_5 = \frac{|\Omega_1|}{|\Omega^{\text{shot}}|}, \text{ where } \Omega_1 = \{t_i \in \Omega^{\text{shot}} \mid t_i^v > T_1\}$$

Figure 4.6: The amount of dynamic scenes for 50 samples

Figure 4.2.2 shows  $\mathcal{F}_5$  as being distinctly higher for music videos than news. Although sports may have higher average motion than commercials, its motion tends to be constant throughout the program, while commercials have both high motion scenes and low motion scenes. Therefore, for this feature, sports have lower values than commercials.

### Static Scenes

The rate of “quiet” visual scenes, where both camera and object motion are very little, varies between different video categories. We expect music videos to have few static scenes, while newscasts should contain a high number of static scenes due to anchor, graphic and interview shots (see figure 4.2.2). The prevalence of static scenes in videos is measured based on the number of frame transitions whose both mean and variance are less than certain thresholds.

$$\mathcal{F}_6 = \frac{|\Omega_2|}{|\Omega^{\text{shot}}|} \quad \text{where} \quad \Omega_2 = \{t_i \in \Omega^{\text{shot}} \mid t_i^\mu < T_2 \text{ and } t_i^\nu < T_3\}$$

Figure 4.7: The amount of static scenes for 50 samples

### 4.2.3 Color Statistics

#### Color Histograms

There are also the distinctions in the distribution of color histograms between video genres. Let  $f_i^{\mathcal{H}}$  denote the luminance histogram of frame  $f_i$  and  $f_i^{\mathcal{H}_k}$  denote the set of indices of  $k$  largest bins in the color histogram, i.e. the set of  $k$  most prevalent luminance levels. We measure the coherence of these  $k$  bins based on  $\delta$ , the standard deviation of  $f_i^{\mathcal{H}_k}$ . Thus:

$$\mathcal{F}_7 = \frac{\sum_{i=1}^{n+1} \theta_i}{|\Delta^{\text{shot}}|} \quad , \text{ where} \quad \theta_i = \begin{cases} \delta(f_i^{\mathcal{H}_k}) & \text{if } f_i \in \Delta^{\text{shot}} \\ 0 & \text{otherwise.} \end{cases}$$

Figure 4.2.3 shows that sports have the lowest value of  $\mathcal{F}_7$ , since the color of the playing field tends to be highly homogeneous, whilst music videos tend to have high values of  $\mathcal{F}_7$  indicating high color variability. Instead of computing the standard deviation, we also investigate the mean

of the indices of  $k$  largest bins. Figure 4.2.3 shows that sports and cartoons have high value for this feature, since their pixels tend to have high luminance. This feature is omitted, since its pattern is similar to color brightness discussed next, which seems to have better discriminating capability.

(a)  $\mathcal{F}_7$ : Stdv. of the indices of largest bins

(b) Mean of the indices of largest bins

Figure 4.8: Color histogram related characteristics for 50 samples

### Brightness

HSV color space provides some other interesting features. We are only interested in the last two components of *HSV* color space: brightness ( $V$ ) and saturation ( $S$ ). The computation of these components from *RGB* color space is as follows.

$$V = \text{MAX}(R, G, B) \quad \text{and} \quad S = \frac{\text{MAX}(R, G, B) - \text{MIN}(R, G, B)}{\text{MAX}(R, G, B)}$$

We experimented with two methods to measure the brightness level of a frame. The first method is threshold-based and  $f_i^{\mathcal{V}}$  is the percentage of pixels having brightness above a threshold  $T_5$ . The second method is mean based and  $f_i^{\mathcal{V}}$  is the average value of pixel brightness. Thus:

$$\mathcal{F}_8 = \frac{\sum_{i=1}^{n+1} \theta_i}{|\Delta^{\text{shot}}|}, \quad \text{where} \quad \theta_i = \begin{cases} f_i^{\mathcal{V}} & \text{if } f_i \in \Delta^{\text{shot}} \\ 0 & \text{otherwise.} \end{cases}$$

with

$$f_i^{\mathcal{V}} = \frac{\sum_{x=1}^X \sum_{y=1}^Y f_i^{\mathcal{V}}(x, y)}{X \times Y}$$

or

$$f_i^{\mathcal{V}} = |\Upsilon| \quad \text{with} \quad \Upsilon = \{(x, y) \in \mathbb{N}_1^X \times \mathbb{N}_1^Y \mid f_i^{\mathcal{V}}(x, y) > T_5\}$$

(a)  $\mathcal{F}_8$ : Threshold based brightness level

(b) Mean based brightness level

Figure 4.9: Brightness related characteristics for 50 samples

Fig 4.2.3 shows that the average brightness for cartoons is much higher than other video genres if the threshold-based method is used, while mean-based approach as shown in figure 4.2.3 is less discriminating between genres. Therefore, we chose the threshold-based approach to compute the average brightness level of a video sequence.

### Saturation

(a)  $\mathcal{F}_9$ : Threshold based saturation level

(b) Mean based saturation level

Figure 4.10: Saturation related features for 50 samples

Similarly to brightness, we have two ways to measure the saturation of a frame. The overall formulation of the saturation feature is:

$$\mathcal{F}_9 = \frac{\sum_{i=1}^k \theta_i}{|\Delta^{\text{shot}}|}, \text{ where } \theta_i = \begin{cases} f_i^{S_{n+1}} & \text{if } f_i \in \Delta^{\text{shot}} \\ 0 & \text{otherwise.} \end{cases}$$

with

$$f_i^S = \frac{\sum_{x=1}^X \sum_{y=1}^Y f_i^S(x, y)}{X \times Y}$$

or

$$f_i^S = |\Upsilon| \quad \text{with} \quad \Upsilon = \{(x, y) \in \mathbb{N}_1^X \times \mathbb{N}_1^Y \mid f_i^S(x, y) > T_6\}$$

Figure shows that the average saturation for cartoons and sports is much higher when compared against commercials and music videos. In figure , the saturation level for cartoons only is high, while saturation level for sports is just like other genre. This suggests that sports have a moderate saturation, just above the threshold  $T_6$ . We chose the threshold-based method to compute the feature, since it can separate most of commercials from sports and cartoons.

### 4.3 Experimental Results

The data set is described in section 3.4.1. The overall flow of our video segmentation and classification system is shown in figure 4.11. The system has two logical subsystems: one for training and one for testing. For the training subsystem, after each frame is decoded, we extract the luminance component of the frame and compute different primitive features such as mean, variance, color histogram, camera motion, brightness and saturation. Histogram, mean and variance are used to detected shot transitions. After having all shots and transitions indexed, we combine all primitive features computed earlier with the segmentation results to construct clip-based features, i.e. one sample of our feature set. The C4.5 classification model is trained on all training samples to produce a classifier. The testing follows similar steps, except the features computed are put through the classifier produced by training subsystem. This classifier will output the genre that matches the features of the video.

The C4.5 decision tree Quinlan (1993) is used to build the classifier for genre labeling. Video clips are divided into units of approximately equal duration. The system was tested with features computed for basic clip durations of 40sec, 60sec, and 80sec. For each duration, 100 sets of training and testing data are generated from each genre, in which 60% of data is randomly selected for training while the remaining 40% of data is used for testing. The overall classification results are presented in Table 4.3. We measure in percentage the best, worst, average classification and standard deviation for each duration, as we expect that slightly different decision trees would be built for different data combinations.

The columns of the table are as follows: *All* represents the classification results when samples from all genres were used in training, while others such as  $\{-Ca\}$  represent classification results

Figure 4.11: Overview of the video genre classification system

obtained omitting samples from one given genre, say cartoons, during both training and testing ( $-\{Co\}$  is for omitting commercials,  $-\{Mu\}$ , music,  $-\{Ne\}$ , news, and  $-\{Sp\}$  for sports). The best result in each group are typeset in bold. In the best case for *All*, the classification rates are 86.2% (60sec), and 89.7% (80sec). The average classification for *All*, is between 80 % and 83%. Examination of the standard deviation of the classification implies that using the video clips of 60 sec duration is the most appropriate, as it offers the best trade off in terms of high classification and low standard deviation.

The best classification rate rises when one genre is omitted to around 92% due to patterns that exist in confusion matrix. It is useful to re-examine plots presented earlier. The average shot length, ( $\mathcal{F}_1$ ) and its trends across samples are similar for  $\langle$ commercial & music $\rangle$ , and also for  $\langle$ sports & news $\rangle$ . Cartoons fall somewhere in between, but can be confused with either of the four genres. The motion feature, ( $\mathcal{F}_3$ ) is similar for  $\langle$ news & commercials $\rangle$  and is close to but lower than music. However, all three categories are close. Further, cartoon features are close to those of news. Feature,  $\mathcal{F}_4$  is high for music, but is still close to commercials. However,  $\mathcal{F}_4$  well separates out news from  $\langle$ commercials & music $\rangle$ .  $\mathcal{F}_5$  clearly separates out  $\langle$ news, commercials, & music $\rangle$  and thus complements motion features  $\mathcal{F}_3$  and  $\mathcal{F}_4$ . Features,  $\mathcal{F}_6$ ,  $\mathcal{F}_8$ , and  $\mathcal{F}_9$  separate out cartoons from all other categories.  $\mathcal{F}_7$  separates out sports from music. A high degree of confusion can exist for news and sports since they are close in all features other than motion. Similarly, music and commercials have almost identical shot length and similar motion, and

Stats.	Dur.	All	-Ca.	-Co.	-Mu.	-Ne.	-Sp.
Best	40'	84.6	88.4	87.2	85.4	89.2	85.3
	60'	86.2	88.3	<b>92.3</b>	90.3	<b>91.5</b>	<b>89.2</b>
	80'	<b>89.7</b>	<b>91.4</b>	90.0	<b>90.4</b>	91.2	89.0
Worst	40'	78.4	83.4	83.1	81.1	83.5	80.3
	60'	<b>81.0</b>	83.1	<b>85.3</b>	<b>85.5</b>	<b>85.2</b>	<b>82.7</b>
	80'	79.5	<b>83.6</b>	83.1	83.7	82.3	80.5
Avg.	40'	80.0	84.8	84.5	82.2	85.2	82.0
	60'	<b>83.1</b>	85.3	<b>86.8</b>	<b>87.2</b>	<b>87.4</b>	84.8
	80'	81.7	<b>85.7</b>	85.0	86.1	85.2	<b>87.4</b>
Stdv.	40'	<b>1.41</b>	<b>1.21</b>	<b>1.05</b>	<b>1.07</b>	<b>1.40</b>	<b>1.45</b>
	60'	1.66	1.54	1.46	1.28	1.90	1.81
	80'	1.91	1.56	1.73	1.80	2.17	2.09

Table 4.1: Genre classification results.

can lead to a mix-up.

## 4.4 Summary

We have presented a set of features that embody the visual characteristics of a video sequence for video genre identification. The experimental results on several hours of videos indicate that these features perform well in classifying videos into sports, news, commercials, cartoons, and music, thus enabling automatic genre-based filtering during annotation and search. Our study on the length of a clip needed to recognize its genre indicates that 60sec can serve as the most appropriate video duration to achieve reliable classification accuracy. Future work will investigate temporal sequencing of shots and their semantics to further improve the performance of our system.

## Chapter 5

# Conclusion

### 5.1 Summary

In this research, we have set out to investigate the problem of video segmentation and classification. In particular, we have aimed at answering following questions:

- How can transitions between shots in a video sequence can be detected, classified and measured?
- What are the patterns in the visual content of a particular video genre?
- How these patterns can be captured, and hence enabling the automatic recognition of the video genre?

The answers to these questions play a crucial part in developing a video database system supporting automatic content indexing and retrieval. The segmentation of a video into shots is required in the early stage of video indexing, whereby shot boundaries are identified before objects and features within the shot can be recognized, segmented and computed. These objects and features can then be indexed and stored for future retrieval. Shot segmentation is also essential for extracting key-frames, a compact representation of the video sequence. The segmentation of a video sequences into logical units, e.g scenes, also relies on the shot transitions. Scenes are constructed based on analyzing the relationship and transitions between shots.

The first question is investigated in Chapter 3. The hard cut detection is performed by first computing the color histogram differences between consecutive frames. A local threshold is

computed based on the mean of all values within a sliding window to detect peaks in this curve. We use the characteristics of luminance mean and variance caused by fade and dissolve operations to detect these transitions. We test our algorithms against two STD tools, WebFlix and an implementation of Twin-Comparison algorithm, on a relatively large data set. The results indicate that our algorithms outperform these tools in all the tested criterion.

The second questions is investigated in Chapter 4. We have devised a feature set capturing editing, motion and color characteristics of a video sequence. Systematic analysis of trends exhibited by each or our features in genres such as cartoons, commercials, music, news and sports is presented, and this enables an understanding of the similarities and dissimilarities between genre. The experimental results confirm the usefulness of this feature set. In addition, we explore the issue of video clip duration required to achieve reliable genre identification and demonstrate its impact on classification accuracy.

Although shot transition detection and video classification at genre level have been studied in the literature, here we re-emphasize the following contributions made in this thesis:

- Provision of an adaptive threshold for improving the performance of hard cut detection.
- Development of an algorithm for fade detection based on the detection of monochrome frames and the shape of mean and variance curves.
- Formulation of an algorithm for dissolve detection based on the shape of mean and variance curves and a mechanism for obtaining robust thresholds by analyzing the mathematical models of dissolves.
- Development of a simple technique for reducing false positives in shot transition detection.
- Formulation of new features for video classification, whose validity has been tested on a large and comprehensive video data set.
- Presentation of systematic analysis of trends exhibited by each of our features in the 5 genre set: cartoons, commercials, music, news and sports.
- Demonstration of the impact of clip duration on classification accuracy.

## 5.2 Future Work

There are some speculative ideas for possible future extensions to the work presented here.

In order to improve the computation complexity, we can consider the approximation of these features using DC-Images extracted from an MPEG sequence (see Section 5.3). Furthermore, it is useful to compare the current detection performance of proposed algorithms against the case when features are approximately computed from DC-Images. The performance of fade and dissolve detection can be further improved if we first divide each frame into blocks, e.g.  $4 \times 4$  blocks, and examine them separately. The features for each blocks are combined to judge the existence of a transition. We ignore the problem of wipe detection in this thesis; however, we have made an initial investigation in the problem. Since wipes often cause a line moving across the screen, capturing the correlations between lines of consecutive images would enable wipe detection.

Some extensions can be made to our work on genre recognition. The existence and layout of text and captions in a video sequence would be different across genres. For example, while texts are common in commercials, they are rarely used in music videos. Further, while texts and captions normally appear at the bottom of a news program, they are placed at the middle of commercials frames. Also, shot distance, if can be automatically measured, would be a good feature for video content characterization. Close-up shots focus attention on a person's expression, feeling and emotion and therefore are very common in music videos, while long shots, which conveys the relationship, are common in sports. Although we only focus on visual features in this thesis, it would be interesting to investigate the integration of both audio and visual features in classifying videos. However, the most interesting extensions to this work is to investigate the temporal sequencing of shots and their semantics to further improve the performance of our system. This also mean we will investigate other learning techniques such as Hidden Markov Models or Bayesian Networks which can capture common structures of a sequence.

## Appendix A

# Shot Transition Types

Figure A.1: An example of a cut between frame 2 and 3

Figure A.2: An example of a fade to/from black

Figure A.3: An example of a fade to/from white

Figure A.4: An example of a dissolve

Figure A.5: An example of a standard wipe

Figure A.6: An example of a “slide-in” wipe

Figure A.7: An example of complex wipe

## Appendix B

# False Positives Identified by the Verification Step

Figure B.1: An falsely detected cut recognized during the verification step

Figure B.2: An falsely detected cut recognized during the verification step

Figure B.3: An falsely detected cut recognized during the verification step

Figure B.4: An falsely detected cut recognized during the verification step

## Appendix C

# Sample Classification Trees and Their Confusion Matrices

### C.1 Labels of Tree Nodes

## C.2 Five Categories: Cartoons, Commercials, Music, News and Sports

### C.2.1 For 40-second clips

0.8

```

Avg Shot Length (F1) <= 89.5455 :
| Avg Sat >20 (F9) <= 0.88986 :
| | Avg Static Scene(F6) <= 0.214971 : Music (43.0/8.9)
| | Avg Static Scene(F6) > 0.214971 :
| | | Avg Hist Mean12 (F7*) <= 126.269 :
| | | | Fade Percent (F2f) > 0.333333 : Music (4.0/3.0)
| | | | Fade Percent (F2f) <= 0.333333 :
| | | | | Cut Percentage (F2c) <= 0.96 : Commercial (58.0/10.3)
| | | | | Cut Percentage (F2c) > 0.96 :
| | | | | | Avg Sat >20 (F9) > 0.781917 : News (4.0/3.0)
| | | | | | Avg Sat >20 (F9) <= 0.781917 :
| | | | | | | Avg Dyn SceneF(F5) <= 0.01391 : Commercial (3.0/1.9)
| | | | | | | Avg Dyn SceneF(F5) > 0.01391 : Music (5.0/3.3)
| | | Avg Hist Mean12 (F7*) > 126.269 :
| | | | Avg Camera (F3) <= 1.40507 : Cartoon (8.0/3.8)
| | | | Avg Camera (F3) > 1.40507 : Commercial (4.0/3.0)
| Avg Sat >20 (F9) > 0.88986 :
| | Avg Mot Run Len(F4) <= 3.41509 :
| | | Avg Static Scene(F6) <= 0.092037 : Commercial (2.0/1.9)
| | | Avg Static Scene(F6) > 0.092037 : Cartoon (34.0/6.2)
| | Avg Mot Run Len(F4) > 3.41509 :
| | | Avg Bri Avg (F8*) > 161.291 : Music (4.0/3.5)
| | | Avg Bri Avg (F8*) <= 161.291 :
| | | | Avg Camera (F3) <= 1.83565 : News (3.0/1.9)
| | | | Avg Camera (F3) > 1.83565 : Sports (5.0/3.3)
Avg Shot Length (F1) > 89.5455 :
| Avg Bri 220 (F8) > 0.332157 : Cartoon (22.0/5.9)
| Avg Bri 220 (F8) <= 0.332157 :
| | Avg Camera (F3) <= 1.81708 :
| | | Fade Percent (F2f) <= 0.117647 :
| | | | Avg Dyn SceneF(F5) <= 0.016854 :

```

```

| | | | | Avg Hist Mean12 (F7*) <= 121.691 : News (91.0/17.7)
| | | | | Avg Hist Mean12 (F7*) > 121.691 :
| | | | | | Avg Mot Run Len(F4) <= 2.90625 :
| | | | | | | Std Shot Length (F1*) <= 292.592 : Cartoon (9.0/2.5)
| | | | | | | Std Shot Length (F1*) > 292.592 : News (2.0/1.9)
| | | | | | | Avg Mot Run Len(F4) > 2.90625 :
| | | | | | | Avg Static Scene(F6) <= 0.519953 : Sports (5.0/2.3)
| | | | | | | Avg Static Scene(F6) > 0.519953 : News (14.0/4.4)
| | | | | Avg Dyn SceneF(F5) > 0.016854 :
| | | | | | Std Shot Length (F1*) <= 92.5913 : Cartoon (6.0/2.4)
| | | | | | Std Shot Length (F1*) > 92.5913 :
| | | | | | | Std Shot Length (F1*) <= 148.697 : News (5.0/3.3)
| | | | | | | Std Shot Length (F1*) > 148.697 : Sports (4.0/3.0)
| | | | Fade Percent (F2f) > 0.117647 :
| | | | | Avg Camera (F3) <= 0.623256 : Commercial (3.0/1.9)
| | | | | Avg Camera (F3) > 0.623256 :
| | | | | | Avg Dyn SceneF(F5) <= 0.003177 : Music (2.0/1.9)
| | | | | | Avg Dyn SceneF(F5) > 0.003177 : News (4.0/3.0)
| | | Avg Camera (F3) > 1.81708 :
| | | | Avg Bri 220 (F8) <= 0.076804 : Sports (40.0/2.9)
| | | | Avg Bri 220 (F8) > 0.076804 :
| | | | | Avg Bri Avg (F8*) <= 129.28 :
| | | | | | Avg Hist Mean12 (F7*) <= 117.474 : News (6.0/2.4)
| | | | | | Avg Hist Mean12 (F7*) > 117.474 : Sports (2.0/1.6)
| | | | | | Avg Bri Avg (F8*) > 129.28 :
| | | | | | Avg Static Scene(F6) <= 0.389738 : Sports (39.0/2.9)
| | | | | | Avg Static Scene(F6) > 0.389738 : News (3.0/2.5)

```

Evaluation on training data (434 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
-----	-----	-----	-----	-----

79 26( 6.0%) 59 40( 9.2%) (26.8%) <<

Evaluation on test data (292 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
79	60(20.5%)	59	60(20.5%)	(26.8%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
36	2	8	1	2	(a): class Commercial
6	62		5	7	(b): class News
7	1	29		1	(c): class Music
	10		59	5	(d): class Sports
	2	3		46	(e): class Cartoon

## C.2.2 For 60-second clips

1

```

Avg Shot Length (F1) <= 84.1579 :
| Avg Sat >20 (F9) <= 0.897165 :
| | Avg Static Scene(F6) <= 0.228705 :
| | | Avg Sat >20 (F9) <= 0.81823 : Music (17.0/2.7)
| | | Avg Sat >20 (F9) > 0.81823 : Commercial (2.0/1.9)
| | Avg Static Scene(F6) > 0.228705 :
| | | Avg Bri Avg (F8*) <= 146.651 :
| | | | Avg Sat >20 (F9) <= 0.414545 : Music (3.0/1.9)
| | | | Avg Sat >20 (F9) > 0.414545 :
| | | | | Fade Percent (F2f) > 0.333333 : Music (2.0/1.6)
| | | | | Fade Percent (F2f) <= 0.333333 :
| | | | | Avg Sat Avg (F9*) <= 101.768 :
| | | | | | Avg Camera (F3) <= 2.11873 : Commercial (41.0/4.9)
| | | | | | Avg Camera (F3) > 2.11873 : Music (3.0/2.5)
| | | | | | Avg Sat Avg (F9*) > 101.768 :
| | | | | | Avg Sat >20 (F9) <= 0.756335 : Music (4.0/2.1)
| | | | | | Avg Sat >20 (F9) > 0.756335 : Commercial (3.0/1.9)
| | | Avg Bri Avg (F8*) > 146.651 :
| | | | Avg Camera (F3) <= 1.31388 : Cartoon (5.0/3.3)
| | | | Avg Camera (F3) > 1.31388 : Commercial (3.0/2.5)
| Avg Sat >20 (F9) > 0.897165 :
| | Avg Mot Run Len(F4) <= 2.75714 : Cartoon (22.0/2.8)
| | Avg Mot Run Len(F4) > 2.75714 :
| | | Avg Shot Length (F1) <= 70.8095 : Music (4.0/2.1)
| | | Avg Shot Length (F1) > 70.8095 : Sports (2.0/1.9)
Avg Shot Length (F1) > 84.1579 :
| Avg Bri Avg (F8*) > 184.367 : Cartoon (13.0/2.7)
| Avg Bri Avg (F8*) <= 184.367 :
| | Avg Camera (F3) <= 1.55927 :
| | | Avg Mot Run Len(F4) <= 4.11888 :
| | | | Avg Hist Mean12 (F7*) <= 121.896 : News (16.0/4.5)
| | | | Avg Hist Mean12 (F7*) > 121.896 :
| | | | | Avg Shot Length (F1) <= 185.444 : Cartoon (12.0/5.3)
| | | | | Avg Shot Length (F1) > 185.444 : News (2.0/1.9)
| | | Avg Mot Run Len(F4) > 4.11888 :

```

```

| | | | Std Shot Length (F1*) > 292.592 : Sports (3.0/1.9)
| | | | Std Shot Length (F1*) <= 292.592 :
| | | | | Avg Bri 220 (F8) <= 0.034302 : Sports (4.0/3.0)
| | | | | Avg Bri 220 (F8) > 0.034302 :
| | | | | | Avg Static Scene(F6) <= 0.358339 : Sports (4.0/3.5)
| | | | | | Avg Static Scene(F6) > 0.358339 : News (37.0/2.9)
| | Avg Camera (F3) > 1.55927 :
| | | Avg Hist Mean12 (F7*) > 109.973 : Sports (45.0/5.0)
| | | Avg Hist Mean12 (F7*) <= 109.973 :
| | | | Avg Shot Length (F1) <= 137.769 : News (7.0/3.7)
| | | | Avg Shot Length (F1) > 137.769 :
| | | | | Avg Hist Mean12 (F7*) <= 101.964 : Sports (16.0/2.7)
| | | | | Avg Hist Mean12 (F7*) > 101.964 :
| | | | | | Std Shot Length (F1*) <= 136.323 : Sports (2.0/1.6)
| | | | | | Std Shot Length (F1*) > 136.323 : News (3.0/1.9)

```

Evaluation on training data (275 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
57	12( 4.4%)	51	15( 5.5%)	(26.4%) <<

Evaluation on test data (189 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
57	27(14.3%)	51	28(14.8%)	(26.4%) <<

(a) (b) (c) (d) (e) <-classified as

---

26		6		1	(a): class Commercial
	43		2	2	(b): class News
3	1	19		1	(c): class Music
	8		42		(d): class Sports
2	1	1		31	(e): class Cartoon

## C.2.3 For 80-second clips

1

```

Avg Shot Length (F1) <= 85.2174 :
| Avg Sat >20 (F9) <= 0.8972 :
| | PD1 lt 0.015 (22) <= 0.207921 : Music (23.0/6.0)
| | PD1 lt 0.015 (22) > 0.207921 :
| | | Cut Percentage (F2c) <= 0.97619 : Commercial (44.0/12.4)
| | | Cut Percentage (F2c) > 0.97619 : Music (3.0/1.9)
| Avg Sat >20 (F9) > 0.8972 :
| | Avg Mot Run Len(F4) <= 2.85714 : Cartoon (20.0/4.6)
| | Avg Mot Run Len(F4) > 2.85714 :
| | | Avg Shot Length (F1) <= 63.1765 : Music (3.0/1.9)
| | | Avg Shot Length (F1) > 63.1765 : News (3.0/2.5)
Avg Shot Length (F1) > 85.2174 :
| Avg Bri 220 (F8) <= 0.259144 :
| | Avg Camera (F3) <= 1.4469 :
| | | Avg Bri Avg (F8*) <= 148.124 : News (45.0/6.4)
| | | Avg Bri Avg (F8*) > 148.124 : Cartoon (5.0/4.6)
| | Avg Camera (F3) > 1.4469 :
| | | Avg Sat >20 (F9) > 0.933183 : Sports (29.0/2.8)
| | | Avg Sat >20 (F9) <= 0.933183 :
| | | | Avg Bri 220 (F8) <= 0.035932 : Sports (12.0/2.7)
| | | | Avg Bri 220 (F8) > 0.035932 :
| | | | | Fade Percent (F2f) > 0.02439 : Sports (4.0/2.1)
| | | | | Fade Percent (F2f) <= 0.02439 :
| | | | | Avg Shot Length (F1) > 194.636 : Sports (3.0/1.9)
| | | | | Avg Shot Length (F1) <= 194.636 :
| | | | | PD1 lt 0.015 (22) <= 0.136364 : Sports (3.0/2.5)
| | | | | PD1 lt 0.015 (22) > 0.136364 : News (11.0/2.6)
| Avg Bri 220 (F8) > 0.259144 :
| | Avg Mot Run Len(F4) <= 4.65563 : Cartoon (16.0/2.7)
| | Avg Mot Run Len(F4) > 4.65563 : News (5.0/3.3)

```

Evaluation on training data (229 items):

Before Pruning

After Pruning

-----

Size	Errors	Size	Errors	Estimate	
45	12( 5.2%)	31	18( 7.9%)	(26.6%)	<<

Evaluation on test data (156 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
45	23(14.7%)	31	23(14.7%)	(26.6%)	<<

(a)	(b)	(c)	(d)	(e)	<-classified as
22		4			(a): class Commercial
1	39		2	1	(b): class News
5		16			(c): class Music
	3		34	1	(d): class Sports
2	3	1		22	(e): class Cartoon

### C.3 Four Categories: Commercials, Music, News and Sports

#### C.3.1 For 40-second clips

1

```

Avg Shot Length (F1) <= 84.0833 :
| Avg Static Scene(F6) <= 0.214971 : Music (44.0/10.1)
| Avg Static Scene(F6) > 0.214971 :
| | Avg Hist Stdv20 (F7) <= 14.6659 :
| | | Dissolve Percent(F2d) > 0 : Music (3.0/2.5)
| | | Dissolve Percent(F2d) <= 0 :
| | | | Std Shot Length (F1*) <= 69.5578 : Sports (3.0/1.9)
| | | | Std Shot Length (F1*) > 69.5578 : News (2.0/1.6)
| | Avg Hist Stdv20 (F7) > 14.6659 :
    
```

```

| | | PD1 lt 0.015 (22) <= 0.198552 :
| | | | Avg Camera (F3) > 1.37949 : Music (6.0/2.4)
| | | | Avg Camera (F3) <= 1.37949 :
| | | | | Std Shot Length (F1*) <= 46.5321 : Commercial (3.0/1.9)
| | | | | Std Shot Length (F1*) > 46.5321 : Music (2.0/1.6)
| | | PD1 lt 0.015 (22) > 0.198552 :
| | | | Avg Static Scene(F6) > 0.361587 : Commercial (42.0/7.6)
| | | | Avg Static Scene(F6) <= 0.361587 :
| | | | | Avg Sat >20 (F9) <= 0.492331 : Music (4.0/2.1)
| | | | | Avg Sat >20 (F9) > 0.492331 :
| | | | | | Cut Percentage (F2c) <= 0.980769 : Commercial (18.0/2.8)
| | | | | | Cut Percentage (F2c) > 0.980769 : Music (2.0/1.6)
Avg Shot Length (F1) > 84.0833 :
| Avg Camera (F3) <= 1.81708 :
| | Avg Sat >50 (51) <= 0.850412 :
| | | Avg Dyn SceneF(F5) <= 0.014472 :
| | | | Std Shot Length (F1*) <= 31.4788 : Commercial (2.0/1.6)
| | | | Std Shot Length (F1*) > 31.4788 :
| | | | | Avg Camera (F3) <= 1.56331 : News (80.0/6.5)
| | | | | Avg Camera (F3) > 1.56331 :
| | | | | | Avg Mot Run Len(F4) <= 3.90323 : Sports (4.0/2.1)
| | | | | | Avg Mot Run Len(F4) > 3.90323 : News (10.0/4.1)
| | | Avg Dyn SceneF(F5) > 0.014472 :
| | | | Std Shot Length (F1*) <= 119.316 : News (11.0/4.2)
| | | | Std Shot Length (F1*) > 119.316 :
| | | | | PD1 lt 0.015 (22) <= 0.347407 : Sports (6.0/3.5)
| | | | | PD1 lt 0.015 (22) > 0.347407 : Commercial (6.0/3.5)
| | Avg Sat >50 (51) > 0.850412 :
| | | Avg Shot Length (F1) > 268.571 : Sports (7.0/2.4)
| | | Avg Shot Length (F1) <= 268.571 :
| | | | Avg Sat >20 (F9) <= 0.955814 : News (5.0/2.3)
| | | | Avg Sat >20 (F9) > 0.955814 : Sports (5.0/3.3)
| Avg Camera (F3) > 1.81708 :
| | PD1 lt 0.015 (22) <= 0.144654 : Sports (55.0/2.9)
| | PD1 lt 0.015 (22) > 0.144654 :
| | | Avg Sat >50 (51) <= 0.687773 : News (4.0/2.1)
| | | Avg Sat >50 (51) > 0.687773 :
| | | | Avg Bri 220 (F8) <= 0.089756 : Sports (20.0/4.6)
| | | | Avg Bri 220 (F8) > 0.089756 :

```

```

| | | | | Avg Hist Mean12 (F7*) <= 116.759 : News (6.0/3.5)
| | | | | Avg Hist Mean12 (F7*) > 116.759 : Sports (8.0/3.8)

```

Evaluation on training data (358 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
59	14( 3.9%)	51	19( 5.3%)	(24.1%) <<

Evaluation on test data (241 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
59	41(17.0%)	51	38(15.8%)	(24.1%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
40	1	7	1		(a): class Commercial
9	67	1	3		(b): class News
4	2	31	1		(c): class Music
2	7		65		(d): class Sports
					(e): class Cartoon

## C.3.2 For 60-second clips

1

```

Avg Shot Length (F1) <= 84.9231 :
|  Cut Percentage (F2c) <= 0.96875 :
|  |  Avg Hist Stdv20 (F7) > 57.8969 : Music (6.0/2.4)
|  |  Avg Hist Stdv20 (F7) <= 57.8969 :
|  |  |  Avg Dyn SceneF(F5) <= 0.003987 :
|  |  |  |  Avg Mot Run Len(F4) <= 1.67241 : Commercial (3.0/2.5)
|  |  |  |  Avg Mot Run Len(F4) > 1.67241 : Music (6.0/2.4)
|  |  |  Avg Dyn SceneF(F5) > 0.003987 :
|  |  |  |  Avg Sat >20 (F9) <= 0.418216 : Music (3.0/1.9)
|  |  |  |  Avg Sat >20 (F9) > 0.418216 : Commercial (48.0/7.7)
|  Cut Percentage (F2c) > 0.96875 :
|  |  Avg Dyn SceneF(F5) <= 0.007948 : Commercial (3.0/2.9)
|  |  Avg Dyn SceneF(F5) > 0.007948 : Music (15.0/2.7)
Avg Shot Length (F1) > 84.9231 :
|  Avg Camera (F3) > 1.65689 : Sports (66.0/11.6)
|  Avg Camera (F3) <= 1.65689 :
|  |  Avg Static Scene(F6) <= 0.444924 :
|  |  |  Avg Shot Length (F1) > 176.667 : Sports (7.0/2.4)
|  |  |  Avg Shot Length (F1) <= 176.667 :
|  |  |  |  Dissolve Percent(F2d) <= 0.181818 : News (6.0/3.5)
|  |  |  |  Dissolve Percent(F2d) > 0.181818 : Sports (2.0/1.6)
|  |  Avg Static Scene(F6) > 0.444924 :
|  |  |  Std Shot Length (F1*) <= 448.766 : News (57.0/5.0)
|  |  |  Std Shot Length (F1*) > 448.766 : Sports (2.0/1.6)

```

Tree saved

Evaluation on training data (224 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
-----		-----		

39 6( 2.7%) 25 14( 6.2%) (21.5%) <<

Evaluation on test data (154 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
39	28(18.2%)	25	17(11.0%)	(21.5%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
31		2			(a): class Commercial
	40		7		(b): class News
7		17			(c): class Music
	1		49		(d): class Sports
					(e): class Cartoon

C.3.3 For 80-second clips

1

```

Avg Shot Length (F1) <= 82.1111 :
| Avg Static Scene(F6) <= 0.219512 : Music (21.0/2.8)
| Avg Static Scene(F6) > 0.219512 :
| | PD1 lt 0.015 (22) <= 0.152871 : Music (5.0/3.3)
| | PD1 lt 0.015 (22) > 0.152871 :
| | | Avg Sat >20 (F9) <= 0.894803 : Commercial (45.0/11.3)
| | | Avg Sat >20 (F9) > 0.894803 : News (3.0/2.5)
Avg Shot Length (F1) > 82.1111 :
| Avg Camera (F3) <= 1.85819 :
| | Std Shot Length (F1*) <= 366.272 : News (64.0/15.1)
| | Std Shot Length (F1*) > 366.272 : Sports (4.0/2.1)
| Avg Camera (F3) > 1.85819 :
| | Avg Sat >50 (51) <= 0.651255 : News (2.0/1.6)
| | Avg Sat >50 (51) > 0.651255 : Sports (43.0/6.3)
    
```

Evaluation on training data (187 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
35	7( 3.7%)	15	19(10.2%)	(24.1%)	<<

Evaluation on test data (128 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
35	22(17.2%)	15	20(15.6%)	(24.1%)	<<

(a)	(b)	(c)	(d)	(e)	<-classified as
----	----	----	----	----	
19	3	4			(a): class Commercial
	42		1		(b): class News
5	2	14			(c): class Music
	5		33		(d): class Sports
					(e): class Cartoon

## C.4 Four Categories: Cartoons, Music, News and Sports

### C.4.1 For 40-second clips

0.8

Avg Shot Length (F1) <= 84.5 :

| Avg Sat >20 (F9) <= 0.905348 :

| | Avg Static Scene(F6) <= 0.446592 :

| | | Avg Bri Avg (F8\*) <= 196.38 : Music (54.0/7.8)

| | | Avg Bri Avg (F8\*) > 196.38 : Cartoon (2.0/1.6)

| | Avg Static Scene(F6) > 0.446592 :

| | | Avg Hist Mean12 (F7\*) <= 124.092 : News (5.0/2.3)

| | | Avg Hist Mean12 (F7\*) > 124.092 : Cartoon (3.0/1.9)

| Avg Sat >20 (F9) > 0.905348 :

| | Avg Mot Run Len(F4) <= 3.83942 : Cartoon (33.0/2.9)

| | Avg Mot Run Len(F4) > 3.83942 :

| | | Std Shot Length (F1\*) <= 42.9651 : News (3.0/2.5)

| | | Std Shot Length (F1\*) > 42.9651 :

| | | | Avg Camera (F3) <= 1.74745 : Music (2.0/1.6)

| | | | Avg Camera (F3) > 1.74745 : Sports (3.0/1.9)

Avg Shot Length (F1) > 84.5 :

| Avg Camera (F3) <= 1.56331 :

| | Avg Bri 220 (F8) <= 0.277771 :

| | | Avg Sat >20 (F9) <= 0.952664 : News (91.0/18.9)

| | | Avg Sat >20 (F9) > 0.952664 :

| | | | Avg Shot Length (F1) > 154.571 : Sports (7.0/3.7)

```

| | | | Avg Shot Length (F1) <= 154.571 :
| | | | | Avg Mot Run Len(F4) <= 2.6 : Cartoon (5.0/2.3)
| | | | | Avg Mot Run Len(F4) > 2.6 : News (2.0/1.6)
| | Avg Bri 220 (F8) > 0.277771 :
| | | Avg Mot Run Len(F4) <= 4.22 : Cartoon (24.0/2.8)
| | | Avg Mot Run Len(F4) > 4.22 :
| | | | Avg Sat Avg (F9*) <= 147.898 : News (8.0/2.5)
| | | | Avg Sat Avg (F9*) > 147.898 : Cartoon (2.0/1.6)
| Avg Camera (F3) > 1.56331 :
| | Avg Camera (F3) > 1.81696 : Sports (84.0/10.5)
| | Avg Camera (F3) <= 1.81696 :
| | | Avg Hist Stdv20 (F7) > 27.368 : News (12.0/4.2)
| | | Avg Hist Stdv20 (F7) <= 27.368 :
| | | | Avg Hist Mean12 (F7*) > 119.374 : Sports (10.0/2.6)
| | | | Avg Hist Mean12 (F7*) <= 119.374 :
| | | | | Avg Static Scene(F6) <= 0.547909 : News (5.0/2.3)
| | | | | Avg Static Scene(F6) > 0.547909 : Sports (6.0/3.5)

```

Evaluation on training data (361 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
55	16( 4.4%)	39	24( 6.6%)	(21.8%) <<

Evaluation on test data (243 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
55	35(14.4%)	39	34(14.0%)	(21.8%) <<

(a) (b) (c) (d) (e) <-classified as

-----

---

				(a): class Commercial
64	2	13	1	(b): class News
	37		1	(c): class Music
8		66		(d): class Sports
7	2		42	(e): class Cartoon

## C.4.2 For 60-second clips

1

```

Avg Shot Length (F1) <= 83.6667 :
| Avg Hist Mean12 (F7*) <= 113.014 : Music (31.0/6.2)
| Avg Hist Mean12 (F7*) > 113.014 :
| | Std Shot Length (F1*) <= 23.5806 : Music (2.0/1.6)
| | Std Shot Length (F1*) > 23.5806 :
| | | Avg Mot Run Len(F4) <= 4.57252 : Cartoon (26.0/2.8)
| | | Avg Mot Run Len(F4) > 4.57252 : Music (3.0/2.5)
Avg Shot Length (F1) > 83.6667 :
| Avg Bri Avg (F8*) > 184.367 : Cartoon (15.0/2.7)
| Avg Bri Avg (F8*) <= 184.367 :
| | Avg Camera (F3) > 1.61177 : Sports (75.0/16.4)
| | Avg Camera (F3) <= 1.61177 :
| | | Std Shot Length (F1*) > 355.866 : Sports (4.0/2.1)
| | | Std Shot Length (F1*) <= 355.866 :
| | | | Avg Bri 220 (F8) <= 0.246041 : News (58.0/9.1)
| | | | Avg Bri 220 (F8) > 0.246041 :
| | | | | Avg Mot Run Len(F4) <= 4.25714 : Cartoon (7.0/2.4)
| | | | | Avg Mot Run Len(F4) > 4.25714 :
| | | | | | Avg Static Scene(F6) <= 0.400564 : Music (2.0/1.9)
| | | | | | Avg Static Scene(F6) > 0.400564 : News (4.0/2.1)

```

Evaluation on training data (227 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
41	8( 3.5%)	21	18( 7.9%)	(22.0%) <<

Evaluation on test data (156 items):

Before Pruning	After Pruning
----------------	---------------

```

-----
Size      Errors   Size      Errors   Estimate
41      16(10.3%)   21      18(11.5%)   (22.0%)  <<

```

```

(a) (b) (c) (d) (e)      <-classified as
-----
                                (a): class Commercial
                                (b): class News
                                (c): class Music
                                (d): class Sports
                                (e): class Cartoon
40      1      4      2
      22      1      1
4      1      45
2      2              31

```

## C.4.3 For 80-second clips

1

```

Avg Shot Length (F1) <= 84.5 :
|   Avg Camera (F3) <= 1.33 :
|   |   Avg Shot Length (F1) <= 35.7447 : Music (6.0/1.0)
|   |   Avg Shot Length (F1) > 35.7447 :
|   |   |   Cut Percentage (F2c) <= 0.333333 : News (2.0)
|   |   |   Cut Percentage (F2c) > 0.333333 : Cartoon (22.0/1.0)
|   Avg Camera (F3) > 1.33 :
|   |   Avg Bri Avg (F8*) <= 174.77 : Music (24.0)
|   |   Avg Bri Avg (F8*) > 174.77 : Cartoon (3.0/1.0)
Avg Shot Length (F1) > 84.5 :
|   PD1 lt 0.015 (22) <= 0.266141 :
|   |   Avg Sat >50      (51) > 0.7912 : Sports (36.0)
|   |   Avg Sat >50      (51) <= 0.7912 :
|   |   |   Std Shot Length (F1*) <= 115.207 : News (6.0/1.0)
|   |   |   Std Shot Length (F1*) > 115.207 : Sports (9.0)
|   PD1 lt 0.015 (22) > 0.266141 :
|   |   Avg Mot Run Len(F4) <= 2.83333 :
|   |   |   Avg Hist Mean12 (F7*) <= 109.071 : News (4.0/1.0)
|   |   |   Avg Hist Mean12 (F7*) > 109.071 : Cartoon (16.0)
|   |   Avg Mot Run Len(F4) > 2.83333 :
|   |   |   Avg Bri Avg (F8*) > 181.648 : Cartoon (2.0)
|   |   |   Avg Bri Avg (F8*) <= 181.648 :
|   |   |   |   Avg Bri 220      (F8) <= 0.035932 : Sports (6.0/1.0)
|   |   |   |   Avg Bri 220      (F8) > 0.035932 :
|   |   |   |   |   Std Shot Length (F1*) <= 366.272 : News (52.0/1.0)
|   |   |   |   |   Std Shot Length (F1*) > 366.272 : Sports (2.0)

```

Evaluation on training data (190 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
-----	-----	-----	-----	-----

27 7( 3.7%) 27 7( 3.7%) (21.7%) <<

Evaluation on test data (130 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
27	16(12.3%)	27	16(12.3%)	(21.7%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
					(a): class Commercial
	40		1	2	(b): class News
		18		3	(c): class Music
	6	1	31		(d): class Sports
	1	2		25	(e): class Cartoon

## C.5 Four Categories: Cartoons, Commercials, News and Sports

### C.5.1 For 40-second clips

1

```

Avg Shot Length (F1) <= 91.5833 :
| Avg Sat >20 (F9) <= 0.9027 :
| | Avg Hist Mean12 (F7*) <= 123.269 :
| | | Avg Mot Run Len(F4) <= 1 : News (2.0/1.9)
| | | Avg Mot Run Len(F4) > 1 : Commercial (66.0/10.4)
| | Avg Hist Mean12 (F7*) > 123.269 :
| | | Std Shot Length (F1*) <= 39.4153 : Commercial (5.0/2.3)
| | | Std Shot Length (F1*) > 39.4153 : Cartoon (6.0/3.5)
    
```

```

|   Avg Sat >20 (F9) > 0.9027 :
|   |   Avg Mot Run Len(F4) <= 3.56522 : Cartoon (39.0/4.9)
|   |   Avg Mot Run Len(F4) > 3.56522 :
|   |   |   Avg Bri Avg (F8*) > 163.071 : Cartoon (3.0/2.5)
|   |   |   Avg Bri Avg (F8*) <= 163.071 :
|   |   |   |   Avg Camera (F3) <= 1.83007 : News (8.0/3.8)
|   |   |   |   Avg Camera (F3) > 1.83007 : Sports (6.0/3.5)
Avg Shot Length (F1) > 91.5833 :
|   Avg Bri Avg (F8*) > 184.829 : Cartoon (14.0/2.7)
|   Avg Bri Avg (F8*) <= 184.829 :
|   |   Avg Camera (F3) <= 1.51668 :
|   |   |   Avg Sat >20 (F9) <= 0.952682 :
|   |   |   |   Avg Dyn SceneF(F5) <= 0.013468 : News (80.0/11.7)
|   |   |   |   Avg Dyn SceneF(F5) > 0.013468 :
|   |   |   |   |   Std Shot Length (F1*) <= 119.469 :
|   |   |   |   |   |   Avg Bri Avg (F8*) <= 129.898 : News (4.0/3.0)
|   |   |   |   |   |   Avg Bri Avg (F8*) > 129.898 : Cartoon (6.0/2.4)
|   |   |   |   |   |   Std Shot Length (F1*) > 119.469 :
|   |   |   |   |   |   |   Avg Sat >20 (F9) <= 0.794522 : Commercial (2.0/1.6)
|   |   |   |   |   |   |   Avg Sat >20 (F9) > 0.794522 : Sports (3.0/1.9)
|   |   |   |   |   |   Avg Sat >20 (F9) > 0.952682 :
|   |   |   |   |   |   |   Avg Shot Length (F1) <= 156.571 : Cartoon (6.0/2.4)
|   |   |   |   |   |   |   Avg Shot Length (F1) > 156.571 : Sports (7.0/3.7)
|   |   |   |   |   |   Avg Camera (F3) > 1.51668 :
|   |   |   |   |   |   |   Avg Sat >20 (F9) > 0.956006 : Sports (37.0/2.9)
|   |   |   |   |   |   |   Avg Sat >20 (F9) <= 0.956006 :
|   |   |   |   |   |   |   |   Avg Hist Stdv20 (F7) <= 26.7702 :
|   |   |   |   |   |   |   |   |   Avg Bri 220 (F8) <= 0.041439 : Sports (27.0/2.8)
|   |   |   |   |   |   |   |   |   Avg Bri 220 (F8) > 0.041439 :
|   |   |   |   |   |   |   |   |   |   Avg Hist Mean12 (F7*) > 107.282 : Sports (25.0/7.2)
|   |   |   |   |   |   |   |   |   |   Avg Hist Mean12 (F7*) <= 107.282 :
|   |   |   |   |   |   |   |   |   |   |   Fade Percent (F2f) > 0.069767 : Sports (2.0/1.6)
|   |   |   |   |   |   |   |   |   |   |   Fade Percent (F2f) <= 0.069767 :
|   |   |   |   |   |   |   |   |   |   |   |   Avg Shot Length (F1) > 135 : News (8.0/2.5)
|   |   |   |   |   |   |   |   |   |   |   |   Avg Shot Length (F1) <= 135 :
|   |   |   |   |   |   |   |   |   |   |   |   |   Avg Static Scene(F6) <= 0.432836 : Sports (3.0/1.9)
|   |   |   |   |   |   |   |   |   |   |   |   |   Avg Static Scene(F6) > 0.432836 : News (3.0/1.9)
|   |   |   |   |   |   |   |   |   |   |   |   Avg Hist Stdv20 (F7) > 26.7702 :
|   |   |   |   |   |   |   |   |   |   |   |   |   Avg Hist Mean12 (F7*) <= 117.474 : News (13.0/2.7)

```

| | | | | Avg Hist Mean12 (F7\*) > 117.474 : Sports (3.0/2.5)

Evaluation on training data (378 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
63	15( 4.0%)	49	23( 6.1%)	(23.3%) <<

Evaluation on test data (254 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
63	33(13.0%)	49	33(13.0%)	(23.3%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
45	2			2	(a): class Commercial
4	66		7	3	(b): class News
					(c): class Music
	6		68		(d): class Sports
3	6			42	(e): class Cartoon

## C.5.2 For 60-second clips

1

```

Avg Shot Length (F1) <= 84.1579 :
| Avg Sat >20 (F9) > 0.913947 : Cartoon (20.0/2.8)
| Avg Sat >20 (F9) <= 0.913947 :
| | Avg Dyn SceneF(F5) > 0.004159 : Commercial (45.0/6.4)
| | Avg Dyn SceneF(F5) <= 0.004159 :
| | | Std Shot Length (F1*) <= 47.6305 : Commercial (6.0/3.5)
| | | Std Shot Length (F1*) > 47.6305 :
| | | | Avg Hist Mean12 (F7*) <= 110.779 : News (2.0/1.9)
| | | | Avg Hist Mean12 (F7*) > 110.779 : Cartoon (3.0/1.9)
Avg Shot Length (F1) > 84.1579 :
| Avg Bri Avg (F8*) > 184.367 : Cartoon (18.0/2.8)
| Avg Bri Avg (F8*) <= 184.367 :
| | Avg Camera (F3) > 1.62611 : Sports (72.0/12.9)
| | Avg Camera (F3) <= 1.62611 :
| | | Avg Sat >20 (F9) <= 0.959456 :
| | | | Avg Static Scene(F6) > 0.449473 : News (57.0/7.8)
| | | | Avg Static Scene(F6) <= 0.449473 :
| | | | | Cut Percentage (F2c) <= 0.777778 : Sports (5.0/3.3)
| | | | | Cut Percentage (F2c) > 0.777778 :
| | | | | Avg Bri Avg (F8*) <= 149.345 : News (5.0/2.3)
| | | | | Avg Bri Avg (F8*) > 149.345 : Cartoon (3.0/1.9)
| | | Avg Sat >20 (F9) > 0.959456 :
| | | | Avg Shot Length (F1) <= 145.4 : Cartoon (2.0/1.6)
| | | | Avg Shot Length (F1) > 145.4 : Sports (3.0/1.9)

```

Evaluation on training data (241 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
33	12( 5.0%)	25	15( 6.2%)	(21.0%) <<

Evaluation on test data (165 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
33	18(10.9%)	25	16( 9.7%)	(21.0%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
33					(a): class Commercial
	38		7	2	(b): class News
		3	47		(c): class Music
3	1			31	(d): class Sports
					(e): class Cartoon

## C.5.3 For 80-second clips

1

```

Avg Shot Length (F1) <= 85.2174 :
| Avg Sat >20 (F9) > 0.902825 : Cartoon (20.0/4.6)
| Avg Sat >20 (F9) <= 0.902825 :
| | Avg Shot Length (F1) <= 64.8276 :
| | | Cut Percentage (F2c) <= 0.978261 : Commercial (36.0/4.9)
| | | Cut Percentage (F2c) > 0.978261 : Cartoon (2.0/1.6)
| | Avg Shot Length (F1) > 64.8276 :
| | | Dissolve Percent(F2d) > 0.045455 : News (3.0/1.9)
| | | Dissolve Percent(F2d) <= 0.045455 :
| | | | Avg Hist Mean12 (F7*) <= 117.14 : Commercial (3.0/1.9)
| | | | Avg Hist Mean12 (F7*) > 117.14 : Cartoon (2.0/1.6)
Avg Shot Length (F1) > 85.2174 :
| Avg Bri Avg (F8*) > 181.648 : Cartoon (10.0/2.6)
| Avg Bri Avg (F8*) <= 181.648 :
| | Avg Camera (F3) > 1.85819 : Sports (48.0/9.0)
| | Avg Camera (F3) <= 1.85819 :
| | | Avg Mot Run Len(F4) <= 2.85714 :
| | | | Avg Camera (F3) > 1.5734 : Sports (2.0/1.6)
| | | | Avg Camera (F3) <= 1.5734 :
| | | | | Cut Percentage (F2c) <= 0.904762 : News (3.0/2.5)
| | | | | Cut Percentage (F2c) > 0.904762 : Cartoon (9.0/4.0)
| | | Avg Mot Run Len(F4) > 2.85714 :
| | | | PD1 lt 0.015 (22) > 0.302029 : News (45.0/5.0)
| | | | PD1 lt 0.015 (22) <= 0.302029 :
| | | | | Std Shot Length (F1*) <= 135.264 : News (11.0/5.2)
| | | | | Std Shot Length (F1*) > 135.264 : Sports (5.0/2.3)

```

Evaluation on training data (199 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
-----		-----		

41 7( 3.5%) 27 11( 5.5%) (24.3%) <<

Evaluation on test data (135 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
41	21(15.6%)	27	16(11.9%)	(24.3%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
25	1				(a): class Commercial
	39		1	3	(b): class News
					(c): class Music
	7		31		(d): class Sports
1	3			24	(e): class Cartoon

## C.6 Four Categories: Cartoons, Commercials, Music, and Sports

### C.6.1 For 40-second clips

0.8

```

Avg Shot Length (F1) <= 87.4167 :
| Avg Sat >20 (F9) <= 0.902263 :
| | Avg Static Scene(F6) <= 0.191723 :
| | | Avg Mot Run Len(F4) <= 2.25714 : Cartoon (4.0/3.0)
| | | Avg Mot Run Len(F4) > 2.25714 : Music (36.0/7.5)
| | Avg Static Scene(F6) > 0.191723 :
| | | Fade Percent (F2f) <= 0.307692 :
| | | | Cut Percentage (F2c) <= 0.952381 : Commercial (62.0/11.6)

```

```

| | | | Cut Percentage (F2c) > 0.952381 :
| | | | | Avg Shot Length (F1) <= 38 : Music (6.0/2.4)
| | | | | Avg Shot Length (F1) > 38 :
| | | | | | Std Shot Length (F1*) <= 27.4983 : Commercial (3.0/1.9)
| | | | | | Std Shot Length (F1*) > 27.4983 :
| | | | | | | Avg Static Scene(F6) <= 0.393939 : Music (3.0/1.9)
| | | | | | | Avg Static Scene(F6) > 0.393939 : Commercial (2.0/1.6)
| | | Fade Percent (F2f) > 0.307692 :
| | | | Fade Percent (F2f) <= 0.416667 : Cartoon (3.0/1.9)
| | | | Fade Percent (F2f) > 0.416667 : Music (4.0/3.0)
| Avg Sat >20 (F9) > 0.902263 :
| | Avg Bri 220 (F8) <= 0.095064 : Sports (3.0/1.9)
| | Avg Bri 220 (F8) > 0.095064 :
| | | Fade Percent (F2f) > 0.096774 : Commercial (4.0/3.5)
| | | Fade Percent (F2f) <= 0.096774 :
| | | | Std Shot Length (F1*) <= 23.9677 : Commercial (3.0/2.5)
| | | | Std Shot Length (F1*) > 23.9677 : Cartoon (34.0/2.9)
Avg Shot Length (F1) > 87.4167 :
| Avg Camera (F3) <= 0.906566 :
| | Fade Percent (F2f) > 0.083333 : Commercial (2.0/1.6)
| | Fade Percent (F2f) <= 0.083333 :
| | | Cut Percentage (F2c) <= 0.333333 : Sports (3.0/1.9)
| | | Cut Percentage (F2c) > 0.333333 : Cartoon (31.0/2.9)
| Avg Camera (F3) > 0.906566 :
| | Avg Sat >20 (F9) <= 0.721357 : Commercial (3.0/2.5)
| | Avg Sat >20 (F9) > 0.721357 :
| | | Avg Bri 220 (F8) <= 0.20916 : Sports (96.0/6.6)
| | | Avg Bri 220 (F8) > 0.20916 :
| | | | Avg Shot Length (F1) <= 122.333 : Cartoon (5.0/3.3)
| | | | Avg Shot Length (F1) > 122.333 : Sports (7.0/2.4)

```

Evaluation on training data (314 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
53	10( 3.2%)	39	18( 5.7%)	(21.2%) <<

Evaluation on test data (212 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
53	41(19.3%)	39	28(13.2%)	(21.2%)	<<
(a)	(b)	(c)	(d)	(e)	<-classified as
42		5		2	(a): class Commercial
					(b): class News
8		27		3	(c): class Music
			71	3	(d): class Sports
5		1	1	44	(e): class Cartoon

## C.6.2 For 60-second clips

1

```

Avg Shot Length (F1) <= 84.9231 :
| Avg Hist Mean12 (F7*) <= 112.724 :
| | Cut Percentage (F2c) > 0.96875 : Music (13.0/4.3)
| | Cut Percentage (F2c) <= 0.96875 :
| | | Avg Sat >20 (F9) <= 0.418216 : Music (6.0/2.4)
| | | Avg Sat >20 (F9) > 0.418216 :
| | | | Avg Sat >20 (F9) <= 0.850279 :
| | | | | Fade Percent (F2f) > 0.318182 : Music (3.0/1.9)
| | | | | Fade Percent (F2f) <= 0.318182 :
| | | | | | Avg Static Scene(F6) > 0.237697 : Commercial (37.0/2.9)
| | | | | | Avg Static Scene(F6) <= 0.237697 :
| | | | | | | Fade Percent (F2f) <= 0.038462 : Commercial (4.0/2.1)
| | | | | | | Fade Percent (F2f) > 0.038462 : Music (5.0/3.3)
| | | | | Avg Sat >20 (F9) > 0.850279 :
| | | | | | Avg Dyn SceneF(F5) <= 0.018299 : Music (5.0/2.3)
| | | | | | Avg Dyn SceneF(F5) > 0.018299 : Commercial (2.0/1.9)
| Avg Hist Mean12 (F7*) > 112.724 :
| | Avg Mot Run Len(F4) <= 3.14035 : Cartoon (32.0/6.2)
| | Avg Mot Run Len(F4) > 3.14035 :
| | | Avg Static Scene(F6) <= 0.335404 : Music (4.0/3.0)
| | | Avg Static Scene(F6) > 0.335404 : Commercial (3.0/2.5)
Avg Shot Length (F1) > 84.9231 :
| Avg Bri 220 (F8) <= 0.240119 : Sports (73.0/5.1)
| Avg Bri 220 (F8) > 0.240119 : Cartoon (19.0/5.8)

```

Evaluation on training data (206 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
29	8( 3.9%)	25	10( 4.9%)	(21.1%) <<

Evaluation on test data (142 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
29	15(10.6%)	25	15(10.6%)	(21.1%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
28		3		2	(a): class Commercial
					(b): class News
2		21		1	(c): class Music
		1	46	3	(d): class Sports
		1	2	32	(e): class Cartoon

C.6.3 For 80-second clips

1

```

Avg Shot Length (F1) <= 84.5 :
| Avg Sat >20 (F9) <= 0.902825 :
| | PD1 lt 0.015 (22) <= 0.152871 : Music (20.0/1.0)
| | PD1 lt 0.015 (22) > 0.152871 :
| | | PD1 lt 0.015 (22) > 0.789474 : Cartoon (2.0)
| | | PD1 lt 0.015 (22) <= 0.789474 :
| | | | Cut Percentage (F2c) > 0.988235 : Music (3.0)
| | | | Cut Percentage (F2c) <= 0.988235 :
| | | | | Avg Sat >50 (51) <= 0.34298 : Music (4.0/1.0)
| | | | | Avg Sat >50 (51) > 0.34298 : Commercial (39.0/1.0)
| Avg Sat >20 (F9) > 0.902825 :
| | Avg Mot Run Len(F4) <= 2.92105 : Cartoon (24.0/1.0)
| | Avg Mot Run Len(F4) > 2.92105 : Music (4.0/1.0)
Avg Shot Length (F1) > 84.5 :
| Avg Camera (F3) <= 1.0423 : Cartoon (15.0)
| Avg Camera (F3) > 1.0423 : Sports (55.0/1.0)
    
```

Evaluation on training data (166 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
17	6( 3.6%)	17	6( 3.6%)	(18.9%)	<<

Evaluation on test data (113 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
17	12(10.6%)	17	12(10.6%)	(18.9%)	<<

(a)	(b)	(c)	(d)	(e)	<-classified as
----	----	----	----	----	
23		2		1	(a): class Commercial
					(b): class News
4		17			(c): class Music
			36	2	(d): class Sports
3				25	(e): class Cartoon

## C.7 Five Categories: Cartoons, Commercials, Music, and News

### C.7.1 For 40-second clips

0.95

```

Avg Shot Length (F1) <= 83.1667 :
| Avg Sat >20 (F9) > 0.934061 : Cartoon (23.0/4.7)
| Avg Sat >20 (F9) <= 0.934061 :
| | PD1 lt 0.015 (22) <= 0.207294 : Music (48.0/14.8)
| | PD1 lt 0.015 (22) > 0.207294 :
| | | Fade Percent (F2f) > 0.454545 : Music (4.0/2.1)
| | | Fade Percent (F2f) <= 0.454545 :
| | | | Avg Hist Mean12 (F7*) <= 116.913 : Commercial (61.0/20.6)
| | | | Avg Hist Mean12 (F7*) > 116.913 :
| | | | | Avg Shot Length (F1) > 60.3333 : Cartoon (11.0/6.2)
| | | | | Avg Shot Length (F1) <= 60.3333 :
| | | | | | Avg Static Scene(F6) <= 0.365998 : Cartoon (4.0/3.0)
| | | | | | Avg Static Scene(F6) > 0.365998 : Commercial (8.0/3.8)
Avg Shot Length (F1) > 83.1667 :
| Avg Bri Avg (F8*) > 177.333 : Cartoon (20.0/2.8)
| Avg Bri Avg (F8*) <= 177.333 :
| | Fade Percent (F2f) <= 0.076923 :
| | | Avg Mot Run Len(F4) <= 2.51282 :

```

```

| | | | Avg Hist Mean12 (F7*) > 120.946 : Cartoon (16.0/4.5)
| | | | Avg Hist Mean12 (F7*) <= 120.946 :
| | | | | Avg Dyn SceneF(F5) <= 0.008929 : News (19.0/2.8)
| | | | | Avg Dyn SceneF(F5) > 0.008929 : Cartoon (3.0/1.9)
| | | Avg Mot Run Len(F4) > 2.51282 :
| | | | Avg Static Scene(F6) <= 0.210577 : Commercial (3.0/2.9)
| | | | Avg Static Scene(F6) > 0.210577 : News (92.0/9.3)
| | Fade Percent (F2f) > 0.076923 :
| | | Avg Shot Length (F1) <= 99.4545 : Commercial (5.0/2.3)
| | | Avg Shot Length (F1) > 99.4545 : News (8.0/5.6)

```

Evaluation on training data (325 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
61	19( 5.8%)	29	39(12.0%)	(26.8%) <<

Evaluation on test data (218 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
61	36(16.5%)	29	33(15.1%)	(26.8%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
41	1	7			(a): class Commercial
5	69	1	5		(b): class News
9		28	1		(c): class Music
					(d): class Sports
	4		47		(e): class Cartoon

## C.7.2 For 60-second clips

1

```

Avg Shot Length (F1) <= 84.9231 :
|   Avg Sat >20 (F9) <= 0.914861 :
|   |   Avg Sat >20 (F9) <= 0.431871 : Music (8.0/2.5)
|   |   Avg Sat >20 (F9) > 0.431871 :
|   |   |   Avg Dyn SceneF(F5) <= 0.004467 :
|   |   |   |   Avg Dyn SceneF(F5) <= 0 : Commercial (4.0/3.0)
|   |   |   |   Avg Dyn SceneF(F5) > 0 : Music (6.0/3.5)
|   |   |   Avg Dyn SceneF(F5) > 0.004467 :
|   |   |   |   Avg Hist Stdv20 (F7) > 57.8969 : Music (4.0/2.1)
|   |   |   |   Avg Hist Stdv20 (F7) <= 57.8969 :
|   |   |   |   |   Cut Percentage (F2c) <= 0.970588 : Commercial (53.0/13.8)
|   |   |   |   |   Cut Percentage (F2c) > 0.970588 : Music (9.0/4.0)
|   Avg Sat >20 (F9) > 0.914861 :
|   |   Avg Mot Run Len(F4) <= 2.75714 : Cartoon (24.0/2.8)
|   |   Avg Mot Run Len(F4) > 2.75714 : Music (3.0/1.9)
Avg Shot Length (F1) > 84.9231 :
|   Avg Bri 220 (F8) > 0.369976 : Cartoon (16.0/4.5)
|   Avg Bri 220 (F8) <= 0.369976 :
|   |   Avg Mot Run Len(F4) > 3.95 : News (49.0/2.9)
|   |   Avg Mot Run Len(F4) <= 3.95 :
|   |   |   Avg Bri 220 (F8) > 0.260165 : Cartoon (4.0/2.1)
|   |   |   Avg Bri 220 (F8) <= 0.260165 :
|   |   |   |   Avg Hist Mean12 (F7*) <= 128.831 : News (15.0/2.7)
|   |   |   |   Avg Hist Mean12 (F7*) > 128.831 :
|   |   |   |   |   Avg Mot Run Len(F4) <= 1.78125 : Cartoon (3.0/1.9)
|   |   |   |   |   Avg Mot Run Len(F4) > 1.78125 : News (4.0/3.0)

```

Evaluation on training data (202 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
-----	-----	-----	-----	-----

33 10( 5.0%) 27 13( 6.4%) (25.0%) <<

Evaluation on test data (139 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
33	14(10.1%)	27	15(10.8%)	(25.0%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
29		4			(a): class Commercial
	46		1		(b): class News
3		21			(c): class Music
					(d): class Sports
2	1	4		28	(e): class Cartoon

## C.7.3 For 80-second clips

1

```

Avg Shot Length (F1) <= 65.963 :
| Avg Sat >50 (51) > 0.840845 : Cartoon (7.0/3.7)
| Avg Sat >50 (51) <= 0.840845 :
| | PD1 lt 0.015 (22) <= 0.152871 : Music (17.0/4.5)
| | PD1 lt 0.015 (22) > 0.152871 :
| | | Cut Percentage (F2c) > 0.97619 : Music (6.0/3.5)
| | | Cut Percentage (F2c) <= 0.97619 :
| | | | Avg Sat >20 (F9) <= 0.411713 : Music (3.0/1.9)
| | | | Avg Sat >20 (F9) > 0.411713 :
| | | | | Avg Dyn SceneF(F5) > 0.005251 : Commercial (32.0/4.8)
| | | | | Avg Dyn SceneF(F5) <= 0.005251 :
| | | | | Avg Camera (F3) <= 1.34602 : Commercial (2.0/1.6)
| | | | | Avg Camera (F3) > 1.34602 : Music (4.0/2.1)
Avg Shot Length (F1) > 65.963 :
| Avg Bri Avg (F8*) > 174.527 : Cartoon (21.0/2.8)
| Avg Bri Avg (F8*) <= 174.527 :
| | Avg Mot Run Len(F4) <= 2.92105 :
| | | Avg Hist Mean12 (F7*) > 115.953 : Cartoon (13.0/4.3)
| | | Avg Hist Mean12 (F7*) <= 115.953 :
| | | | Std Shot Length (F1*) <= 39.852 : Commercial (3.0/1.9)
| | | | Std Shot Length (F1*) > 39.852 : News (5.0/3.3)
| | | Avg Mot Run Len(F4) > 2.92105 :
| | | | Avg Shot Length (F1) <= 72.1429 : Commercial (4.0/3.5)
| | | | Avg Shot Length (F1) > 72.1429 : News (57.0/2.9)

```

Evaluation on training data (174 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
27	7( 4.0%)	25	8( 4.6%)	(23.4%) <<

Evaluation on test data (118 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
27	13(11.0%)	25	12(10.2%)	(23.4%) <<

(a)	(b)	(c)	(d)	(e)	<-classified as
24		2			(a): class Commercial
1	42				(b): class News
2	1	16		2	(c): class Music
					(d): class Sports
1	3			24	(e): class Cartoon

## Appendix D

The paper submitted to  
ICPR'2000

# Bibliography

- Aigrain, P., Joly, P., and Longueville, V. (1998). Medium knowledge-based macro-segmentation of video into sequences. In M. T. Maybury, editor, *Intelligent Multimedia Retrieval*, chapter 8, pages 159–174. AAAI Press/The MIT Press.
- Alattar, A. M. (1993). Detecting and compressing dissolve regions in video sequences with a DVI multimedia image compression algorithm. In *Proceedings of 1993 IEEE International Symposium on Circuit and Systems*, pages 13–16.
- Alattar, A. M. (1997). Detecting fade regions in uncompressed video sequences. In *Proceedings of 1997 IEEE International Conference on Acoustics Speech and Signal Processing*, pages 3025–3028.
- Alattar, A. M. (1998). Wipe scene change detector for use with video compression algorithms and MPEG-7. *IEEE Transaction on Consumer Electronics*, **44**(1), 43–51.
- Antani, S., Kasturi, R., and Jain, R. (1998). Pattern recognition methods in image and video databases: Past, present and future. In *Seventh International Workshop on Structural and Symbolic Pattern Recognition and Second Workshop on Statistical Techniques in Pattern Recognition (SSPR/SPR'98)*.
- Ardebilian, M., Tu, X., and Chen, L. (1997). Improvement of shot detection methods based on dynamic threshold selection. In *Proceedings of SPIE Conference on Multimedia Storage and Archiving Systems*, volume 3229, Dallas, USA.
- Ariki, Y. and Matsuura, K. (1999). Automatic classification of TV news articles based on telop character recognition. In *Proceedings of IEEE International Conference on Multimedia and Systems*, volume 2, pages 148–152, Florence, Italy.
- Ariki, Y., Shibutani, A., and Sugiyama, Y. (1997). Classification and retrieval of TV Sports News by DCT features. In *IPSI International Symposium on Information System and Technologies for Network Society*, pages 269–272.
- Arman, F., Hsu, A., and Chiu, M.-Y. (1993). Image processing on compressed data for large video databases. In *ACM Multimedia'93*, pages 267–272.

- Colombo, C., Bimbo, A. D., and Pala, P. (1999). Retrieval of commercials by semantic content: the semiotic perspective. *Multimedia Tools and Applications*, (to appear).
- Deardorff, E., Little, T., Marshall, J., Venkatesh, D., and Walzer, R. (1994). Video scene decomposition with the motion picture parser. In *IS&T/SPIE*, volume 2187, pages 44–45.
- Effelsberg, W., Fischer, S., and Lienhart, R. (1995). Automatic recognition of film genres. In *The Third ACM International Multimedia Conference and Exhibition (MULTIMEDIA '95)*, pages 367–368, New York. ACM Press.
- Feng, J., Lo, K.-T., and Mehrpour, H. (1996). Scene change detection algorithm for MPEG video sequence. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 821–824.
- Ferman, A. and Tekalp, A. M. (1998). Efficient filtering and clustering methods for temporal video segmentation and visual summarization. *Journal of Visual Communication and Image Representation*.
- Gall, D. L. (1991). MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, **34**(4), 46–58.
- Gamaz, N., Huang, X., and Panchanathan, S. (1998). Scene change detection in MPEG domain. In *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 12–17.
- Gauch, J. M., Gauch, S., Bouix, S., and Zhu, X. (1999). Real time video scene detection and classification. *Information Processing and Management*, **35**(3), 341–400.
- Gu, L., Tsui, K., and Keightley, D. (1997). Dissolve detection in MPEG compressed video. In *Proceedings of IEEE International Conference on Intelligent Processing Systems*, volume 2, pages 1692–1696.
- Hammoud, R., Chen, L., and Fontaine, D. (1998). An extensible spatial-temporal model for semantic video segmentation. In *1st International Forum on Multimedia and Image Processing*, Anchorage, Alaska.
- Hampapur, A., Jain, R., and Weymouth, T. (1994). Digital video segmentation. In *Proceedings of the Second ACM International Conference on Multimedia (MULTIMEDIA '94)*, pages 357–364, New York. ACM Press.
- Hanjalic, A., Lagendijk, R. L., and Biemond, J. (1996). A novel video parsing method with improved thresholding. In *Third Annual Conference of the Advanced School for Computing and Imaging, ASCI'97*, , Neitherland.
- Hanjalic, A., Lagendijk, R. L., and Biemond, J. (1997). Achievements and challenges in visual search of video. In *17<sup>th</sup> Symposium on Information Theory in the BELENUX*, Heijean, Neitherland.

- Hanjalic, A., Lagendijk, R., and Biemond, J. (1999). Automated high-level movie segmentation for advanced video retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Heng, W., Ngan, K., and Lee, M. (1998). Validity of scene cut detection using bit rate information of VBR video. In *Symposium on Image, Speech, Signal Processing and Robotics'98*, volume II, pages 133–138, Hongkong.
- Iyengar, G. and Lippman, A. B. (1998). Models for automatic classification of video sequences. In *Storage and Retrieval VI*, San Jose.
- Kawashima, T., Tateyama, K., Iijima, T., and Aoki, Y. (1998). Indexing of baseball telecast for content-based video retrieval. In *IEEE 1998 International Conference on Image Processing ICIP'98*, pages 871–874, Chicago.
- Kim, H., Park, S.-F., Lee, J., King, W. M., and Song, S. M.-H. (1999). Processing of partial video data for detection of wipes. In *Proceedings of SPIE*.
- Kobla, V., Doermann, D., and Rosenfeld, A. (1996). Compressed domain video segmentation. Technical Report 839, Center for Automation Research, College Park, MD 20742-3275.
- Kobla, V., DeMenthon, D., and Doermann, D. (1999). Detection of slow-motion replay sequences for identifying sports videos. In *IEEE 1999 International Workshop on Multimedia Signal Processing*, Copenhagen, Denmark.
- Kuo, T. C., Lin, Y., L.P.Chen, A., Chen, S.-C., and Ni, C. (1996). Efficient shot change detection on compressed video. In *Proceedings of International Workshop on Multimedia Database Management Systems*, pages 101–108.
- Li, D. and Sethi, K. (1999). MDC: A software tool for developing MPEG applications. In *Proceedings of IEEE International Conference on Multimedia and Systems*, volume 1, pages 445–451, Florence, Italy.
- Lienhart, R. (1999). Comparison of automatic shot boundary detection algorithms. In *Proceedings of SPIE, Image and Video Processing VII*, volume SPIE 3656-29.
- Liu, Z., Huang, J., Wang, Y., and Chen, T. (1997). Audio feature extraction & analysis for scene classification. In *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, New Jersey.
- Liu, Z., Huang, J., and Wang, Y. (1998). Classification of TV programs based on audio information using hidden markov model. In *IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing*, pages 27–32.
- Lupatini, G., Saraceno, C., and Leonardi, R. (1998). Scene break detection: a comparison. In *Proceedings of 8th Workshop on Continuous-Media Databases and Applications*, pages 34–41.

- Meng, J., Juan, Y., and Chang, S.-F. (1995). Scene change detection in a MPEG compressed video sequence. In *IS&AT/SPIE Symposium Proceedings Vol 2419*.
- Millerson, G. (1990). *The Technique of TV Production*. Focal Press, London.
- Nagasaka, A. and Tanaka, Y. (1992). Automatic video indexing and full-motion search for object appearance. In *Proceedings of Second Working Conference on Visual Databases Systems*, pages 113–127.
- Ngo, C., Pong, T., and Chin, R. (1999). Detecting gradual transitions through temporal slice analysis. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 750–755, Colorado.
- Patel, N. V. and Sethi, I. K. (1996). Compressed video processing for cut detection. In *IEE Proceedings: Vision, Image and Signal Processing*, volume 134, pages 315–322.
- Pleiffer, S., Lienhart, R., and Effelsberg, W. (1998). Scene determination based on video and audio features. Technical Report TR20/98, Praktische Informatik IV, Mannheim University.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Mateo, California.
- Rui, Y., Huang, T. S., and Mehrotra, S. (1998). Exploring video structure beyond shots. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, pages 237–240, Austin, Texas.
- Rui, Y., Huang, T. S., and Mehrotra, S. (1999). Constructing table-of-content for videos. *ACM Multimedia System Journal*, **7**(5), 359–368.
- Sahouria, E. and Zakhor, A. (1998). Content analysis of video using principal components. In *IEEE 1998 International Conference on Image Processing ICIP'98*, volume 3, pages 541–545, Chicago.
- Saraceno, C. and Leonardi, R. (1997). Audio as a support to scene change detection and characterization of video sequence. *Lecture Notes in Computer Science*, **1311**.
- Shararay, B. (1995). Scene change detection and content-based sampling of video sequences. In *SPIE/IS&T Symposium on Electronic Imaging Science and Technology: Digital Video Compression: Algorithms and Technologies*.
- Shen, K. and Delp, E. J. (1995). A fast algorithm for video parsing using MPEG compressed sequences. In *Proceedings of IEEE International Conference on Image Processing, ICPR'95*, volume 2, pages 252–255.
- Song, S. M.-H., Kim, W. M., Kim, H., and Rhee, B.-D. (1998). On detection of gradual scene changes for parsing of video data. In *Proceedings IS&T/SPIE Storage and Retrieval for Image and Video Databases*, volume 3312, pages 404–413.

- Srinivasan, M., Venkatesh, S., and Hosie, R. (1997). Qualitative extraction of camera parameters. *Pattern Recognition*, **30**(4), 593–606.
- Sugano, M., Nakajima, Y., Yanagihara, H., and Yoneyama, A. (1998). A fast scene change detection on MPEG coding parameter domain. In *Proceedings of IEEE International Conference on Pattern Recognition, ICPR'98*, pages 889–892.
- Sun, X., Kankanhalli, M. S., Zhu, Y., and Wu, J. (1998). Content-based representative frame extraction for digital video. In *International Conference on Multimedia Computing and Systems, ICMCS'98*, pages 190–193, Austin, Texas.
- Vasconcelos, N. and Lippman, A. (1997). Towards semantically meaningful feature space for the characterization of video content. In *International Conference on Image Processing ICPR'97*, volume 1, pages 25–28, Santa Barbara, California.
- Wurtzel, A. and Rosenbaum, J. (1995). *Television Production*. McGraw-Hill, Inc.
- Xiong, W., Lee, J. C.-M., and Ma, R.-H. (1997). Automatic video data structuring through shot partitioning and key frame computing. *Machine Vision and Applications*, **10**(2), 51–65.
- Yeo, B.-L. and Liu, B. (1995a). On the extraction of DC sequence from MPEG compressed video. In *Proceedings of IEEE International Conference on Image Processing, ICPR'95*, volume 2, pages 260–263.
- Yeo, B.-L. and Liu, B. (1995b). Rapid scene analysis on compressed video. *IEEE Transaction on Circuits and Systems for Video Technology*, **2**, 533–544.
- Yeung, M., Yeo, B.-L., and Liu, B. (1998). Segmentation of video by clustering and graph analysis. *Computer Vision and Image Understanding*, **7**(1), 94–109.
- Zabih, R., Miller, J., and Mai, K. (1999). A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, **7**(2), 119–128.
- Zettl, H. (1997). *Television Production Handbook*. Wad Sworth Publishing Company, 6 edition.
- Zhang, H., Kankanhalli, A., and Smoliar, S. (1993). Automatic partitioning of full-motion video. *Multimedia System*, **1**, 10–28.
- Zhang, H., Low, C., Gong, Y., and Molliar, S. (1994). Video parsing using compressed data. In *IS&T/SPIE, Image and Video Processing II*, pages 142–149.
- Zhang, H., Tan, S. Y., Smoliar, S. W., and Yihong, G. (1995). Automatic parsing and indexing of news video. *Multimedia Systems*, **2**(6), 256–266.
- Zhang, H., Low, C. Y., Smoliar, S. W., and Wu, J. (1998). Video parsing, retrieval and browsing: An integrated and content based solution. In M. T. Maybury, editor, *Intelligent Multimedia Retrieval*, chapter 7, pages 139–158. AAAI Press/The MIT Press.

---

Zhuang, Y., Rui, Y., Huang, T. S., and Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. In *IEEE Int Conf on Image Processing*, pages 866–870, Chicago.