GUILLERMO CÁMARA CHÁVEZ

ANÁLISE DE CONTEÚDO DE VÍDEO POR MEIO DO APRENDIZADO ATIVO

Belo Horizonte
06 de julho de 2007

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

ANÁLISE DE CONTEÚDO DE VÍDEO POR MEIO DO APRENDIZADO ATIVO

Tese apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

GUILLERMO CÁMARA CHÁVEZ

Belo Horizonte
06 de julho de 2007

Federal University of Minas Gerais
Instituto de Ciências Exatas
Graduate Program in Computer Science

VIDEO CONTENT ANALYSIS BY ACTIVE LEARNING

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

GUILLERMO CÁMARA CHÁVEZ

Belo Horizonte
July 6, 2007

UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Análise de Conteúdo de Vídeo por meio do Aprendizado Ativo

GUILLERMO CÁMARA CHÁVEZ

Tese defendida e aprovada pela banca examinadora constituída por:

Prof. Doutor Arnaldo Albuquerque de Araújo – Orientador
Departamento de Ciência da Computação - ICEx - UFMG

Profa. Doutor Sylvie Philipp-Foliguet – Co-orientador
Equipe Traitement des Images et du Signal-ENSEA,
Université de Cergy-Pontoise

Prof. Doutor Matthieu Cord – Co-orientador
Laboratoire d'Informatique de Paris 6,
Université Pierre et Marie Curie

Prof Doutor Hani Camille Yehia
Departamento de Engenharia Eletrônica - DEE - UFMG

Prof. Doutor Mário Fernando Montenegro Campos
Departamento de Ciência da Computação - ICEx - UFMG

Prof. Doutor Neucimar J. Leite
Instituto de Computação - IC - UNICAMP

Prof. Doutor Zhao Liang
Instituto de Ciências Matemáticas e de Computação - USP

Belo Horizonte, 06 de julho de 2007

Resumo Estendido

Avanços em técnicas de compressão, diminuição no custo de armazenamento e transmissões em grande velocidade têm facilitado a forma como os vídeos são criados, armazenados e distribuídos. Como conseqüência, os vídeos passaram a ser utilizados em várias aplicações. Devido ao aumento na quantidade de dados dos vídeos distribuídos e usados em aplicações atuais, estes se destacam como um tipo de dado multimídia, introduzindo, porém, o requerimento de um gerenciamento mais eficiente destes dados. Tudo isto tem aberto o caminho para novas áreas de pesquisa, tais como a indexação e recuperação de vídeo baseadas no conteúdo semântico, visual e espaço-temporal.

Esta tese apresenta um trabalho dirigido à criação de um suporte unificado para a indexação semi-automática de video e recuperação iterativa. Para criar uma indexação unificada, é selecionado um conjunto de *quadros-chave* que capturam e encapsulam o conteúdo do vídeo. Isso é conseguido através da segmentação do vídeo em *tomadas* constitutivas e selecionando um número ótimo de quadros dentre os limites da tomada. Primeiro, desenvolvemos um algoritmo para segmentação automática (detecção de cortes de cena). A fim de prescindir da definição de limiares e parâmetros, utilizamos um método de classificação supervisionado. Adotamos um classificador SVM devido à habilidade para utilizar espaços de características de alta dimensão (utilizando funções de *kernels*) preservando a grande capacidade de generalização. Igualmente, avaliamos profundamente diferentes combinações de características e *kernels*. Avaliamos o desempenho do nosso classificador utilizando diferentes funções *kernel* visando encontrar aquele que apresente melhor desempenho. Nossos experimentos, seguem estritamente o protocolo da Avaliação TRECVID. Apresentamos os resultados obtidos na tarefa de detecção de cortes de cenas da Avaliação TRECVID de 2006. Os resultados obtidos foram satisfatórios lidando com um grande conjunto de características graças a nosso classificador SVM baseado em *kernels*.

O passo seguinte depois da segmentação é a extração de quadros-chave. Eles são selecionados a fim de minimizar a redundância de representação enquanto preservam o conteúdo da tomada, i.e., selecionando um número ótimo de quadros dentro dos limites da tomada. Nós propomos um sistema interativo de recuperaçao de vídeo: *RETINVID* baseano no sistem *RETIN*, uma máquina de busca e recuperação por conteúdo de imagens. O objetivo do aprendizado ativo quando utilizando em indexação é reduzir significativamente o número de quadros-chave anotados pelo usuário. Usamos o aprendizado ativo para ajudar no etiquetado semântico de bases de dados de vídeos. A abordagem de aprendizado propõe amostras de tomadas-chave do vídeo para serem anotadas e posteriormente atualizar a base de dados com as novas anotações. Logo, o sistema usa o aprendizado cumulativo adquirido para propagar as etiquetas ao resto da base de dados, este processo é executado toda vez que uma amostra de quadros-chave é apresentada ao usuário para ser anotada. As amostras de quadros-chave apresentadas são selecionadas baseadas na habilidade do sistema para incrementar o conhecimento obtido. Portanto, temos escolhido o aprendizado ativo devido à capacidade de recuperar categorias complexas, especificamente a traves do uso das funções *kernel*. A falta de dados para treinamento, categorias não-balanceadas e o tamanho do vetor de características podem ser superados através do aprendizado ativo. Avaliamos o desempenho do nosso sistema usando a base da dados utilizada na tarefa de alto-nível da Avaliação TRECVID de 2005.

Abstract

Advances in compression techniques, decreasing cost of storage, and high-speed transmission have facilitated the way videos are created, stored and distributed. As a consequence, videos are now being used in many applications areas. The increase in the amount of video data deployed and used in today's applications reveals not only the importance as multimedia data type, but also led to the requirement of efficient management of video data. This management paved the way for new research areas, such as indexing and retrieval of video with respect to their spatio-temporal, visual and semantic contents.

This thesis presents work towards a unified framework for semi-automated video indexing and interactive retrieval. To create an efficient index, a set of representative key frames are selected which capture and encapsulate the entire video content. This is achieved by, firstly, segmenting the video into its constituent shots and, secondly, selecting an optimal number of frames between the identified shot boundaries. We first developed an automatic segmentation algorithm (shot boundary detection) to get rid of parameters and thresholds, we explore a supervised classification method. We adopted a SVM classifier due to its ability to use very high dimensional feature spaces (using the kernel trick) while at the same time keeping strong generalization guarantees from a few training examples. We deeply evaluated the combination of features and kernels in the whole data set. We evaluate the performance of our classifier with different kernel functions. Our experiments, strictly following the TRECVID Evaluation protocol. We present the results obtained, for shot extraction TRECVID 2006 Task. We provide good results dealing with a large amount of features thanks to our kernel-based SVM classifier method.

The next step after segmentation is the key frame extraction. They will be selected to minimize representational redundancy whilst still portraying the content in each shot, i.e., selecting an optimal number of frames between the identified shot boundaries. We propose an interactive video retrieval system: *RETINVID* based on *RETIN* system, a content-based search engine image retrieval. The goal of active learning when applied to indexing is to significantly reduce the number of key frames annotated by the user. We use active learning to aid in the semantic labeling of video databases. The learning approach proposes sample key-frame(s) of a video to the user for annotation and updates the database with the new annotations. It then uses its accumulative knowledge to propagate the labels to the rest of the database, after which it proposes new key frames samples for the user to annotate. The sample key frames are selected based on their ability to increase the knowledge gained by the system. Therefore, we have chosen an active learning approach because of its capacity to retrieve complex categories, specifically through the use of kernel functions. The lack of training data, the unbalance of the classes and the size of the feature vectors can be overcome by active learning. We perform an experiment against the 2005 TRECVID benchmark in the high-level task.

To my parents Rodolfo and Laura

Acknowledgments

Writing this part of thesis gives me a formal opportunity to thank the people who have supported me and consequently had influence on the accomplishment of this work.

To God, for being always with me.

I am deeply indebted to my advisors, Prof. Arnaldo, Prof. Sylvie and Prof Matthieu, for offering me an opportunity. I would like to thank you for all the guiding, rewarding discussions, cooperation, encouragements, and lasting support throughout the studies.

I would also like to thank the past and present members of NPDI research group at UFMG and IMAGE research group at ETIS.

Thanks to all my friends made in Brazil and France who have always given me tremendous supports and encouragement. Specially to my friend from north and north-east of Brazil. I would also like to thank to Fred for your constant inspiration and endless faith that I could actually do this. Your advice has been invaluable.

Thanks to MUSCLE Network of Excellence, CNPq and CAPES for the financial support of this work.

Last, but most importantly, I would like to dedicate this thesis for my mum and dad to express my deepest gratitude. They are the best parents who are so willing in giving me the best in life (including education) without hoping for anything in return.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Video data is becoming increasingly important in many commercial and scientific areas with the advent of applications such as digital broadcasting, interactive-TV, video-on-demand, computer-based training, video-conferencing and multimedia processing tools, and with the development of the hardware and communications infrastructure necessary to support visual applications. The availability of bandwidth to access vast amount of video multimedia data will lead to the need for video database management techniques to allow browsing and search of video in digital libraries, such as current text databases techniques allow online browsing and keyword search. Finding methodologies to handle the temporal segmentation, storage, retrieval, searching, and browsing of digitized video data has been an active area of recent research. There are two important aspects, among many others, surrounding the development of video indexing and retrieval systems: temporal segmentation and content classification.

1.1 Temporal Segmentation

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 a single camera. In other words, it is a sequence of images acquired by a camera from the time it starts recording an action to the time it stops recording it 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. There are two different types of transitions that can occur between shots: abrupt (discontinuous) shot transitions, also referred as cuts; or gradual (continuous) shot transitions, which include camera movements (panning, tilting, zooming) and video editing special effects (fade-in, fade-out, dissolving, wiping). These transitions can be defined as follows:

- *cut:* an instantaneous change of visual content from one shot to another;

- *fade-in:* a shot gradually appears from a constant image;
- *fade-out:* a shot gradually disappears from a constant image;
- *dissolve:* the current shot fades out while the next shot fades in;
- *wipe*: the next shot is revealed by a moving boundary in the form of a line or pattern;

Detection of all the categorized transitions will segment a video sequence into its individuals shots, each representing a different time or space, ready for further higher-level processing to characterize it.

Based in people experience, after watching innumerable hours of television and/or film during their lifetime, it is possible to say that they share an implicit film/video "grammar", particulary when it comes to shot transitions. For example, a dissolve from one shot to another usually means a relatively short amount of time that has passed. Founded on this, producers use this implicit grammar with the objective to help viewers understand the video. Violating this grammar will frustrate the expectations of the viewer. The audience's perception of screen time and the rhythm of the events are influenced by the dissolve. A fade denotes the beginning or the end of a scene, episode or idea. The significance of a fade implies in a more significant change of place or time lapse than a dissolve. The cut is the simplest, most common way of moving from one shot to the next. Due to this grammar being used consistently, the most common edit effects found in video sequences are cuts, fades and dissolves. For this reason, the most of previous work and the present work focus on detecting only these types of transitions.

1.2 Content Classification

Larger number of video information repositories are becoming available every day. Indexes are essential for effective browsing, searching, and manipulation of collections of video sequences. Such indexes are central to applications such as digital libraries containing multimedia information. To support effective use of video information, and to provide to ever-changing user requirements, these indexes must be as rich and complete as possible.

Present-day commercial video search engines such as Google[1] and *Blinkx*[2] often rely on just a filename and text metadata in the form of closed captions (Google) or transcribed speech (Blinkx). This results in a disappointing performance, as quite often the visual content is not mentioned, or properly described by the associated text. The text often covers the emotion of the video, but this is highly specific for context and wears quickly. Because natural language is highly ambiguous, simply matching the exact terms given in a search often results in a set of documents that are not closely or significantly related. There are two fundamental problems:

- *polysemy*, many of the documents retrieved may use the terms that were specified in the search in a manner that is different from the way the users intended; item *synonymy*,

[1] Google Video Search 2007 [Online]. Available: http://video.google.com/

[2] Blink Video Search 2007 [Online]. Available: http://www.blinkx.tv/

> many documents may have been excluded because the documents do not contain the terms specified in the search, even though they do contain some term that has the same meaning Lancaster [1986].

In contrast to text-based video retrieval, the content-based image retrieval research community has emphasized a visual only approach. It has resulted in a wide variety of image and video search systems Flickner et al. [1995]; Gupta and Jain [1997]; Pentland et al. [1996]. A common denominator in these prototypes is that they first partition videos into a set of access units such as shots, objects or regions Deng and Manjunath [1998], and then follow the paradigm of representing video via a set of features (low-level visual information), such as color, texture, shape, layout and spatiotemporal features Al-Omari and Al-Jarrah [2005]; Shahraray and Gibbon [1997]. Initial work on content-based retrieval focused on extracting global features from an entire image. More recent work extended content extraction to region-based analysis where feature vectors are computed from segmented regions and similarity is evaluated between individual regions Jing et al. [2004]. Those features, global and/or regional, are properly indexed, according to some indexing structure, and are then used for video retrieval. Retrieval is performed by matching the features of the query object with those of videos in the database that are nearest to the query object in high-dimensional spaces, see Figure 1.1.

Query-by-example can be fruitful when users search for the same object under slightly varying circumstances and when the target images are actually available. If proper example images are unavailable, content-based image retrieval techniques are not effective at all. Moreover, users often do not understand similarity of low-level visual features. They expect semantic similarity. In other words, when searching for cars, an input image of a red car should also trigger the retrieval of yellow colored cars. The current generation of video search engines offers low-level abstractions of the data, where users seek high-level semantics. Thus, query-by-example retrieval techniques are not that effective in fulfilling the needs of the users. The main problem for any video retrieval methodology aiming for access is the semantic gap between image data representation and their interpretation by humans Smeulders et al. [2000]. Not surprisingly, the user experience with (visual only) video retrieval is one of frustration. Therefore, a new paradigm of semantics is required when aiming for access to video archives. In a quest to narrow the semantic gap, recent research efforts have concentrated on automatic detection of semantic concepts in video. The feasibility of mapping low-level (visual) features to high-level concepts was proven by pioneering work, which distinguished between concepts such as indoor and outdoor Vailaya and Jain [2000], and cityscape and landscape Vailaya et al. [1998]. The introduction of multimedia analysis, coupled with machine learning, has paved the way for generic indexing approaches Adams et al. [2003]; Fan et al. [2004]; Naphade and Huang [2001]; Snoek et al. [2006a,b, 2005].

The perceptual similarity depends upon the application, subject/user, and the context of usage. Therefore, the machine not only needs to learn the associations, but also has to learn them on-line with a user in the loop. Today's state-of-the-art Content-Based Image Retrieval uses the combination of low-level features and relevance feedback Eakins [2002]; Santini et al.

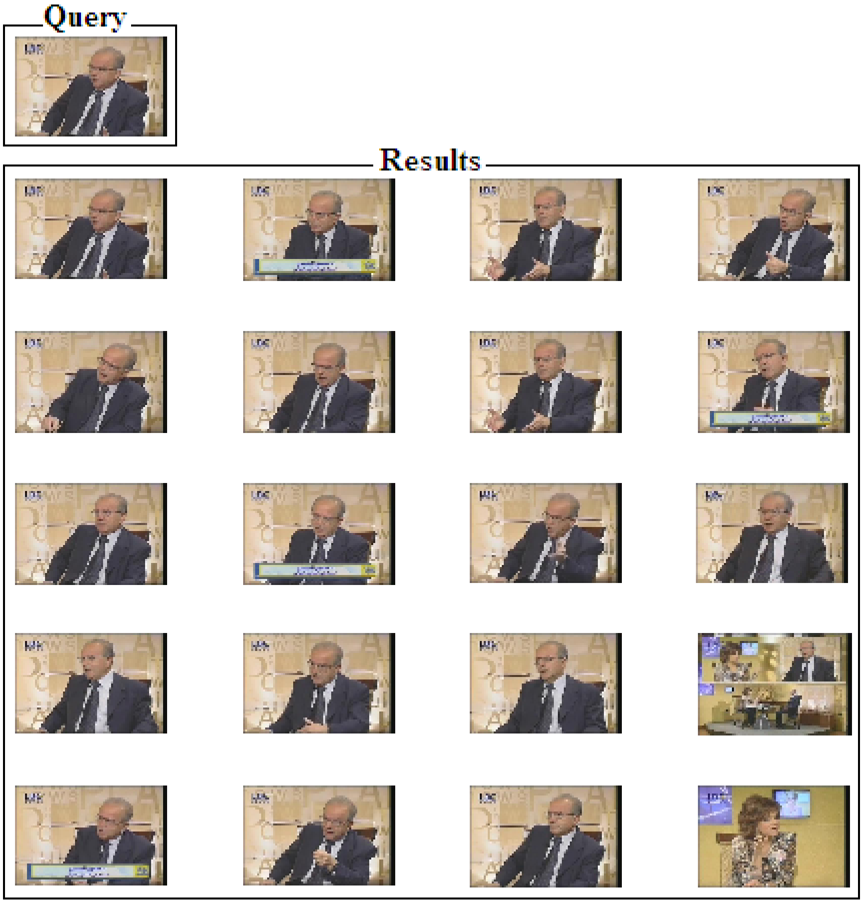


Figure 1.1: Relevant images retrieved.

[2001] to bridge the gap between low-level features and their high-level semantic meaning. Studies have shown that semantic information and relevance feedback greatly facilitate image retrieval Lu et al. [2000]. However, the old problems of labor-intensive manual annotation and subjectivity of human perception still persist. The easiest way to reduce the labeling effort is to request a human to label some selected data, and automatically propagate the labels to the entire collection using a supervised learning algorithm.

The conventional relevance feedback algorithms converge slowly because users are led to label only the most relevant documents, which is usually not informative enough for systems to improve the learned query concept model. Active learning algorithms have been proposed to speed up the convergence of the learning procedure Schohn and Cohn [2000]; Tong [2001]. In active learning, the system has access to a pool of unlabeled data and can request the user's label for a certain number of instances in the pool. However, the cost of this improvement is that users must label documents when the relevance is unclear or uncertain for the system. These "uncertain documents" are also proven to be very informative for the system to improve the learned query concept model quickly Xu et al. [2003]. Recently, active learning is being used on video analysis Qi et al. [2006]; Song et al. [2006]; Yang and Hauptmann [2006].

1.3 Aims and Objectives

The considerable amount of video data in multimedia databases requires sophisticated indices for its effective use Brunelli et al. [1999]. The most effective method for doing this task is the manual indexing, but it is slow and expensive. Thus, for this reason there is a need for automated methods to annotate video sequences and to provide a content description. Indeed, solving the problem of video segmentation (shot boundary detection) is one of the principal prerequisites for revealing video content structure in a higher level. Based on these observations, this work aims to develop an automatic technique for video segmentation and content-based retrieval.

According to Hanjalic [2002], two points are essential in relation to robustness of a shot boundary detector: an excellent detection performance for all types of shot boundaries and a constant quality of detection performance with minimized need for manual fine tuning of detection parameters in different sequences. Therefore, instead of investigating new features in which the effect of shot is used and detected, we focus on improving existing algorithms and detect automatically the shot boundaries, without setting any threshold or parameter. To cope with the problem of parameter setting, we can see video shot segmentation from a different perspective, as a categorization task. We adopt a machine learning approach to overcome this problem.

This research presents an approach to active learning for video indexing. The goal of active learning when applied to indexing is to significantly reduce the number of images annotated by the user. We use active learning to aid in the semantic labeling of video databases. The learning approach proposes sample key frame(s) of a video to the user for annotation and updates the database with the new annotations. It then uses its accumulative knowledge to propagate the labels to the rest of the database, after which it proposes new image samples for the user to annotate. The sample images are selected based on their ability to increase the knowledge gained by the system.

1.4 Contributions

The diagram in Figure 1.2 shows an automated video indexing system. The process begins segmenting temporally the video sequence into shots and selects representative key frames. Then, these key frames can be used to browse the video content or extracted features can be used to match video content to a user's query to enable shot retrieval. In Figure 1.2, we can find the main contributions highlighted in gray(blue): temporal segmentation and video indexing.

- *Video segmentation*

  1. We propose a hierarchical classification system which views temporal video segmentation as a 2-class clustering problem ("scene change" and "no scene change"). Our method consists of first detecting abrupt transitions using a learning-based ap-

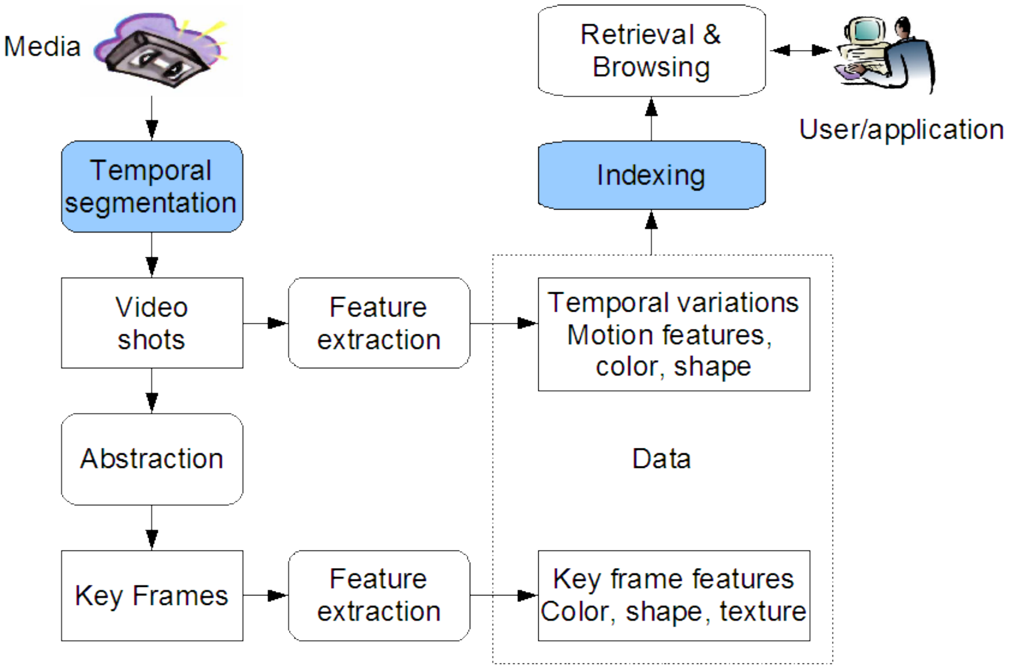


Figure 1.2: A diagram of an automated video indexing system.

proach, then non-abrupt transitions are split into gradual transitions and normal frames. Since our objective is to develop an automatic shot boundary detector we avoid to define as much as possible thresholds and parameters such as sliding windows, as Qi et al. [2003] suggest in their hierarchical system, because it is necessary to define the size of the window. Thus, our system maintains the characteristic to be parameter free.

2. Previous classification approaches consider few visual features. As a consequence of this lack, these methods need pre-processing and post-processing steps, in order to deal with illumination changes, fast moving objects and camera motion. We decided to use the well known kernel-based Support Vector Machine (SVM) classifier Cortes and Vapnik [1995] which can deal with large feature vectors. We combine a large number of visual features (color and shape) in order to avoid pre-processing and post-processing steps. Our system requires a small training set and we do not have to set any threshold or parameter.

3. We propose to use entropy as the *goodness-of-fit* measure in a block-based correlation coefficients to measure the visual content similarity between frame pairs. The entropy is applied in each block in order to describe the block information. We executed tests for abrupt transition (cut) detection and our entropy-based method, shows better performance than maximum correlation Porter et al. [2003]. This is because the entropy gives a global information of the block, instead of the information of a single element of the block.

4. Our dissolve (gradual transition) detection is based on three steps: a pattern detec-

tion based on curve matching and a refinement level based on a gradual transition modeling error, feature extraction of dissolve regions using an improved method and a learning level for classifying gradual transitions from no gradual transitions. The improved double chromatic difference is based on the work by Yu et al. [1997]. We propose a modification, reducing significatly the complexity of its computation preserving its accuracy. Indeed, we use projection histograms Trier et al. [1996] (1D) instead of the frame itself (2D).

5. We present a method for fade (gradual transition) detection based on our improved feature developed for dissolve detection. Instead of examining the *constancy of the sign* of the mean difference curve Truong et al. [2000a], we apply our improved feature (used in dissolve detection) for fade detection. Some of the techniques used for detecting fades are not tolerant to fast motion, which produces the same effect of a fade. Our feature is more tolerant to motion and other edition effects or combinations of them.

- *Video indexing*

  We propose an interactive video retrieval system: *RetinVid* based on *Retin* system, a content-based search engine image retrieval Gosselin and Cord [2006]. We have chosen an active learning approach because of its capacity to retrieve complex categories, specifically through the use of kernel functions. The lack of training data, the unbalance of the classes and the size of the feature vectors can be overcome by active learning Gosselin and Cord [2006]. We use color L^*a^*b system and *Gabor* texture features plus shape features extracted for shot boundary detection.

1.5 Thesis Outline

This thesis is organized as follows. In Chapter 2, we present the video model, basic definitions that will hold within this document. Chapter 3 provides a detailed review of previous approaches to video segmentation. Chapter 4 describes our learning-based approach for abrupt transition detection. We present the color and shape features that our system computes, also we describe the modifications that we suggest to improve the accuracy of correlation coefficients. On a large and comprehensive video data set (TRECVID[3] 2002 and 2006), the performance of proposed algorithms are compared against two other existing shot boundary detection methods in terms of precision and recall. Chapter 5 describes our learning-based approach for dissolve detection and our fade detector. We present our improvement over a widely used descriptor for dissolve detection and extend it also for fade detection. We test our system using TRECVID 2006 data set. Chapter 6 describes an interactive machine learning system for video retrieval: RetinVid. On a large and comprehensive video data set (TRECVID 2005), the performance of proposed system is compared against other retrieval methods in

[3] Trec video retrieval evaluation. Available: http://www.nlpir.nist.gov/projects/trecvid/.

terms of mean average precision (MAP, which is the area under the Precision/Recall curve). Chapter 7 concludes de thesis and provides some directions for future work.

Chapter 2

Video Model

Digital video now plays an important role in education, entertainment and other multimedia applications. It has become extremely important to develop mechanisms for processing, filtering, indexing and organizing the digital video information, hence useful knowledge can be derived from the mass information available. The two most important aspects of video are its contents and its production style Hampapur et al. [1995]. The former is the information that is being transmitted and the latter is associated with the category of a video (commercial, drama, science fiction, etc.). In this chapter, we will define some of the concepts used in literature; like shot, scene and key frame. Also, we present the most popular types of transitions (abrupt transitions, gradual transitions and camera movements) and a video database system.

2.1 Terminology

Before we go into the details of the discussion, it will be beneficial to first introduce some important terms used in the digital video research field.

Video: A video V is a sequence of frames f_t with an accompanying audio track and can be defined as $V = (f_t)_{t \in [0, T-1]}$, where T is the number of frames.

Frame: A frame has a number of discrete pixels locations and is defined by $f_t(x, y) = (r, g, b)$, where $x \in \{1 \ldots M\}$, $y \in \{1 \ldots N\}$, (x, y) represents the location of a pixel within an image, $M \times N$ represents the size of the frame and (r, g, b) represents the brightness values in the red, green and blue bands respectively.

Intensity: The intensity i of color q corresponds to its relative brightness in the sense of monochromatic gray levels.

Brightness: Brightness is defined by the *Commission Internationale de L'Ecleritage* (CIE) as the attribute of a visual sensation according to which an area appears to emit more or less light. Brightness is a perceptual quantity; it has no firm objective measure.

Frame histogram: The distinct number of values each pixel can have is discretized and a histogram is created for a frame counting the number of times each of the discrete values appears in the frame.

Feature: In image processing the concept of feature is used to denote a piece of information which is relevant for solving the computational task related to a certain application. More specifically, features can refer to

- the result of a general neighborhood operation (feature extractor or feature detector) applied to the image,
- specific structures in the image itself, ranging from simple structures such as points or edges to more complex structures such as objects.

Shot: A shot is the fundamental unit of a video, because it captures a continuous action from a single camera where camera motion and object motion is permitted. A shot represents a spatio-temporal frame sequence. This is an important concept, we will try to find the limits of shots within a video. Figure 2.1 shows the structure embedded in a video.

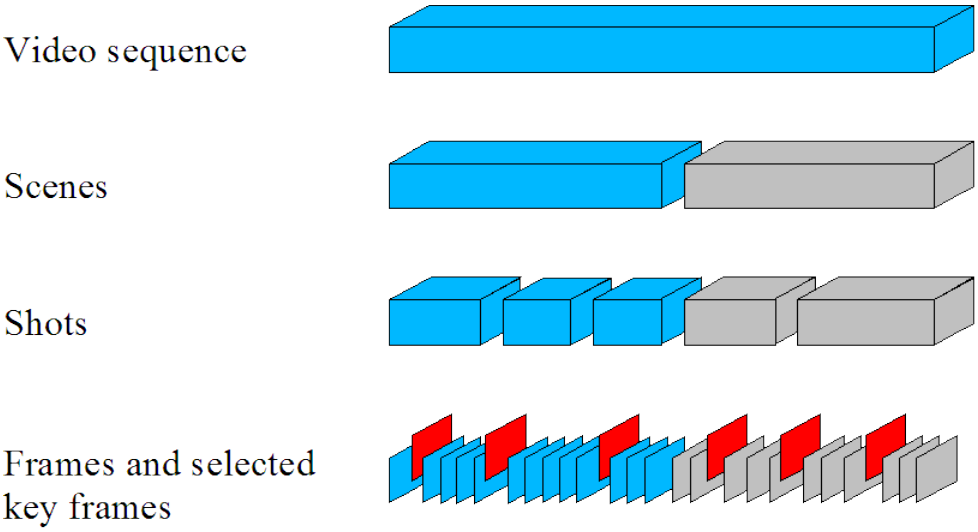


Figure 2.1: Hierarchical structure within a video sequence

Scene: A scene is composed of a small number of shots that are interrelated and unified by similar features and by temporal proximity. While a shot represents a physical video unit, a scene represents a semantic video unit.

Key frame: The frame that represents the salient visual content of a shot. Depending on the complexity of the content of the shot, one or more key frames can be extracted. This concept is also important. We will try to find the key frames that will be used later for video indexing.

The number of frames is directly associated with the frequency and the duration of visualization. In other words, we can say that a video is generated by composing several shots by a process called *editing*. This is also referred to as the *final cut* Hampapur et al. [1994].

Transition: Shots are separated by editing effects (an interruption between shots), these effects are known as transitions. The process of editing may introduce additional frames into the final cut. Different kinds of transitions separate a shot from another. There exist sharp and gradual transitions.

Edit Frame: The set of images generated during the process of editing two shots.

Scene Activity: Changes that occur in the video caused by changes that occurred in the world during the production process. For example, changes in the image sequence due to movement of objects, the camera or changes in lighting, etc.

Histogram: A histogram is obtained by splitting the range of the data into equal-sized *bins* (class-intervals), each bin representing a certain intensity value range. The histogram $H(f_t, j)$ is computed by examining each pixel in the image f_t and assigning it to a j-th bin depending on the pixel intensity. The final value of a bin is the number of pixels assigned to it.

Similarity: Similarity is a quantity that reflects the strength of relationship between two features. If the similarity between feature x and feature y is denoted by $s(x, y)$, we can measure this quantity in several ways depending on the scale of measurement (or data type) that we have.

A common similarity measure for vectorial features is the geometric distance. Many similarity measure are based on the $L_p(x, y) = (\sum_{i=0}^{k})|x_i - y_i|^p)^{1/p}$. This is also often called the Minkowski distance. For $p = 2$, this yields the Euclidean distance L_2. For $p = 1$, we get the Manhattan distance L_1.

Dissimilarity: The dissimilarity $d(x, y)$ between features i and j is also based on the notion of distance. Dissimilarity functions are supposed to be increasing the more dissimilar two points get. A common relationship between dissimilarity and similarity is define by $d(x, y) = 1 - s(x, y)$. Special cases of dissimilarity functions are *metrics*.

Metric: A metric is a dissimilarity (distance) measure that satisfies the following properties:

1. $d(x, y) \geq 0$ (non-negativity);
2. $d(x, y) = d(y, x)$ (symmetry);
3. $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality).

*Pattern*Therrier [1989]: Objects of interest are generically called patterns and may be images, printed letters or characters, signals, "states" of a system or any number of other things that one may desire to classify.

2.2 Types of Transitions

The process of video production involves shooting and edition operations. The first is for production of shots and the second one is for compilation of the different shots into a structured visual presentation Hampapur et al. [1995]. When we refer to compilation, we mean the transition between consecutive shots. Figure 2.2 shows an example of an abrupt transitions and a gradual transition.

Definition 2.1 (Transition) *A transition T_i between two consecutive shots $S_i =< \ldots, f_{s-1}, f_s >$ and $S_{i+1} =< f_t, f_{t+1}, \ldots >$ with $s < t$ is the set of frames $T_i = (f_{s+1}, \ldots, f_t)$*

For example, in Figure 2.2 $S_1 =< f_1, \ldots, f_{s_1} >$, $S_2 =< f_{t_1}, \ldots, f_{s_2} >$ and $T_1 = \emptyset$ (abrupt transition).

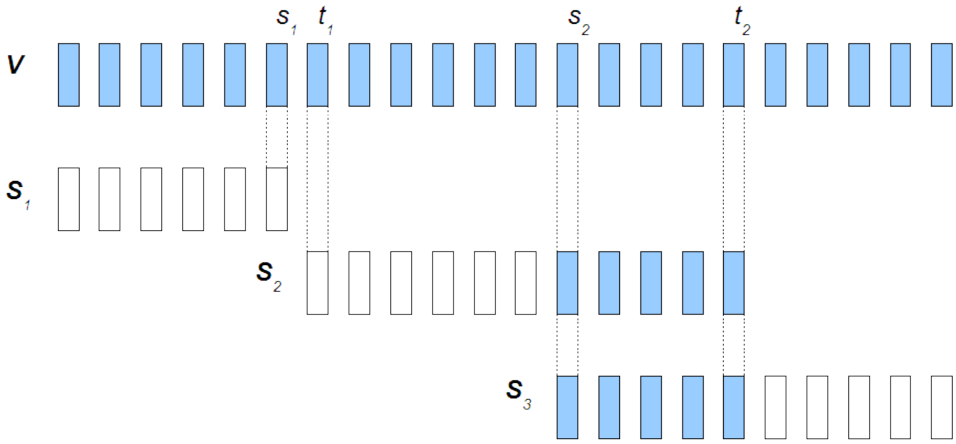


Figure 2.2: Transitions illustration from shot S_i to shot S_{i+1}.

Transitions are usually subdivided into abrupt transitions (cuts) and gradual transitions (dissolves, fades and wipes).

2.2.1 Cut

The simplest transition is the cut, and it is also the easiest transition to identify.

Definition 2.2 (Cut) *Also known as a sharp transition, a cut is characterized by the abrupt change between consecutive shots, where $t = s + 1$, as illustrated in Figure 2.2.*

We can see an example of an abrupt transition in Figure 2.3.

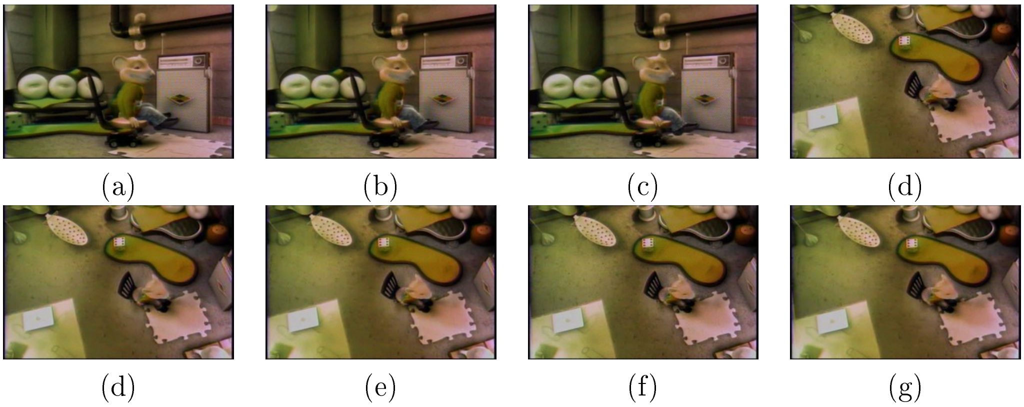


Figure 2.3: An example of a cut.

2.2.2 Fades and Dissolves

Fades and dissolves are video editing operations that make the boundary of two shots spread across a number o frames del Bimbo [1999]. Thus, they have a starting and an ending frame that identify the transition sequence. Gradual transitions occur when $t > s+1$, where the frames between the interval s and t are edited, created by a composition of the original frames.

Definition 2.3 (Fade-out) *The fade-out process is characterized by a progressive darkening of a shot S_i until the last frame becomes completely black. The frames of a fade-out can be obtained by*

$$T_{f_0}(t) = \alpha(t)G + (1-\alpha(t))S_i(t) \tag{2.1}$$

where $\alpha(t)$ is a monotonically increasing function that is usually linear, G represents the last frame, which is monochromatic (e.g. white or black) and $t \in]s_i, s_i+d[$ where d represents the duration of the fade.

Definition 2.4 (Fade-in) *The fade-in process is characterized by a progressive appearing of shot S_{i+1}. The first frame of the fade-in is a monochromatic frame G. The frames of a fade-in can be obtained by*

$$T_{f_i}(t) = (1-\alpha(t))G + \alpha(t)S_{i+1}(t) \tag{2.2}$$

where $\alpha(t)$ is a monotonically increasing function that it is usually linear. Figure 2.4 shows examples of fade-in and fade-out sequences.

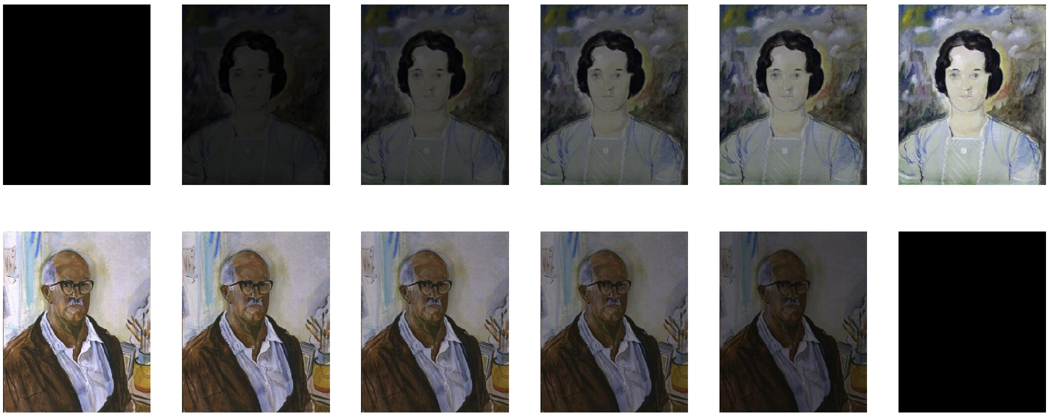


Figure 2.4: Examples of fade-in (top) and fade-out (bottom).

Definition 2.5 (Dissolve) *The dissolve is characterized by a progressive change of a shot S_i into a shot S_{i+1} with non-null duration. Each transition frame can be defined by*

$$T_d(t) = (1-\alpha(t))S_i(t) + \alpha(t)S_{i+1}(t) \tag{2.3}$$

where $\alpha(t)$ is a monotonically increasing function that it is usually linear. Figure 2.5 displays an example of dissolving.

Figure 2.6 shows examples of most used transitions, where TP is the transition period. The first transition is a cut, two shots are concatenated without inserting new edit frames.

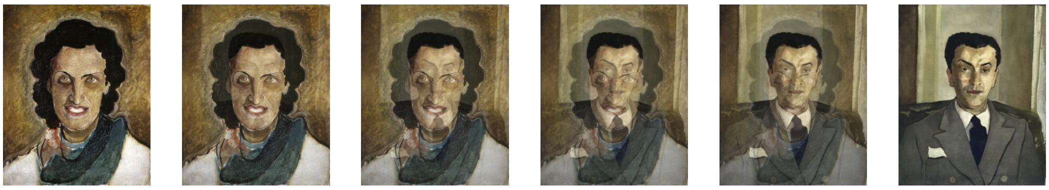


Figure 2.5: An example of dissolve.

The next transition is a fade-out, where the shot slowly get dark until it disappears. A number of "black" frames separate the fade-out from the fade-in. This transition is called fade out-in. In the case of the fade-in, the shot appears slowly from dark frames. The last transition in the figure is a dissolve, while one shot appears the other disappears.

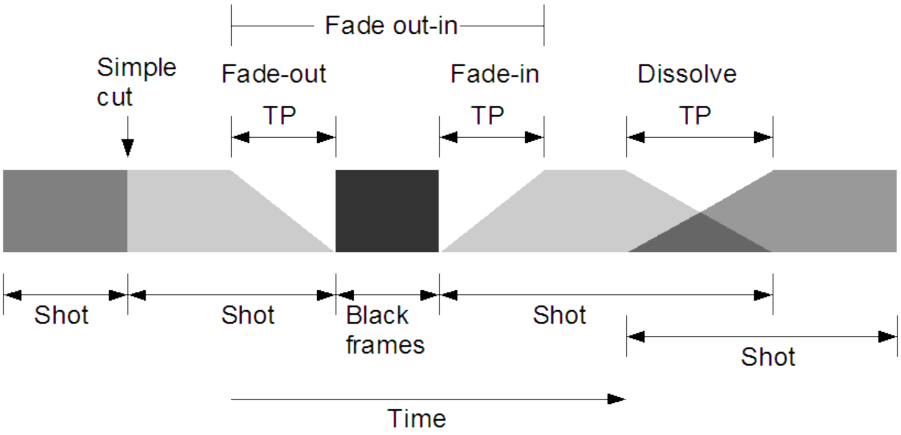


Figure 2.6: Illustration of a video sequence with shots and transitions

2.2.3 Wipe

In a wipe, one shot is (linearly, usually) replaced over time by another shot.

Definition 2.6 (Wipe) *We can model the changing characteristic of a wipe transition as*

$$T_w(t) = \left\{ \begin{array}{ll} S_i(x,y,t), & \forall (x,y) \in R_w \\ S_{i+1}(x,y,t), & \forall (x,y) \notin R_w \end{array} \right\}$$

where S_i, S_{i+1} are shots and R_w defines the uncovered wipe region, as illustrated in Figure 2.7.

Figure 2.8 displays an example of a horizontal wipe, where a "vertical line" is horizontally shifted left or right subdividing a frame in two parts.

Gradual transitions are more difficult to detect than cuts. They must be distinguished from camera operations and object movement that exhibit temporal variances of the same order and may cause false positives. It is particularly difficult to detect dissolves between

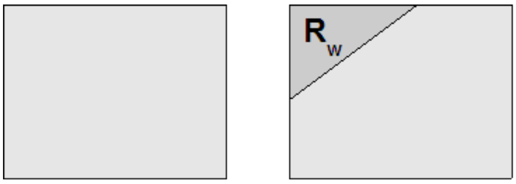


Figure 2.7: First two frames of a wipe.

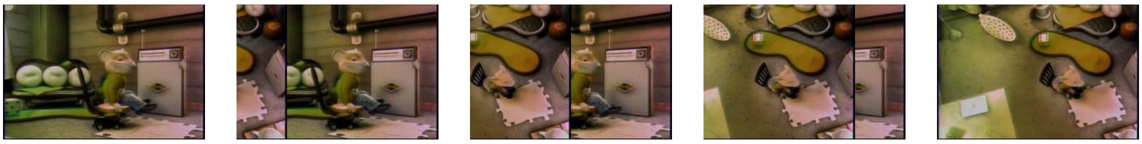


Figure 2.8: An example of a horizontal wipe.

sequences involving intensive motion Nam and Tewfik [2005]; Truong et al. [2000a]; Zabih et al. [1999].

2.3 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.

2.3.1 Camera movement

A camera can be described with a position, an orientation, and a zoom-factor. The configuration (position and orientation) of a camera can be described in a few different ways. The camera can move in five different ways (often combined). As depicted in Figure 2.9, the camera can translate, that is, move to a new position (track, boom or doll), it can rotate horizontally (pan), it can rotate vertically (tilt), and it can roll around its main axis.

Camera motion produces a global motion field across the whole image, as shown in Figure 2.10. The motion vectors in vertical and horizontal movements are typically parallel and magnitudes of motion vectors are approximately the same. In the case of zooming, the field of motion vectors has a focus of expansion (zoom in) or focus of contraction (zoom out). Most of the camera motion detection techniques are based on the analysis of the motion vector field.

2.3.2 Object Motion

Camera operation detection is based mainly in global motion detection in a frame. Object motion detection uses typically the same kind of basic algorithms but the goal is to detect regions with coherent motion witch are merged to form a moving object. Individual object tracking is a very difficult task in general. The one big problem is object occlusion. Occlusion occurs when an object is not visible in an image because some other object or structure is

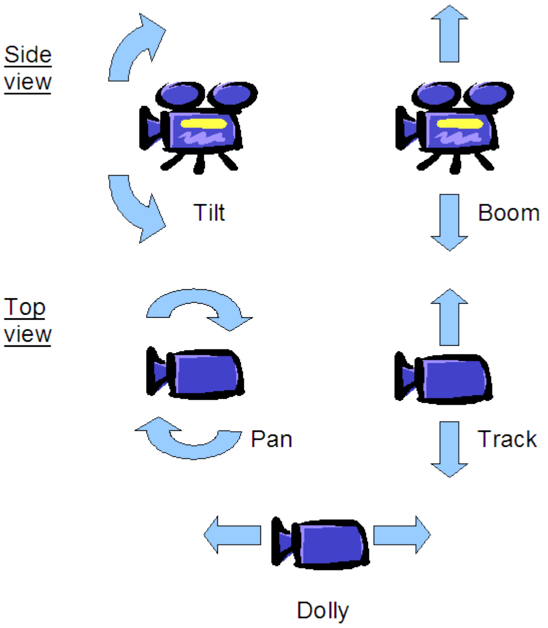


Figure 2.9: Basic camera operations: fixed, zooming (focal length change of a stationary camera), panning/tilting (camera rotation around its horizontal/vertical axis), tracking/booming (horizontal/vertical transversal movement) and dollying (horizontal lateral movement).

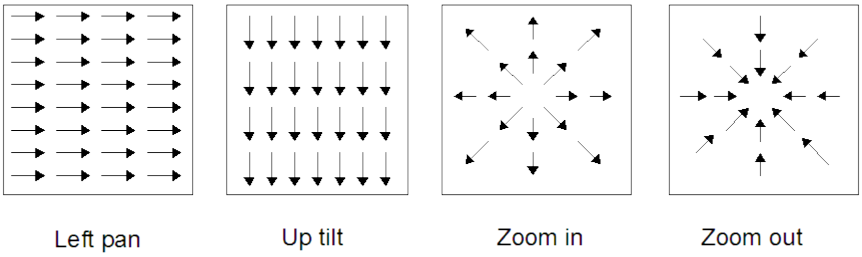


Figure 2.10: Motion vector pattern resulting from various camera operations Koprinska and Carrato [2001].

blocking its view. There are lot of studies of object tracking in literature and comprehensive study of all methods is out of scope of this work.

2.4 Video Database Systems

A video sequence is a rich multimodal Snoek and Worring [2005], Maragos [2004] information source, containing audio, speech, text (if closed caption is available), color patterns and shape of imaged object, and motion of these objects Lui et al. [1998]. Research on how to efficiently access to the video content has become increasingly active in the past years Al-Omari and Al-Jarrah [2005]; Antani et al. [2002]; Lu et al. [2000]; Zhang et al. [1997]. Considerable progress has been made in video analysis, representation, browsing, and retrieval, the four fundamental bases for accessing video content.

- *Video analysis:* deals with the signal processing part of the video system, including shot boundary detection, key frame extraction, etc.
- *Video representation:* concerns with the structure of the video. An example of video representation is the tree structured key frames hierarchy Zhang et al. [1997].
- *Video browsing:* build on the top of the video representation. Deals with how to use the representation structure to help the viewers browsing the video content.
- *Video retrieval:* concerns with retrieving interesting video objects for the viewer.

The relationship between these four research areas is illustrated in Figure 2.11. Most of the research effort has gone into video analysis since it is required in the early stages of video browsing, retrieval, genre classification, and event detection. It is a natural choice for segmenting a video into more manageable part. Though it is the basis for all the other research activities, it is not the ultimate goal. Video browsing and retrieval are on the very top of the diagram. They directly support users' access to the video content. To access a temporal medium, such as a video clip, browsing and retrieval are equally important. Browsing helps a user to quickly understand the global idea of the whole data, whereas retrieval helps a user to find a specific query's results.

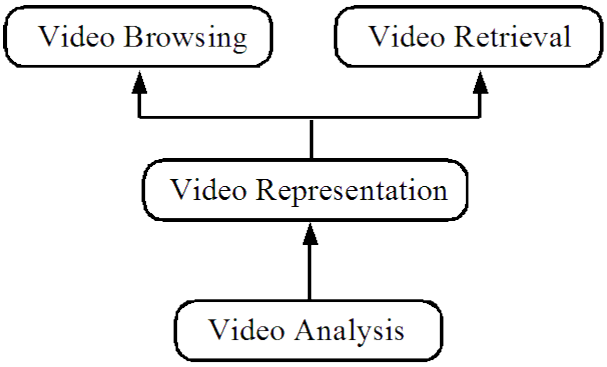


Figure 2.11: Relations between the four research areas Rui and Huang [2000b].

An analogy explains this argument. For example, the way how can a reader efficiently access the content of a book. Without needing to read the whole book, the reader can first go to the Table of Contents of the book (ToC), find which chapters or sections suit his need. If he has specific questions (queries), such as finding a key word, he can go to the Index and find the corresponding book sessions that contain that question. In resume, a ToC of a book helps a reader *browse* and the Index helps a reader *retrieve*. Both aspects are equally important for users in order to understand the content of the book. Unfortunately, current videos do not dispose a ToC and an Index. Thus, techniques are urgently needed for constructing a ToC and an Index to facilitate the video access. The scope of this work is orientated to develop an automatic technique for video analysis and video retrieval.

In the case of video retrieval, a video index is much smaller and thus easier to construct and use if it references whole video shots instead of every video frame. Shot transitions provide convenient jump points for video browsing. The detection of a shot change between two adjacent frames simply requires the computation of an appropriate continuity or similarity metric. However, this simple concept presents some major complications:

- gradual transition (GT) detection could not be based on the same assumption of abrupt transitions (high similarity between frames corresponding to the same shot and low similarity between frames corresponding to two successive shots), since similarity is also high in GT. The visual patterns of many GT are not as clearly or uniquely defined as that of abrupt transitions (AT);

- maintain a constant quality of detection performance for any arbitrary sequence, with minimized need for manual fine tuning of detection parameters in different sequences (defined parameters must work with all kind of videos);

- most of previous works in shot boundary detection consider a low number of features because of computational and classifier limitations. Then to compensate this reduced amount of information, they need pre-processing steps, like motion compensation or post-processing steps, like illuminance change filtering;

- camera or object motions may result in a sustained increase in the inter-frame difference the same as GT and cause false detection, and illuminance changes are cause of false detection in AT.

Video retrieval continues to be one of the most exciting and fastest growing research areas in the field of multimedia technology. The main challenge in video retrieval remains bridging *the semantic gap*. This means that low level features are easily measured and computed, but the starting point of the retrieval process is typically the high level query from a human. Translating or converting the question posed by a human to the low level features illustrates the problem in bridging the semantic gap. However, the semantic gap is not merely translating high level features to low level features. The essence of a semantic query is understanding the meaning behind the query. This can involve understanding both the intellectual and emotional sides of the human.

Studies have shown that semantic information and relevance feedback greatly facilitate image retrieval Lu et al. [2000]. However, the old problems of labor-intensive manual annotation and subjectivity of human perception still persist. Recently, a machine learning technique called active learning has been used to improve query performance in image retrieval systems Cord et al. [2007]; Tong and Chang [2001]. The major difference between conventional relevance feedback and active learning is that the former only selects top-ranked examples for user labeling, while the latter adopts more intelligent sampling strategies to choose informative examples from which the classifier can learn the most.

2.5 Our Propositions

There are two important aspects, among many others, surrounding the development of a video indexing and retrieval systems: temporal segmentation and content-based retrieval.

2.5.1 Segmentation

We propose an automatic machine learning approach for video segmentation, in order to overcome the parameter setting problem. Instead on investigating new features for shot boundary detection, we focus on improving existing algorithms. Our kernel-based SVM approach can efficiently deal with a large number of features with the objective to get a robust classification: better handle of illumination changes and fast movement problems, without any pre-processing step. After partitioning a video sequence into shots and detect their boundaries, we have the basis for a more complex task, like video retrieval.

2.5.2 Video Retrieval

A video retrieval system generally consists of 3 components:

- feature extraction from video frames (key frames) and an efficient representation strategy for this pre-computed data, in this stage we compute frame features and use shape features computed in video segmentation stage;
- a set of similarity measures, each one captures some perceptively meaningful definition of similarity;
- a user interface for the choice of which definition(s) of similarity should be applied to retrieval, and for the ordered and visually efficient presentation of retrieved shot videos and for supporting active learning.

2.6 Conclusion

In this chapter, we present some basic definitions that will be used in this work. These definitions let us situate in the context of temporal video segmentation and video indexing. For temporal video segmentation, first we present the definitions of principal transitions that separate two consecutive shots, then how they are detected based in the similarities of frame features. We also show some problems that affect the performance of shot boundary detections methods and present our propose to handle these problems. In the case of video indexing, we show the importance of accessing video content. Thus, techniques for video indexing are urgently needed to facilitate the video access. We present our proposal for the main challenge in video retrieval, i.e., bringing the semantic gap. We use active learning to aid in the semantic labeling of video databases.

Chapter 3

State of the Art of Video Segmentation

A vast majority of all the works published in the area of content-based video analysis and retrieval are related in one way or another with the problem of video segmentation. In this chapter we present a review of different approaches for abrupt and gradual transition detection, also known as shot boundary detection.

3.1 Introduction

Since shots are the basic temporal units of video, the shot segmentation, generally called shot boundary detection, is the groundwork of video retrieval. To fulfill the task of partitioning the video, video segmentation needs to detect the joining of two shots in the video stream and locate the position of these joins. There are two different types of these joins, abrupt transition (AT) and gradual transition (GT). According to the editing process of GTs, 99% of all edits fall into one of the following three categories: cuts, fades, or dissolves Lienhart [1999].

The basic idea of temporal segmentation is to identified the discontinuities of the visual content. No matter what kind of detection techniques, it consists of three core elements: the representation of visual content, the evaluation of visual content continuity and the classification of continuity values.

1. *Representation of visual content*: The objective is to represent the visual content of each frame f_t, this is done extracting some kind of visual features from each frame and obtain a compact content representation. The problem of content representation is to seek an appropriate feature extraction method. There are two major requirements for an appropriate content representation: *invariance* and *sensitivity.* The invariance means that the feature is stable to some forms of content variation, e.g., rotation or translation of the picture. Inversely, the sensitivity reflects the capacity of the features for capturing the details of visual content. The sensitivity is a reverse aspect of invariance. That is, the more details the feature can capture, the more sensitive it is because it can reflect

tiny changes in the visual content. With the invariance, the features within a shot stay relatively stable, while with sensitivity, the features between shots shows considerable change. Therefore, a benefic relation between invariance and sensitivity must be taken into account to achieve a satisfactory detection performance.

2. *Construction of dissimilarity signal*: the way for identifying the transitions between shots consists in first calculate the dissimilarity (distance) values of adjacent features. Thus, the visual content flow is transformed into a 1-D temporal signal. In an ideal situation, the dissimilarity within the same shot is low, while rise to high values surrounding the positions of shot transitions. Unfortunately, various disturbances such as illumination change and large object/camera motion affect the stability of temporal signal obtained by inter-frame comparison of features. In order to overcome this problem, it is important to consider not only inter-frames differences but also incorporate the variations within the neighborhood of the particular position, i.e., contextual information.

3. *Classification of dissimilarity signal*: The final critical issue is to classify the 1-D temporal signal of content variation into boundaries or nonboundaries, or identify the types of transitions. The thresholding scheme is the simplest classifier, where the threshold is the unique parameter. However, these thresholds are typically highly sensitive to the specific type of video. The main drawback of threshold-based approaches lies in detecting different kinds of transitions with a unique threshold. To cope with this problem, video shot segmentation may be seen, from a different perspective, as a categorization task. Through learning-based approaches, it is possible to eliminate the need for threshold setting and use multiple features simultaneously. Learning-based approaches could be divided in "supervised" and "unsupervised" learning. The former learns from examples provided by a knowledgable external supervisor and in the latter no teacher defines the classes *a priori*. A common problem of machine learning methods consist in deciding which features use, i.e., what combination of features are more adequate for shot boundary detection.

The three major challenges to current shot boundary detection are: the detection of GTs, the elimination of disturbances caused by abrupt illumination change and large object/camera motion.

1. *Detection of gradual transitions*: the detection of GTs remains a difficult problem. Lienhart [2001a] presents a depth analysis and find an explanation why the detection of GTs is more difficult than the detection of ATs in the perspective of the temporal and spatial interrelation of the two adjacent shots. There are three main reasons why this task is difficult. First, GTs include various special editing effects (dissolve, wipe, fade-in, fade-out, etc.). Each effect results in a distinct temporal pattern over the dissimilarity signal curve. Second, due to the wide varying lengths of GTs, the task of detecting the type and location of transitions in videos is very complex, e.g., the duration of some fast

dissolves is less than 6 frames and some fade out-in can take more than 100 frames of duration. The inter-frame difference during a GT is usually high. This makes it difficult to distinguish changes caused by a continuous edit effect from those caused by object and camera motion Finally, the temporal patterns of GTs are similar to those caused by object/camera motion, since both of them are essentially processes of gradual visual content variation.

2. *Disturbances of abrupt illumination change*: most of the methods for content representation are based on color feature, in which illumination is a basic element. Luminance changes are often detected to be AT by mistake, this occurs because of the significant discontinuity of inter-frame feature caused by the abrupt illumination change. Several illumination-invariant methods have been proposed to deal with this problem. These methods usually face a difficult dilemma, they can remove some disturbance of illumination change but with a big cost, because they also lose the information of illumination change which is critical in characterizing the variation of visual content.

3. *Disturbances of large object/camera movement*: as shot transitions, object/camera movements also conduce to the variation of visual content. Sometimes, the abrupt motion will cause similar change than the one produced by AT. In the case of persistent slow motion, they produce similar temporal patterns over the dissimilarity signal than the patterns produced by GTs. Therefore, it is difficult to distinguish the motion from the shot boundaries, since the behaviors of the content variation are similar.

With the emergence of numerous shot boundary detection approaches, several excellent surveys have been presented Boreczky and Rowe [1996], Gargi et al. [2000], Lienhart [2001b], Hanjalic [2002], Koprinska and Carrato [2001] and, Cotsaces et al. [2006]. In this chapter, we present some existing methods but focus on categorizing and analyzing them in the guide of the formal framework of chapters 4 and 5.

3.2 Methods of Visual Content Representation

The visual content of a frame can be represented by visual features extracted from it. The tradeoff between invariance and sensitivity (the two major requirements for an appropriate content representation) must be taken into account to achieve a satisfactory detection performance. Features are not only based on the extraction of image attributes, but also the difference between two successive frames is considerate as feature. A better way is to consider not only inter-frame differences but also incorporate the variations within the neighborhood of the particular position.

3.2.1 Pixel-based Methods

The simplest way to quantify the difference between two frames is to compare the intensity values of corresponding pixels. If the mean of the differences in the intensity value of the pixels is greater than a threshold, then a transition is detected. One of the first methods described in literature was from Nagasaka and Tanaka [1992]. Shot changes are detected using a simple global inter-frame difference measure. Also, they propose a shot change detection method based on pixel pair difference called template matching. For every two successive frames, differences of intensities are computed on pixels having the same spatial position in the two frames. Then, the cumulated sum of differences is compared to a fixed threshold in order to determinate if a shot change has been detected. Zhang et al. [1993] propose a pair-wise pixel comparison, the objective is to determine the percentage of pixels that have changed considerably between two frames. A pixel is deemed to have changed considerably if is greater than a given threshold. An AT is then declared present if the percentage of changed pixels is greater than a second threshold. Obviously, this is the most sensitive method, since it has captured any detail of the frame. To speed the efficiency of pixel-based methods, several methods, known as visual rhythm Chang et al. [2000]; Guimarães et al. [2003, 2004] or spatio-temporal slice Bezerra [2004]; Ngo et al. [2001] subsample the pixels from the particular positions of each frame to represent the visual content. The drawback of these methods are the number of parameters to be set. Ngo [2003] and Bezerra and Lima [2006] observed this shortcoming and propose a learning approach for the classification task of visual rhythm features in order to avoid the definition of fixed thresholds. Pixel-based approach is sensitive to object and camera motion. For example, a camera pan could cause the majority of pixels to appear significantly changed. To handle the drawbacks, several variants of pixel-based methods have been proposed. For example, Zhang et al. [1995] propose to smooth the images by a 3×3 filter before performing the pixel comparison. The average intensity measure takes the average value for each RGB component in the current frame and compares it with the values obtained for the previous and successive frames Hampapur et al. [1994]. Although less sensitive to motion than pixel-level comparisons, two shots with different color distributions can have similar average intensity values resulting in a missed detection.

Although some pixel-based methods are the simplest way to quantify the difference between two frames, they are the most sensitive methods, since they capture any detail of the frames. They are very sensitive with object and camera motion, and illuminance changing. Subsampling methods overcome these problems, reducing their impact in the accuracy of the detection.

3.2.2 Histogram-based Methods

Color histograms which capture the ratio of various color components or scales, are a popular alternative to the pixel-based methods. Since color histograms do not incorporate the spatial distribution information of various color components, they are more invariant to local or small global movements than pixel-based methods. This method is based on the assumption that two frames with a constant background and constant objects will show little difference in their corresponding histograms. This approach should be less sensitive to motion than the pixel-level comparison as it ignores changes in the spatial distribution within a frame, but herein also lies its weakness. There can exist two neighboring shots with similar histograms but entirely different content, resulting in a difference measure similar to that caused by camera and object motion. This means that it can be difficult to detect all the ATs without also incurring false detections. However, histogram approaches offer a reasonable trade-off between accuracy and computational efficiency and are the most commonly used methods in use today.

Y. Tonomura [1990] proposes a method based on gray-level histograms. Images are compared by computing a distance between their histograms. Nagasaka and Tanaka [1992] propose also a method based on gray-level histograms. However, they report that the metric is not robust in the presence of momentary noise, such as camera flashes and large object motion. A more robust measure is suggested to compare the color histograms of two frames. The authors propose using a 6 bit color code obtained by taking the two most significant bits of each RGB (Red, Green and Blue Pratt [1991]) component resulting in 64 color codes. To make the difference between two frames containing an AT be more strongly reflected they also propose using the χ^2 statistic which can be used to measure the difference between two distributions Press et al. [1988-1992]. An extensive comparison of different color spaces and frame difference measures is given in Boreczky and Rowe [1996]; Dailianas et al. [1995]; Gargi et al. [2000]. Histograms in different color spaces such RGB, HSV (Hue, Saturation and Value Foley et al. [1990]), YIQ (luminance and chrominance Pratt [1991]), L*a*b (L* present the luminance, a* correlates with redness-greenness and b* correlates with yellowness-blueness Pratt [1991]), Munsell Miyahara and Yoshida [1988] and opponent color axes Furht et al. [1995] are tested . Different comparisons as metrics have also been used as the bin-to-bin difference, χ^2 test and histogram intersection. The results show that YIQ, L^*a^*b and Munsell spaces are seen to perform well in terms of accuracy, follow by the HSV and L*u*v (luminance and chrominance Pratt [1991]) spaces and finally by RGB. Zhang et al. [1995] use a quantize color histogram, only the upper two bits of each color intensity are used to compose the color code. The comparison of the resulting 64 bins has been shown to give sufficient accuracy. Drawbacks with color histograms are the sensibility to illuminance changes, like flash lights, and the lost of spatial information, two different frames may have the same color distribution.

This approach is less sensitive to motion than pixel-based methods, because it ignores changes in the spatial distribution within a frame, but herein also lies its weakness. Two neighboring shots with similar histograms but entirely different content can cause the same effect of camera and object motion. Histogram approaches offer a reasonable relation between accuracy and computational efficiency and are the most commonly used methods in shot boundary detection.

3.2.3 Block-based Methods

A weakness of the global-level comparisons is that they can miss changes in the spatial distribution between two different shots. Yet, pixel-level comparisons lack robustness in the presence of camera and object motion. As a trade-off between both of these approaches, Zhang et al. [1993] propose the comparison of corresponding regions (blocks) in two successive frames. The blocks are compared on the basis of second-order statistical characteristics of their intensity values using the likelihood ratio. An AT is then detected if the number of blocks with a likelihood ratio is greater than a given threshold. The number of blocks required to declare an AT obviously depends on how the frame has been partitioned.

Nagasaka and Tanaka [1992] also propose dividing each frame into 4×4 regions and comparing the color histograms of corresponding regions. They also suggest that momentary noise such as camera flashes and motion usually influence less than half the frame. Based on this observation, the blocks are sorted and the 8 blocks with the largest difference values are discarded. The average of the remaining values is used to detect an AT. Ueda et al. [1991] propose an alternative approach by increasing the number of blocks to 48 and determining the difference measure between two frames as the total number of blocks with a histogram difference greater than a given threshold. This method is found to be more sensitive to detecting ATs than the previous approach Otsuji and Tonomura [1993]. Although the latter approach removes the influence of noise by eliminating the largest differences, it also reduces the difference between two frames from different shots. In contrast, Ueda's approach puts the emphasis on the blocks that change the most from one frame to another. A combination of this and the fact that the blocks are smaller makes this method more sensitive to camera and object motion Hanjalic [2002]. Demarty and Beucher [1999] split each image sequence into blocks of 20×20 pixels. A Euclidean distance is calculated between two corresponding blocks in two successive frames in order to build a grey level mask, followed by the study of the temporal evolution curve of this criterion for the whole sequence. From this curve, shot transitions are extracted by means of a morphological filter. This algorithm also gives access to a spatial model of the transition.

This highlights the problem of choosing an appropriate scale for the comparison between features relating to the visual content of two frames. Using a more local scale increases the susceptibility of an algorithm to object and camera motion, whilst using a more global scale decreases the sensitivity of an algorithm to changes in the spatial distribution.

3.2.4 Motion-based Approaches

To overcome further the problem of object and camera motion several methods have been proposed which attempt to eliminate differences between two frames caused by such motions before performing a comparison. Methods have been suggested that incorporate a block-matching process to obtain an inter-frame similarity measure based on motion Lupatini et al. [1998]; Shahraray [1995]. For each block in frame f_{t-1}, the best matching block in a neighborhood around the corresponding block in frame f_t is sought. Block-matching is performed on the image intensity values and the best matching block is chosen to be the one that maximizes the normalized correlation coefficient. The maximum correlation coefficient is then used as a measure of similarity between the two blocks.

The main distinction between these approaches is how the measures of all the blocks are combined to obtain a global match parameter. Akutsa et al. [1992] use the average of the maximum correlation coefficient for each block. This has the disadvantage of combining poor matches with good ones to obtain a passable match between two frames belonging to the same shot. Shahraray [1995] uses a non-linear digital order statistic filter. This allows the similarity values for each block to be weighted so more importance could be given to the blocks that have matched well. This improves its performance for cases when some of the blocks being compared have a high level of mismatch. The drawback of this approach is that there can exist good matches between two frames from different shots resulting in a less significant change indicating an AT. To overcome this, the authors suggest that blocks be weighted such that a number of the best matching blocks are also excluded. This suggests that the coefficients for the non-linear averaging filter must be chosen carefully when the distribution of similarity values between two frames vary greatly.

Lupatini et al. [1998] sum the motion compensated pixel difference values for each block. If this sum exceeds a given threshold between two frames an AT is declared. On the other hand, Liu et al. [2003] base their method on motion-compensated images obtained from motion vector information. A motion-compensated version of the current frame is created using the motion vectors of the previous frame. Then the motion-compensated image is normalized in order to get the same energy as the original frame. The original frame is compared to the two modified frames, motion-compensated and motion-compensated normalized, using χ^2 test Zhang et al. [1993]. The result is compared to an adaptive threshold in order to detect ATs.

Vlachos [2000] and Porter et al. [2003] use phase correlation to obtain a measure of content similarity between two frames. The latter proposes a technique inspired by motion-based algorithms. Correlation between two successive frames is computed and used as a shot change detection measure. In order to compute the inter-frame correlation, a block-based approach working in the frequency domain is taken. Frames are divided into blocks of 32×32 pixels. Every block in a frame f_{t-1} is matched with a neighbouring block in frame f_t by first computing the normalized correlation between blocks and then seeking and locating the correlation coefficient with the largest magnitude. The normalized correlation is computed in the frequency domain instead of the spatial domain to limit computation time. The average correlation is then obtained for a couple of frames. Shot changes are detected in the presence of local minima of this value. Phase correlation methods are insensitive to changes in the global illumination and lend themselves to a computationally tractable frequency domain implementation. As in the spatial domain, there can exist good matches between two frames belonging to two different shots in the frequency domain.

Finally, Fernando et al. [1999] exploit the fact that motion vectors are random in nature during an AT. The mean motion vector between two frames is determined and the Euclidean distance with respect to the mean vector calculated for all the motion vectors. If there exists an AT, the majority of motion vectors will have a large variance due to the poor correlation between the two frames. A large increase in the Euclidean distance can then be used to declare an AT. Akutsa et al. [1992]; Bouthemy et al. [1999] also exploit these characteristics.

Motion based algorithms tend to be more robust in the presence of local or global motion than frame comparison techniques. However, Yusoff et al. [1998] show that the process of computing the pixel difference can still lead to false detections in the presence of sudden intensity changes or miss detections if two shots have similar intensities.

3.2.5 Edge-based Approaches

Zarih et al. [1996] propose a method that can detect ATs and GTs like dissolves, fades and wipes. The objective is to detect the appearance of intensity edges that are distant from edges in the previous frame, i.e, when a transition occurs new intensity edges appear far from the locations of old edges. Similarly, old edges disappear far from the location of new edges. The processes needed for computing the edges change calculation are: motion compensation, edge extraction, edge change ratio and the entering and exiting edges. Although this method illustrate the viability of edge features to detect a change in the spatial decomposition between two frames, its performance is disappointing compared with simpler metrics that are less computationally expensive Dailianas et al. [1995]; Lienhart [2001b]. Lienhart [1999] compares the edge change ratio based AT detection against histogram based methods. The experiments reveal that edge change ratio usually do not outperform the simple color histogram methods, but are computationally much more expensive. Despite this depressing conclusion, the edge feature finds their applications in removing the false alarms caused by abrupt illumination change, since it is more invariant to various illumination changes than color histogram. Kim and Park [2002] and Heng and Ngan [2003] independently design flashlight detectors based on the edge feature, in which edge extraction is required only for the candidates of shot boundaries and thus the computational cost is decreased.

During a dissolve, the edges of objects gradually disappear while the edges of new objects gradually become apparent. During a fade-out the edges gradually disappear, whilst during a fade-in edge features gradually emerge. This is exploited by the edge change ratio used to detect ATs, which is extended to detect GTs as well Zabih et al. [1999].

During the first half of the dissolve the number of exiting edge pixels dominates whilst during the second half the number of entering edge pixels is larger. Similarly, during a fade-in/out the number of entering/exiting edge pixels are the most predominant. This results in an increased value in the edge change ratio for a period of time during the sequence which can be used to detect the boundaries of GTs. Although, the detection rate of GTs with this method is reported to be good, the false positive rate is usually unacceptably high Lienhart [1999]; Lupatini et al. [1998]. There are several reasons for this. The algorithm is compensated only for translational motion, meaning that zooms are a cause of false detections. Also, the registration technique only computes the dominant motion, making multiple object motions within the frame another source of false detections. Moreover, if there are strong motions before or after a cut, the cut is misclassified as a dissolve and cuts to or from a constant image are misclassified as fades.

Lienhart [1999] also used edge information to perform dissolve detection. First, edges extracted with the Canny edge detector Canny [1986] are confronted with two thresholds to determinate weak and strong edges. Then the edge-based contrast is obtained from two images, one containing the strong edges and the other the weak edges. Finally dissolves are detected when the current value edge-based is a local minimum. Yu et al. [1997] use edge information to detect GTs. ATs are first detected using a histogram difference measure computed between two successive sub-sampled frames. Then a second pass is necessary for detecting GTs. Heng and Ngan [1999] also propose a method based on edge information. They introduce the notion of edge object, considering the pixels close to the edge. Occurrences of every edge object are matched on two successive frames. Shot changes are detected using the ratio between the amount of edge objects persistent over time and the total amount of edge objects. Nam and Tewfik [1997] propose a coarse-to-fine shot change detection method based on wavelet transforms. Image sequences are first temporally sub-sampled. Frames processed are also spatially reduced using a spatial two-dimensional (2D) wavelet transform. Intensity evolution of pixels belonging to coarse frames is analyzed using a temporal one-dimensional (1D) wavelet transform. Sharp edges define possible shot change locations. Video frames around these locations are further processed at full-rate. Temporal 1D wavelet transform is applied again to the full-rate video sequence. Edge detection is also performed on every coarse frame and the number of edge points is computed on a block-based basis. Difference between two successive frames is computed using the number of edge points for each block. True shot boundaries are located on sharp edges in the 1D wavelet transform and high values of inter-frame difference considering block-based amount of edge points. Zheng et al. [2004] propose a separation method of fade-in and fade-out from object motion based on Robert edge detector. First, compute the edges using the Robert operator. Then, count the number of edges in the frame, a fade-in/fade-out is detected if there exists a frame without edge pixels. The search area is constrained by a interval bounded by two ATs.

The performance of edge-based methods are disappointing compared with other simpler methods that are less computationally expensive, e.g. several experiments reveal that edge methods usually do not outperform the simple color histogram methods. The computational cost is not only due to the process of edge detection, but also for pre-process like motion compensation. Even though there have been improvements in the detection process, the false positive rate is still high. The reasons for this is as a result of zoom camera operations (the method is compensated only for translational motion) and multiple object motions.

3.2.6 Variance-based Approach

Another method for detecting GTs is to analyze the temporal behavior of the variance of the pixel intensities in each frame. This was first proposed by Alattar [1993] but has been modified by many other authors as well Fernando et al. [2000]; Truong et al. [2000a]. It can be shown that the variance curve of an ideal dissolve has a parabolic shape, see Figure 5.2. Thus, detecting dissolves becomes a problem of detecting this pattern within the variance time series. Alattar [1993] proposes to detect the boundaries of a dissolve by detecting two large spikes in the second-order difference of this curve.

Although these models are reported to perform well, assumptions made about the behavior of an ideal transition do not generalize well to real video sequences Nam and Tewfik [2005]. The two main assumptions are: (i) the transition is linear (Eq.5.4) and (ii) there is no motion during the transition. These assumptions do not always hold for real transitions and as a result of noise and motion in the video sequences the parabolic curve is not sufficiently pronounced for reliable detection. To overcome this problem, Nam and Tewfik [2005] present a novel technique to estimate the actual transition curve by using a B-spline polynomial curve fitting technique. However, some motion contour can be well fitted by B-spline interpolation, too. Therefore, using the "goodness" of fitting to detect GTs is not so reliable. Moreover, Truong et al. [2000a] note in their study of real dissolves that the large spikes are not always obvious and instead exploit the fact that the first derivative during a dissolve should be monotonically increasing and thus they constrain the length of a potential dissolve.

Many approaches have been proposed specifically for the detection of fade transitions Lienhart [1999]; Lu et al. [1999]; Truong et al. [2000a]. They start by locating monochrome images (see Definitions 2.3 and 2.4) which are identified as frames with little or no variance of their pixel intensities. The boundaries are then detected by searching for a linear increase in the standard deviation of the pixel intensities. Lienhart [1999] reported accurate detection with this approach on a large test set.

3.3 Methods of Constructing Dissimilarity Signal

Features representing the visual content of frames, i.e., pixels, edges, motion, blocks or the whole frame are stored as scalar values, vectors, histograms or sets of vectors (it depends on the feature used). The next step for identifying the transitions between shots consists in calculating the dissimilarity values of adjacent features. Therefore, the visual content flow is transformed into a 1-D temporal signal. Various disturbances such as illumination change and large object/camera motion affect the stability of temporal signal obtained by inter-frame comparison of features. In order to overcome this problem, it is also important to incorporate the variations within the neighborhood. The existing methods can be classified into two categories according to whether they have incorporated the contextual information, i.e., two frames (pair-wise comparison) Hanjalic [2002]; Matsumoto et al. [2006]; Yu-Hsuan et al. [2006] and N-frame window (contextual information) Joyce and Liu [2006]; Nam and Tewfik [2005]. A comparison among different metrics is evaluated by Ford et al. [1997].

3.3.1 Pair-wise Comparison

The two frames measure is the simplest way to detect discontinuity between frames. The straightforward way to evaluate the continuity is to directly compare their features. In pixel-based methods it is obtained by comparing pixels between consecutive frames. With histogram methods, *L1 norm*, χ^2 test, intersection and cosine similarity have been tried to calculate and detect the discontinuity Cabedo and Bhattacharjee [1998]; Gargi et al. [2000].

The absolute bin-wise difference, also known as *L1 norm*, is the most extended metric used in shot boundary detection Ewerth et al. [2006]; Lienhart [1999]; Mas and Fernandez [2003]. The *L1 norm* between two histograms is then determined using:

$$d_t = \sum_{j=1}^{n} |H(f_t, j) - H(f_{t+1}, j)| \tag{3.1}$$

where $H(f_t, j)$ is the $j-$th bin of the histogram of the $t-$th frame. To enhance the difference between two frames across a cut, Cooper et al. [2006]; Nagasaka and Tanaka [1992] propose the use of the χ^2 test to compare the histograms. The χ test is defined as:

$$d_t = \sum_{j=1}^{n} \frac{(H(f_t, j) - H(f_{t+1}, j))^2}{H(f_t, j) + H(f_{t+1}, j)}. \tag{3.2}$$

Zhang et al. [1995] show that χ^2 test not only enhances the difference between two frames across an AT but also increases the difference due to camera and object movements. Hence, the overall performance is not necessarily better than the linear histogram comparison. Similarity can also be evaluated thanks to histogram intersection. Histogram intersection is computed using different operators, for example a min function as:

$$d_t = 1 - \frac{\sum_{j=1}^{n} min(H(f_t, j), H(f_{t+1}, j))}{\sum_{j=1}^{n} H(f_t, j)}. \tag{3.3}$$

Haering et al. [2000] threshold the histogram intersection of two consecutive frames. After that, Javed et al. [2000] propose an extension to Haering et al. [2000] method. Instead of thresholding the histogram intersection of two successive frames, they compute the difference between two successive histogram intersection values and compare this derivative to a threshold. Cabedo and Bhattacharjee [1998]; O'Toole [1998] use another measure of similarity between histograms. This measure considers the two histograms as $n-$dimensional vectors, where n is the number of bins in each histogram. This measure is related to the cosine of the angle between the two vectors. The cosine dissimilarity is defined as:

$$d_t = 1 - \frac{\sum_{j=1}^{n} (H(f_t, j) \times H(f_{t+1}, j))}{\sqrt{\sum_{j=1}^{n} (H(f_t, j)} \times \sqrt{\sum_{j=1}^{n} H(f_{t+1}, j))}}. \tag{3.4}$$

This measure outperforms other similar methods Cabedo and Bhattacharjee [1998].

In edge-based methods, the matching ratio of edge maps of the adjacent frames is used Zarih et al. [1996]. To obtain a motion independent metric, the mapping can be constructed by block matching Hanjalic [2002], it is defined as the accumulation of the continuities between the most suited block-pairs of two consecutive frames. With machine learning methods, different histogram differences are computed from consecutive frames and categorized by a classifier Ardizzone et al. [1996]; Ling et al. [1998]; Matsumoto et al. [2006].

One major drawback of the pair-wise comparison scheme is its sensitivity to noises. The approach can fail to discriminate between shot transitions and changes within the shot when there is significant variation in activity among different parts of the video or when certain shots contain events that cause brief discontinuities. There exist several techniques refining the original continuity signal to suppress the disturbances of various noises. Yuan et al. [2004] propose a so-called second-order difference method to construct the discontinuity signal. Their experiments show that the method can effectively reduce some disturbances of motion. Jun and Park [2000] propose to first smooth the original signal by a median filter, and then subtract the smoothed one from the original signal, finally obtain a clear measured signal. Actually, these techniques of refining the signal are some implicit ways of using the contextual information of the nearby temporal interval.

3.3.2 Contextual Information Scheme

The objective is to detect the discontinuity by using the features of all frames within a suitable temporal window, which is centered on the location of the potential discontinuity. Hanjalic [2002] points out that as much additional information as possible should be embedded into the shot boundary detector to effectively reduce the influence of the various disturbances. For example, not only the variation between the adjacent frames should be examined but also the variations within the temporal interval nearby should be investigated (contextual information). Recently some methods have been explicitly proposed using contextual information Cooper [2004]; Feng et al. [2005]; Qi et al. [2003]. Cooper [2004] summarizes these ideas as a similarity analysis framework to embed the contextual information. First, a similarity matrix is generated by calculating the similarities between every pair of frames in the video sequence. Next, the continuity signal is computed by correlating a small kernel function along the main diagonal of the matrix. Designing an appropriate kernel function for correlation is the critical issue within this method. Cooper performs a comparison of four kernel functions. The kernel sizes are: 4, 6, 8 and 10. Qi et al. [2003] calculate the features differences for each of 30 frame pairs between frame t and frame $t-1$, up to frame t and frame $t-30$. These window-based differences represent a frame's temporal relationship with its neighborhood. Nam and Tewfik [2005] propose a GT detection algorithm using b-splines interpolation. The authors make use of the "goodness" of fitting to determinate the occurrence of GT transition. They perform a time-localized window analysis to effectively identify the gradual change transition. A window of 1 sec. time-length (30 frames for video data of 30 frame/s) is used for dissolve/fade detection. However, some motion contour can be well fitted by B-spline interpolation, too. Therefore, using the "goodness" of fitting to detect GT is not so reliable.

One major drawback with window-based difference methods is how to determine the size of the window, there exists not a general consensus. A transition process may last more than 100 frames Yuan et al. [2004], e.g., a fade out-in transition. Note that the methods with N-frame window embeds the contextual information while constructing the continuity signal, which is different from the pair-wise comparison (two frames) scheme which incorporates contextual information by additional post-processing procedure.

3.4 Methods of Classification

Having defined a feature (or a set of features) computed from one or more frames (and, optionally, a similarity metric), a shot change detection algorithm needs to detect where these features exhibit discontinuity. This can be done in the following ways Cotsaces et al. [2006]: statistical machine learning, static thresholding and adaptive thresholding.

3.4.1 Statistical Machine Learning

There have been some recent efforts treating shot boundary detection as a pattern recognition problem and turning to the tools of machine learning. Frames are separated through their corresponding features into two classes, namely “shot change” and “no shot change”, and train a classifier to distinguish between the two classes Ngo [2003]. Through machine learning approaches we can avoid the problem of thresholds and parameters setting, which is a difficult task and depends on the type of the input video. We can merge different features in order to improve the accuracy of the detector, we do not need to set a threshold for each type of feature. All these parameters are found by the classifier. Recently, works on shot boundary detection exploit the advantages that machine learning approaches provide. In this section some works done on shot boundary detection using *supervised* and *unsupervised* learning are seen briefly.

Supervised Learning

In supervised learning, classifiers are trained and tested on a set of sample and test data. The classifier creates its own internal rules on the cases that are presented. The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output). To achieve this, the learner has to generalize from the presented data to unseen situations in a “reasonable” way.

Various approaches, including Support Vector Machines (SVM) Feng et al. [2005]; Matsumoto et al. [2006]; Ngo [2003], k-Nearest Neighbor algorithm(kNN) Cooper [2004]; Cooper et al. [2005], and neural networks Lienhart [2001b] have been employed to perform shot boundary detection. With the statistical machine learning methods, the parameters of the models are chosen via cross validation processes and the shapes of decision boundaries are constructed automatically during the training procedure. One difficulty that machine learning methods have to face is how to construct the features for the classifiers. Cooper [2004]; Cooper and Foote [2001] and the FXPAL Cooper et al. [2005] system use dissimilarity features within the particular temporal interval as the input for kNN and Yuan et al. [2005] use a SVM classifier. Similarly, Feng et al. [2005] use features within a sliding window as the features of SVM. Ngo [2003] proposes a dissolve pattern descriptor based on temporal slices. Potential dissolves are selected by cut detection in low-resolution space and classified by SVM system. Qi et al. [2003] propose a hierarchical approach with a sliding window, one level for AT detection and second level for GT detection. They compare the performance of several binary classifiers: kNN, the Naïve Bayes probabilistic classification Jain et al. [2000] and SVM. They combine different classifiers for the two different stages, ATs and GTs detection. The one that has the best performance use kNN for both stages. Another problem that machine learning methods for shot boundary detection has to face is how to obtain a well-chosen training set with relatively balance positive and negative examples, since within each video sequence the number of negative examples usually significantly exceeds that of positive examples. Lienhart [2001a] uses a dissolve synthesizer to create an infinite amount of dissolve examples and produce the non-dissolve pattern set by means of so called bootstrap method. Chua et al. [2003] and Yuan et al. [2005] adopt the active learning strategy to handle the unbalance training data.

Unsupervised Learning

In the case of unsupervised learning, no teacher defines the classes *a priori*. Thus, the system itself must find some way of clustering the objects into classes, and also find descriptions for these classes. The resulting rules from such a system will be a summary of some properties of the objects in the database: which classes are present and what discerns them. This will of course only be what the system has found as most prominent, but there may be many other ways of dividing the objects into classes, and many ways of describing each class. While in supervised learning the patterns are known in the sample and need to be generalized, in unsupervised learning the patterns are not known.

Gunsel et al. [1998] and Ewerth and Freisleben [2004] propose an unsupervised method for temporal video segmentation and shot classification. The classification is based on 2-class clustering ("scene change" or "no scene change") and the well-known K-means algorithm Pappas [1992] is used to cluster frame dissimilarities. Gao and Tang [2002] argue that a clear distinction between the two classes can not be made and suggest a fuzzy c-means algorithm. However, in the end the representatives of the "fuzzy" set must be assigned to one of the classes "cut" and "non-cut". As a limitation we can note that the approach is not able to recognize the type of the GTs. Ferman and Tekalp [1998] incorporate two features in the clustering method. Lee et al. [2006] propose a method using an improved Artificial Resonance Theory (ART2) neural network G. Carpenter [1987] for scene change detection.

3.4.2 Static Thresholding

This is the most basic decision method, which involves comparing a metric expressing the similarity or dissimilarity of the features computed on adjacent frames against a fixed threshold. In early work, heuristically chosen global thresholds were used Gargi et al. [1995]; Lienhart et al. [1997a]. Zhang et al. [1993] propose a statistical approach for determining the threshold, based on measure mean value μ and standard deviation σ of frame-to-frame differences. The threshold T is determined as $T = \mu + \alpha\sigma$. They also suggest that α should have values between 5 and 6. Even the most robust techniques fail when there is a lot of camera movement in the shot. In severe cases, nearly every frame in a video stream could be marked as a cut when objects move significantly and/or the camera operation changes very quickly. Many methods Cernekova et al. [2006]; Guimarães et al. [2003]; Nam and Tewfik [2005]; Qian et al. [2006] use a prefixed threshold for detecting the transitions, the values of the thresholds are set empirically. Static thresholding only performs well if video content exhibits similar characteristics over time and only if the threshold is manually adjusted for each video. The main drawback of these approaches lies in selecting an appropriate threshold for different kind of videos.

3.4.3 Adaptive Thresholding

The obvious solution to the problems of static thresholding is to use a sliding window and computing the threshold locally within the sliding window Cernekova et al. [2006]; Robles et al. [2004]; Truong et al. [2000a]. A much better alternative is to work with adaptive thresholds, which incorporate the contextual information taking into account the local activity of the content. 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. Ewerth and Freisleben [2004] select the threshold based on the second largest value within the window. Hanjalic [2002] combines sliding-window approach and general statistical models for the frame-to-frame difference curve to detect hard cuts. Osian and Gool [2004] analyze the value of the differences in a sliding window of 15-20 frames and compute several statistical parameters. The evaluated difference must be higher than a fixed minimum threshold and larger than a multiple of the average difference (computed over the entire window). The multiplication coefficient is proportional to the variance within the window. There is an exception from the previous criterion when the average difference of the previous frames is very high and the average difference of the next frames is very low or vice versa because current difference delimits a high activity shot from a low activity one. Urhan et al. [2006] combine global and local thresholds. If the dissimilarity is below a global threshold, they compute a local threshold based on the average differences within the window. Cernekova et al. [2006] use a local average difference within a sliding window without considering the current frame difference that is evaluated. The ratio between average difference and current difference is then compared to a threshold in order to detect the peaks that correspond to the ATs. Adaptive threshold shows better performance than global thresholding scheme Hanjalic [2002]. Related surveys with discussions on thresholding scheme can be found in Hanjalic [2002]; Lienhart [2001b]. With adaptive threshold the problem of the threshold tuning is changed by the decision of the size of the sliding window, thus one problem changes to another.

In threshold-based methods the decision boundaries are actually manually designed, which requires the developers to be familiar with the characteristics of various types of videos.

3.5 Methods of Gradual Transition Detection

ATs are based on the fact that there is a big difference between the frames across a cut that results in a high peak in the frame-to-frame difference and can be easily detected using one threshold. However, such one-threshold based approaches are not suitable to detect GTs. Although during a GT the frame to frame differences are usually higher than those within a shot, they are much smaller than the differences in the case of AT and can not be detected with the same threshold. On the other hand, object and camera motions might entail bigger differences than the gradual transition. Hence, lowering the threshold will increase the number of false positives. For relatively comprehensive surveys refer to Lienhart [2001b] and Hanjalic [2002]. In the following we present an overview of the existing methods:

1. Fade out-in: during the fade out-in, two adjacent shots are spatially and temporally well separated by some monochrome frames Lienhart [2001a], where as monochrome frames not often appear elsewhere. Lienhart [1999] proposes to first locate all monochrome frames as the candidates of fade out-in transitions. Thus, the key of the fade out-in detection is the recognition of monochrome frames. For this purpose, the mean and the standard deviation of pixel intensities are commonly adopted to represent the visual content. The effectiveness of monochrome frame detection has been reported in Cao et al. [2006]; Lienhart [2001b]; Truong et al. [2000a], and Bezerra and Leite [2007]. The latter use visual rhythm for detecting fade transitions. They consider a slice as a set of strings which may be matched using the longest common sequence (string comparison metric) Navarro [2001]. The segmentation technique for detecting patterns representing transitions is based on morphological, topological and discrete geometry. This segmentation technique is applied to the longest common sequence signal, finally the transition is detected by k-means clustering algorithm. Guimarães et al. [2003] use a similar approach, but instead of the longest common sequence signal and k-means algorithm, they use directly the image formed by slices and detect inclined edges using morphological geometry and line approximation. This method does not detect fade out-in transition as a compound set.

2. Wipe: For wipes, the adjacent shots are not temporally separated but spatially well separated at any time Lienhart [2001a]. One common method of wipe detection involves extracting and counting edges in the image; this statistic will monotonically change during a transition, from the old shot's value to the new shot's value Yu et al. [1997]. An interesting method for wipe detection is the so-called spatio-temporal slice analysis Ngo et al. [1999] and visual rhythm Bezerra and Leite [2007]. For various styles of wipes, there are corresponding patterns on the spatio-temporal slices. Based on this observation, Ngo et al. [2005] transform the detection of wipes to the recognition of the specific patterns on spatio-temporal slices. Bezerra and Leite [2007] propose a new metric *maximum matching distance*, derived from the longest common sequence. This metric gives information of pattern translations instead of measured similarity in order to discriminate motion from wipes. K-means algorithm is used for detecting wipe transitions. Other wipe detection methods such as Naci and Hanjalic [2005] are also based on the fact that two adjacent shots before and after wipes are spatially well separated at any time.

3. Dissolve: During dissolve transition two adjacent shots are temporally as well as spatially combined Lienhart [2001a]. A popular dissolve detection method is based on the characteristic of the change of intensities variance, i.e., the so-called downwards-parabolic pattern, which was originally proposed by Alattar [1993]. A drawback of this

method is that motion produces the same pattern as dissolves. Several improvements on this idea can be found in Yu et al. [1997] and Truong et al. [2000b]. Yu et al. [1997] propose a verification process, named double chromatic difference, among all candidate regions extracted using the method of Alattar [1993]. Through this verification process it is possible to separate downwards-parabolic pattern produced by motion from the ones produced by real dissolves. A method using visual rhythm and machine learning is also proposed Ngo [2003]. The method consists in reducing the temporal resolution of a slice. When different dissolve arrives at different multi-resolution, they gradually become AT depending on their temporal length. Then the strategy is to detect ATs at the low resolution space. After detecting the transitions, the AT boundaries are projected back to the original scale. They compute Gabor wavelet features from projected regions through a support window. However, regions with fast camera and object motion also appear as ATs. Thus, a SVM classifier is used to filter false matches and retaining the correct dissolves.

4. General approaches for GTs: With global color feature adopted, various types of GTs such as wipes and dissolves exhibit similar characteristics over the continuity signal curve. Therefore, it is possible to develop a unified technique to detect several types of GTs simultaneously. For example, the well-known twin-comparison technique, proposed by Zhang et al. [1993], is a general approach to detect GTs. The twin-comparison algorithm uses two threshold values, the first, the higher, is used to detect AT and the second, lower threshold is used to detect GTs. Nevertheless, it often truncates the long GTs because of the mechanism of the global low threshold. In addition, it has difficulties in reducing the disturbances of camera and object motion. To overcome the shortcomings, Zheng et al. [2005] propose an enhanced twin-comparison method, i.e., finite state automata method, in which motion-based adaptive threshold is utilized. This method yields the best performance of GT detection on the benchmark of TRECVID 2004. Different from ATs, GTs extend across varying temporal duration, which makes it difficult for a single fixed scale transition detector to detect all the GTs. The success of the twin-comparison based methods is somewhat due to the exploitation of the multi-resolution property of GTs, i.e., low threshold for high resolution and high threshold for low resolution. Several other methods have been proposed in the form of explicit temporal multi-resolution analysis. Lin et al. [2000] and Chua et al. [2003] exploit the multi-resolution edge phenomenon in the feature space and design a temporal multi-resolution analysis based algorithm which uses Canny wavelets (first order derivative of the Gaussian function) to perform temporal video segmentation. The experimental results show that the method could locates ATs and GTs in a unified framework. However, as noted by the author, the Canny wavelet transform is computationally intensive. Another multi-resolution idea is to adjust the sample rate of the video. For example, Lienhart [2001a] employs a fixed scale transition detector to run on sequences of different resolutions to detect dissolves. Similarly, Ngo [2003] reduced the problem of dissolve

detection to an AT detection problem in a multi-resolution representation and machine learning classification. Other machine learning approaches are proposed by Feng et al. [2005]; Gunsel et al. [1998]; Lee et al. [2006]; Qi et al. [2003]. A limitation of *general approaches for GTs* is that these methods are not able to recognize the type of the GT.

3.6 Conclusion

We present in this chapter a general overview of the principal approaches for shot boundary detection. Different approaches were studied, like threshold-based methods and learning-based methods. Many factors influence the performance of a shot boundary detection method. In the case of static threshold-based methods, there are many drawbacks: parameters are set empirically, do not work well for different kinds of videos and combination of different features is a difficult task because is necessary to set thresholds for each type of feature. Adaptive thresholds try to overcome the problem of threshold setting computing thresholds inside a sliding window, i.e., changing a global threshold (static threshold methods) for local thresholds. Unfortunately, adaptive threshold methods change one problem for other, because now they have to set the size of the sliding window. The size of the sliding window is crucial for the performance of the detector. Some methods use pre-processing and post-processing operations to overcome problems like abrupt luminance changes and motion compensation.

In order to overcome all these problems, a machine learning approach can handle the problem of threshold and parameter setting. Other characteristic of learning methods is that it is possible to combine different features, i.e., combine features that make the detection more robust since weakness of some features are compensated by strongness of other features. It is also possible to avoid pre-process and post-process operations, e.g., use illumination invariant features to avoid flash filtering process. On the other hand, the data available is unbalance, i.e., the number of negative examples are much bigger than positive examples. We can handle this problem using a SVM classifier which has an excellent generalization. Therefore, through a SVM-based method we can handle the problems of threshold and parameter setting, combinations of features, pre-processing and post-processing operations and unbalanced data.

Chapter 4

Abrupt Transition Detection

In this work, we focus on the exploitation of features based on frame differences (histograms, projection histograms, Fourier-Mellin moments, phase correlation method, etc.). After the feature extraction step, these features are classified by *Support Vector Machines*. Most of previous works consider a low number of features because of computational and classifier limitations. Then to compensate this reduced amount of information, they need pre-processing steps, like motion compensation. Our kernel-based SVM approach can efficiently deal with a large number of features in order to get a robust classification: better handle of illumination changes and fast movement problems, without any pre-processing step.

4.1 Introduction

In recent years, methods for automatic shot boundary detection received considerable attention due to many practical applications. For example, in video databases the isolation of shots is of interest because the shot level organization of video documents is considered most appropriate for video browsing and content-based retrieval. Shots also provide a convenient level for the study of styles of different filmmakers. Moreover, other research areas can profit considerably from successful automation of shot boundary detection processes as well. A good example is the area of video restoration. There, the restoration efficiency can be improved by comparing each shot with previous ones, if a similar previous shot in terms of visual characteristics is found, restoration settings already used before can be adopted.

For the processes of high-level video content analysis, fulfilling of the aforementioned criteria by the shot boundary detector has even a larger importance. First, bad detection performance may negatively influence the performance of subsequent high-level video analysis modules (e.g., movie segmentation into episodes, movie abstraction, broadcast news segmentation into reports). Second, if we cannot expect a video restoration/coloring operator (expert) to adjust the shot boundary detector settings to different sequences, this can be expected even less from a nonprofessional user of commercial video retrieval equipment.

The isolation of shot in a video is relatively easy when the transition from one shot to another consist of ATs. The development of shot boundary detection algorithms was initiated some decades ago with the intention of detecting ATs in video sequences. The aim of any AT detection method is to select some feature related to the visual content of a video such that:

- any frames within the same shot exhibit similar properties, and
- frames belonging to different shots would have dissimilar feature characteristics.

The basis of detecting shot boundaries in video sequences is the fact that frames surrounding a boundary generally display a significant change in their visual contents. The detection process is the recognition of considerable discontinuities in the visual-content flow of a video sequence. Figure 4.1 illustrates a general framework for AT detection. In the first stage of this process different visual features (color, shape, texture, etc.) are extracted in order to describe the content of each frame (*feature extraction*). Most of the existing methods use some inter-frame difference metric, i.e., the metric is used to quantify the feature variation from frame t to frame $t + l$, with l being the inter-frame distance (skip) and $l \geq 1$. This *dissimilarity computation* is executed in the second stage of the AT detection. The discontinuity value is the magnitude of this variation and serves as an input into the *detector*. There, it is compared against a threshold. If the threshold is exceeded, a shot boundary is detected.

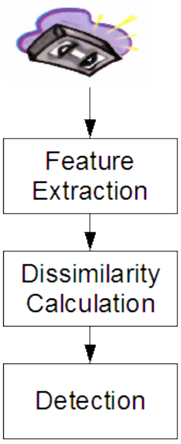


Figure 4.1: General framework for AT detection.

To be able to draw reliable conclusions about the presence or absence of a shot boundary between frames f_t and f_{t+l}, we need to use the features and metrics for computing the discontinuity values that are as discriminating as possible. This means that a clear separation should exist between discontinuity-value ranges for measurements performed within shots and at shot boundaries. There are mainly two factors that influences in the accuracy of the detector: object/camera motion and lighting changes. These two factors are cause of false detections.

We follow the same stages in our AT detector:

1. Feature extraction: we consider different visual features like color histograms in different color spaces, shape descriptors like moments and other features. We present the features used in our detector in Section 4.2;

2. Dissimilarity measures: a pair-wise dissimilarity is performed in this stage. We evaluate the dissimilarity measures applied for matching visual information in Section 4.3;

3. Detection: a machine learning approach is presented in the Section 4.4. We adopt the machine learning approach in order to avoid the setting of parameters and thresholds. Also this approach let us combine different features in order to get a more robust detector.

We test our detector with TRECVID data sets of 2002 and 2006. The first data set (2002) was used to study the different features and dissimilarity measures adopted in our work. The second data set (2006) was used to compare the performance of our method with other methods. These results are presented in Section 4.5. Finally, we discuss our conclusion in Section 4.6.

4.2 Visual Features

Automatic detection is based on the information that is extracted from the shots which can tell us when an AT occurs (brightness, color distribution change, motion, edges, etc.). It is easy to detect ATs between shots with little motion and constant illumination, this is done by looking for sharp brightness changes. In the presence of continuous object motion, or camera movements, or change of illumination in the shot, it is difficult to understand when the brightness changes are due to these conditions or to the transition from one shot to another. Thus, it is necessary to use different visual features to avoid this kind of problems. In the next subsections we will review some visual features used for shot boundary detection.

4.2.1 Color Histogram

The color histogram-based shot boundary detection algorithm is one of the most reliable variants of histogram-based detection algorithms. Its basic idea is that the color content does not change rapidly within but across shots. Thus, ATs and other short-lasting transitions can be detected as single peaks in the time series of the differences between color histograms of contiguous frames or of frames a certain distance l apart.

Let $f(x,y)$ be a color image (frame) of size $M \times N$, which consists of three channels $f = (I_1, I_2, I_3)$, the color histogram used here is:

$$h_c(m,t) = \tfrac{1}{M \times N} \textstyle\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \begin{cases} 1 & \text{if} \quad f(x,y) \text{ in bin } m \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Histograms are invariant to image rotation and change slowly under the variations of viewing angle and scale Swain [1993]. As a disadvantage one can note that two images with similar histograms may have completely different content. However, the probability for such events is low enough, moreover techniques for dealing with this problem have already been proposed in Pass and Zabih [1999].

4.2.2 Color Moments

The basis of color moments lays in the assumption that the distribution of color in an image can be interpreted as a probability distribution. Probability distributions are characterized by a number of unique moments (e.g. Normal distributions are differentiated by their mean and variance). If the color in an image follows a certain probability distribution, the moments of that distribution can then be used as features to characterize that image, based on color information.

Color moments have been successfully used in many retrieval systems. The *first order (mean)*, the *second (variance)* and the *third order (skewness)* color moments have proven to be efficient and effective in representing color distributions of images Feng et al. [2003]. The first three order moments are calculated as:

$$\mu_t = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} f_t(i,j) \tag{4.2}$$

$$\sigma_t = \left(\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_t(i,j)-\mu_t)^2\right)^{\frac{1}{2}} \tag{4.3}$$

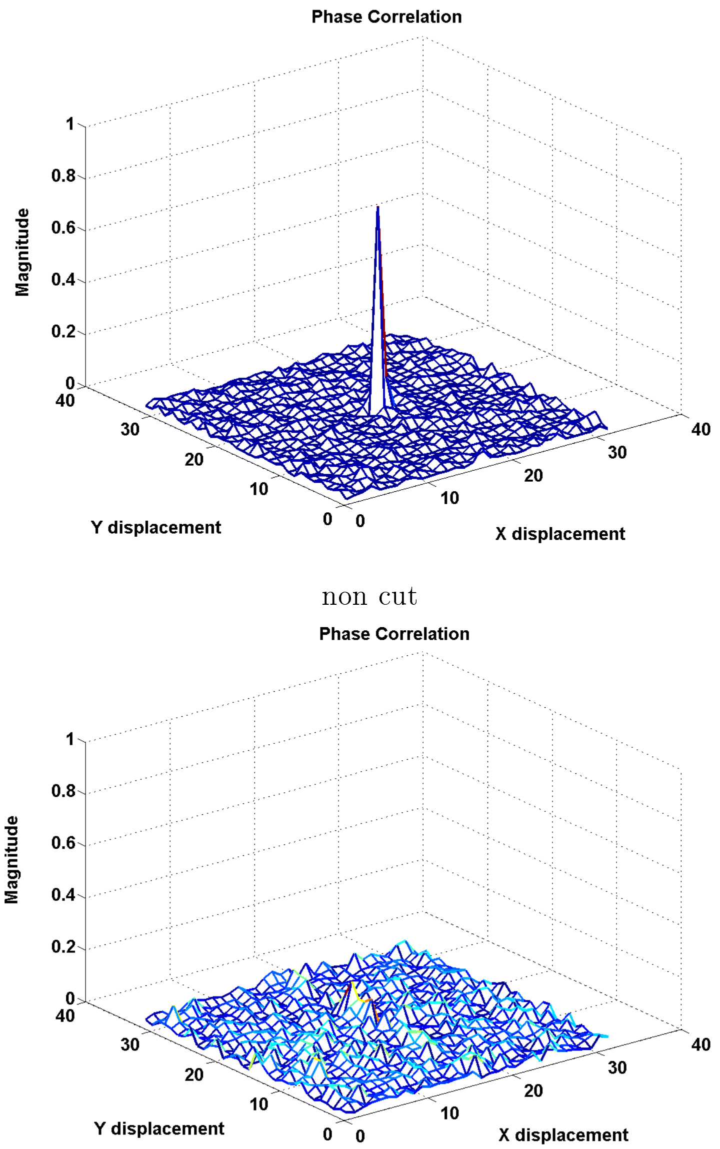
$$s_t = \left(\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_t(i,j)-\mu_t)^3\right)^{\frac{1}{3}} \tag{4.4}$$

where f_t if the tth frame of size $M \times N$.

4.2.3 Phase Correlation Method between frames f_t and f_{t+1} (PCM)

Another useful motion feature is the phase correlation method (PCM) between two frames Wang [2001]. For each frame pair in the video sequence, the first frame is divided into a regular grid of blocks. A similarity metric for each frame pair can then be derived by comparing the edge features contained within each block. The next step is to estimate the motion for each block between the two frames to compensate for differences caused by camera and object motions. For each block in the first frame, the best matching block in the neighborhood around the corresponding block in the second frame is searched. The location of the best matching block can be used to find the offset of each block between the two frames to then compute a motion compensated similarity metric. This metric is performed by a normalized correlation.

The phase correlation method measures the motion directly from the phase correlation map (a shift in the spatial domain is reflected as a phase change in the spectrum domain). This method is based on block matching: for each block r_t in frame f_t is sought the best match in the neighbourhood around the corresponding block in frame f_{t+1}. When one frame is the translation of the other, the PCM has a single peak at the location corresponding to the translation vector. When there are multiple objects moving, the PCM tends to have many peaks, see Figure 4.2.



non cut

cut

Figure 4.2: Phase correlation.

The PCM for one block r_t is defined as:

$$\rho(r_t) = \frac{FT^{-1}\{\widehat{r_t}(\omega)\widehat{r_{t+1}}^*(\omega)\}}{\sqrt{\int |\widehat{r_t}(\omega)|^2 d\omega \int |\widehat{r_{t+1}}(\omega)|^2 d\omega}} \tag{4.5}$$

where ω is the spatial frequency coordinate vector, $\widehat{r_t}(\omega)$ denotes the Fourier transform of block r_t, FT^{-1} denotes the inverse Fourier transform and $\{\}^*$ is the complex conjugate. Figure 4.2 shows the coefficients in $\rho(r_t)$ map of block r_t. In Figure 4.2(a) we show the correlation coefficients resulted of matching two blocks of frames within the shot and in Figure 4.2(b) we show the correlation coefficients of an AT.

By applying a high-pass filter and performing normalized correlation, this method is robust to global illumination changes Porter et al. [2003]. The value of the maximum correlation coefficient is suggested as a measure for each block Porter et al. [2003], but a problem with this measure is that no information of the neighbors of the maximum correlation coefficient is available. Instead of using that measure, we propose the use of the entropy E_r of the block r as the *goodness-of-fit* measure for each block. If the entropy of the correlation metrics is high then it means that there is not much common information between the blocks. If the entropy is low then it means there is a lot of information concentrated in a single coefficiente, i.e., the blocks are similar. We use the Shanon entropy to calculate the similarity between to blocks. The entropy of a variable X is defined as

$$H(X) = -\sum_x P(X)\log_2[P(x)] \tag{4.6}$$

where $P(X)$ is the probability that X is in the state x, and $P\log_2 P$ is defined as 0 if $P = 0$

The similarity metric M_t is defined by the median of all block entropies instead of the mean to prevent outliers Porter et al. [2003]:

$$M_t = \text{median}(E_r) \tag{4.7}$$

4.2.4 Projection Histograms

Projection is defined as an operation that maps an image into a one-dimensional array called projection histogram. The values of the histogram are the sum of the pixels along a particular direction Trier et al. [1996]. Two types of projection histograms are defined. They are at 0-degree (horizontal projection histogram) and 90-degrees (vertical projection histogram) with respect to the horizontal axis:

$$\begin{aligned} M_{hor}(y) &= \frac{1}{x_2 - x_1}\int_{x_1}^{x_2} f(x,y)dx \qquad (4.8)\\ M_{ver}(x) &= \frac{1}{y_2 - y_1}\int_{y_1}^{y_2} f(x,y)dy \qquad (4.9)\end{aligned}$$

Thus, a horizontal projection histogram $h(x)$ of a binary image $f(x,y)$ is the sum of black pixels projected onto the vertical axis x. A vertical projection histogram $v(y)$ of a binary image $f(x,y)$ is the sum of black pixels projected onto the horizontal axis y. The horizontal and vertical projection histograms of the digit 2 is shown as an example in Figure 4.3.

4.2.5 Shape Descriptors

As shape descriptors we use orthogonal moments like Zernike moments Kan and Srinath [2001] and Fourier-Mellin moments Kan and Srinath [2002].

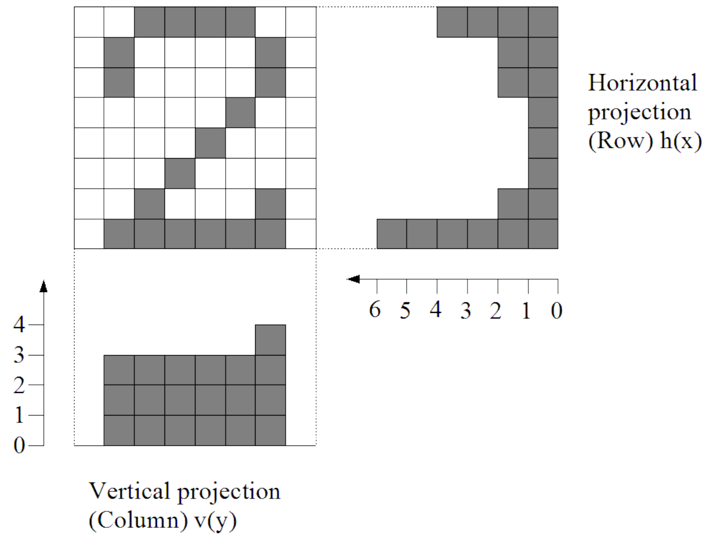

Horizontal projection (Row) h(x)
6 5 4 3 2 1 0
4 3 2 1 0
Vertical projection (Column) v(y)

Figure 4.3: Projection histograms of digit 2.

Zernike Moments

Zernike polynomials, pioneered by Teague Teague [1980] in image analysis, form a complete orthogonal set over the interior of the unit circle $x^2 + y^2 \leq 1$. The Zernike function of order (p, q) is defined in the polar coordinate system (ρ, θ) as

$$V_{p,q}(\rho, \theta) = R_{p,q} e^{jq\theta}, \tag{4.10}$$

where $V_{p,q}$ is a complete set of complex polynomials, p is a positive integer value $p \geq 0$ that represents the polynomial degree, q is the angular dependency and must complain that $|q| \leq p$ with $p - |q|$ even and $R_{p,q}$ is a set of radial polynomials that have the property of being orthogonal inside the unity circumference. These functions have the following expression:

$$R_{p,q}(\rho) = \sum_{k=q, p-|k|=even}^{p} \frac{(-1)^{(p-k)/2}((p+k)/2)!}{((p-k)/2)!((q+k)/2)!((k-q)/2)!}. \tag{4.11}$$

The Zernike moments (ZM) of an image order are the projections of the image function onto these orthogonal basis functions. The ZM of order p is defined as:

$$Z_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta) V_{pq}^*(\rho, \theta) \rho d\rho d\theta \tag{4.12}$$

where $p = 0, 1, 2, \ldots, \infty$ defines the order, $f(\rho, \theta)$ is the image in polar coordinates (ρ, θ), V_{pq} is the Zernike polynomial and $\{\}^*$ denotes the conjugate in complex domain.

For the discrete image, the Equation 4.12 becomes:

$$Z_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x, y) V_{pq}^*(\rho, \theta) \Delta x \Delta y \tag{4.13}$$

where $x^2 + y^2 \leq 1$, $x = \rho \cos\theta$ and $y = \rho \sin\theta$.

Zernike moments are orthogonal and rotation invariant. But when they are used for scale invariant pattern recognition, Zernike moments have difficulty in describing images of small size.

Fourier-Mellin Moments

The circular Fourier or radial Mellin moments of an image function $f(\rho, \theta)$ are defined in the polar coordinate system (ρ, θ) as:

$$F_{pq} = \int_0^{2\pi} \int_0^{\infty} \rho^p f(\rho, \theta) e^{jq\theta} \rho d\rho d\theta, \tag{4.14}$$

where $q = 0, \pm1, \pm2, \ldots$ is the circular harmonic order and the order of the Mellin radial transform is an integer p with $p \geq 0$. Now introduce the polynomial $Q_p(\rho)$ defined in Sheng and Shen [1994] as:

$$Q_p(\rho) = \sum_{k=0}^{p} (-1)^{p+k} \frac{(p+k+1)!}{(p-k)!k!(k+1)!}. \tag{4.15}$$

Then the (p, q) order Orthogonal Fourier Mellin Moments (OFMM) function U_{pq} and the OFMM moments O_{pq} can be defined in polar coordinate system (ρ, θ) as:

$$U_{pq}(\rho, \theta) = Q_p(\rho) e^{-jq\theta}, \tag{4.16}$$

$$O_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x, y) U_{pq}(\rho, \theta) \Delta x \Delta y, \tag{4.17}$$

where $x^2 + y^2 \leq 1$, $x = \rho \cos\theta$ and $y = \rho \sin\theta$.

For a given degree p and circular harmonic order q, $Q_p(\rho) = 0$ has p zeros. The number of zeros in a radial polynomial corresponds to the capacity of the polynomials to describe high frequency components of the image. Therefore, for representing an image with the same level of quality, the order of p ortogonal Fourier-Mellin is always less than the order of other moments (high order moments are more sensitive to noise) Kan and Srinath [2002].

Fourier-Mellin moments are also orthogonal and rotation invariant. Fourier-Mellin moments are better able to describe images of small size Kan and Srinath [2002].

4.3 Dissimilarity Measures

This section describes the dissimilarity measures used for matching visual information. The dissimilarity is determined as a distance between some extracted features. Different features are computed from each frame, thus each frame is represented by a set of features. The dissimilarity measure is computed between same features (for example Zernike moments) of two consecutive frames. Feature vectors are considered as histograms in terms of dissimilarity measure. Figure 4.4 shows the dissimilarity schema, where $H^i(f_t)$ is ith feature extracted from frame f_t and R is the number of features (for example: $H^1(f_t)$ and $H^1(f_{t+1})$ represent RGB color histograms of frame t and frame $t+1$ respectively, $H^2(f_t)$ and $H^2(f_{t+1})$ represent HSV color histograms of frame t and frame $t+1$ and so on). Then, all dissimilarities computed between frame f_t and f_{t+1} form a new vector $\mathbf{d_t}$ that will be used as input data to the classifier.

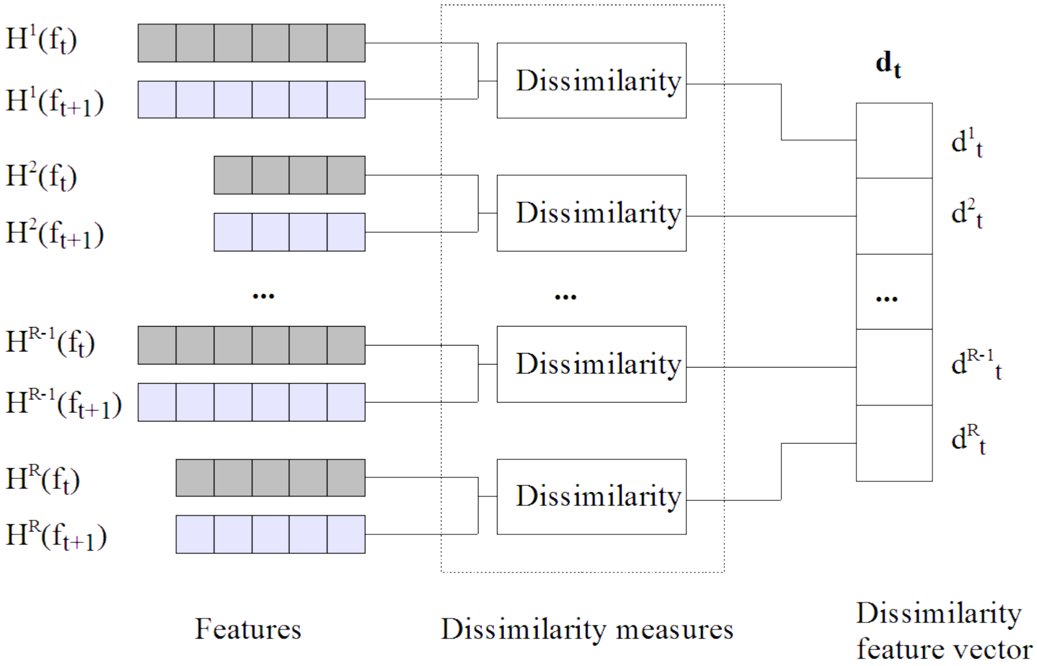


Figure 4.4: Pairwise dissimilarity measures. $H^i(f_t)$ represent the i-th "histogram" feature of frame f_t

Many dissimilarity measures have been used for content analysis. Among the most used we focus on $L1$ norm, cosine dissimilarity, histogram intersection and χ^2 which seemed to be more appropriate to our features. In this case $L1$ norm distance and χ^2 distance are used as dissimilarity measure.

Several other statistical measures have been reviewed and compared in Ford et al. [1997] and Ren et al. [2001]. Then, the pairwise dissimilarity measure between features is used as an input in the SVM classifier.

Figures 4.5 and 4.6 display the dissimilarity vector of different features. We include some motion and abrupt illumination change in the video segments. The isolate picks are the ATs and the other high values that stay together are caused by camera or object motion. We can see in both figures that color histograms are more tolerant to motion but also is very sensitive to illumination changes as it is seen in the second figure, more or less at frame position 2250 (where we can find an isolate pick). Other feature that stays stable is the correlation between consecutive frames, the strength of the pick are higher using this feature. But the correlation of some frames that belongs to different shots has low value misleading the detection.

After computing the dissimilarity vector, compound by the pairwise dissimilarities of all features, we are now able to detect the transitions. Thus, an AT occurs if the dissimilarity

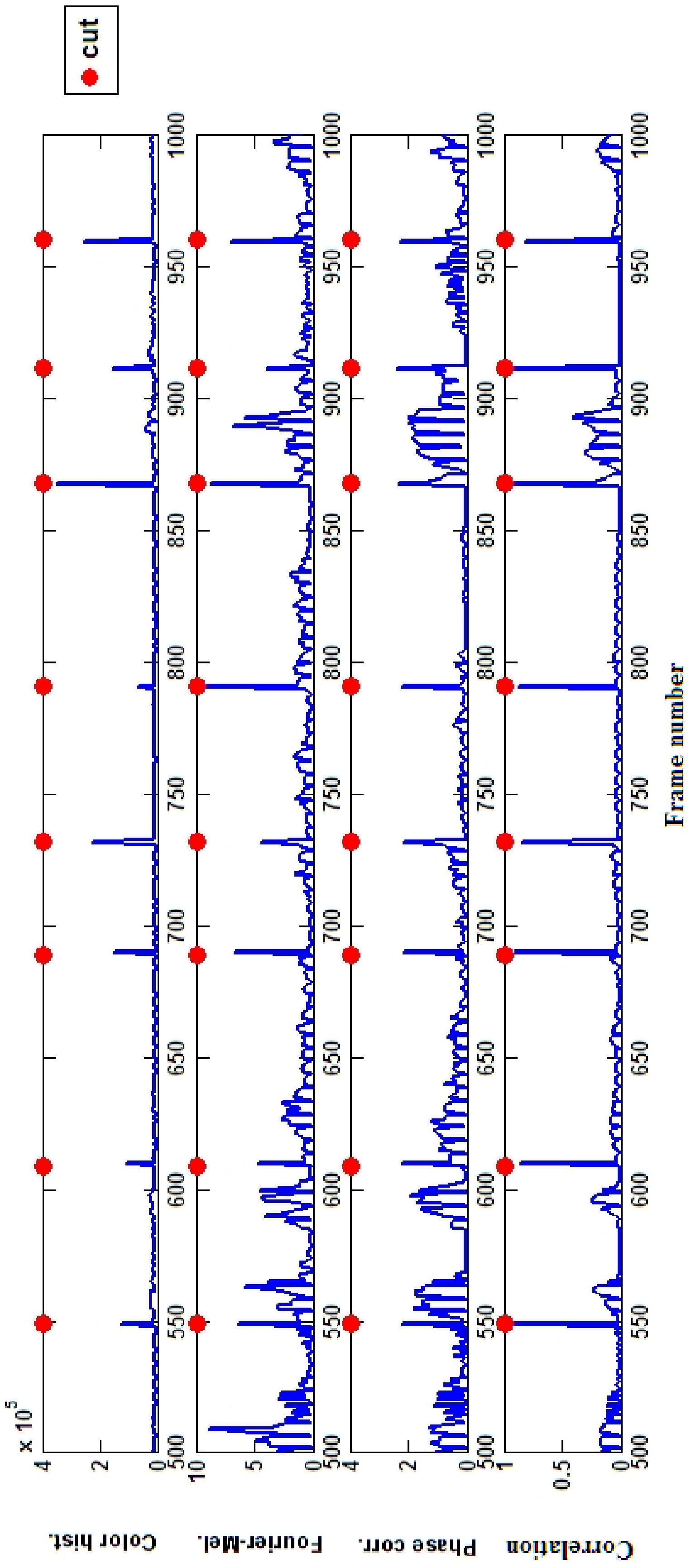


Figure 4.5: Dissimilarity vectors for different features

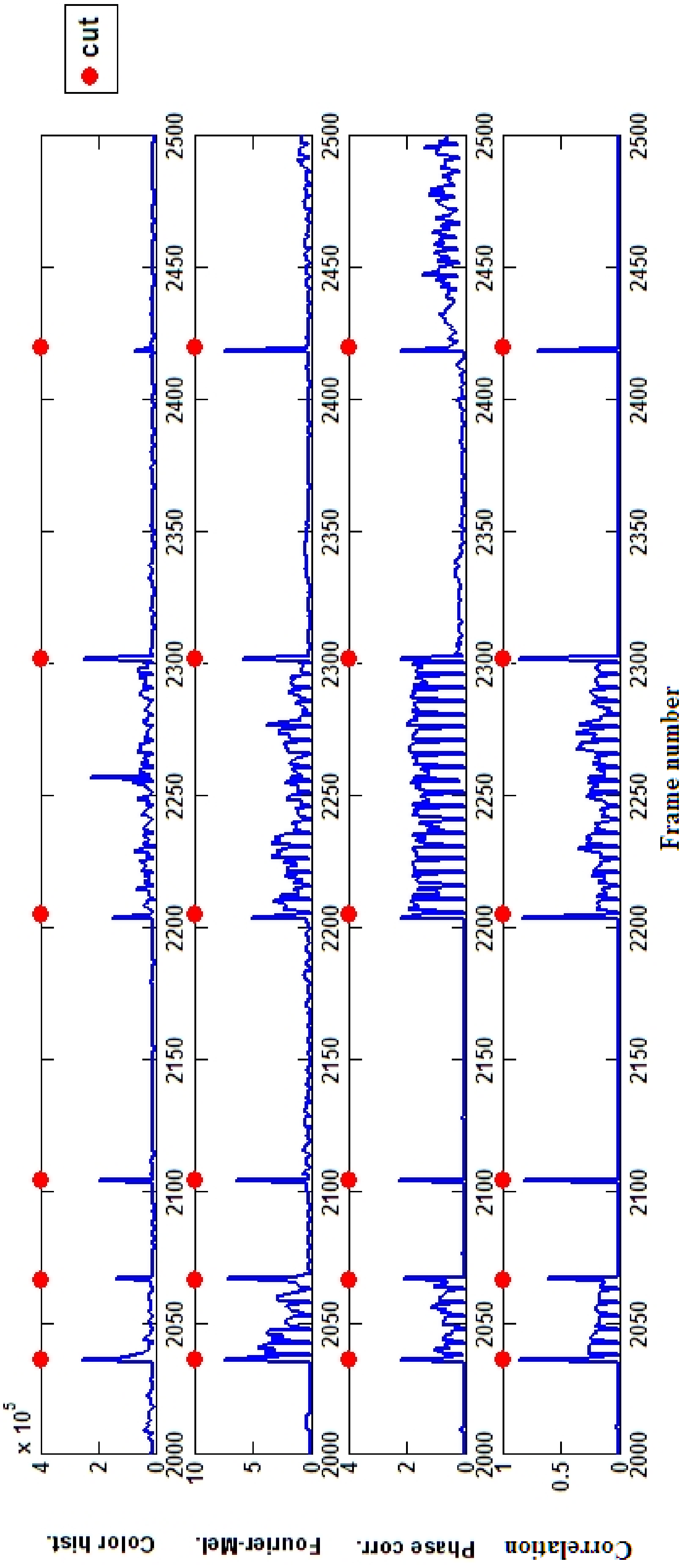


Figure 4.6: Dissimilarity vectors for different features

is high. If we adopt a threshold-based approach, we need to set thresholds for each feature. The second problem is how to choose the features and blend them. Since we are proposing a learning-based approach, we eliminate the need for threshold setting and we are able to use multiple features simultaneously. Thus, this dissimilarity vector will be used as input data to the SVM classifier in order to detect the ATs.

4.4 Machine Learning Approach

The system which we propose, deals with a statistical learning approach for video cut detection. However, our classification framework is specific. Following the structure presented in Section 4.1, in the first stage we choose as features: color histograms in different color spaces (RGB, HSV and opponent color), shape descriptors (Zernike and Fourier-Mellin moments), projections histograms, color moments (luminance variance) and phase correlation. In the second stage, we test different dissimilarity measures: $L1$ norm, cosine dissimilarity, histogram intersection and χ^2 distance. Then in the third stage, each dissimilarity feature vector (distance for each type of feature: color histogram, moments and projection histograms) is used as input to the classifier.

In Algorithm 4.1, we present the steps for computing the dissimilarity vectors. In the first loop, $Hist[t]$ corresponds the color histograms in the different color spaces, thus we have 3 color histograms, $Shape[t]$ corresponds the Zernike and Fourier-Mellin moments, $Colormom[t]$ is the variance of luminance and $Proj[t]$ corresponds to horizontal and vertical projection histograms. In the second loop we calculate the dissimilarity between features of consecutive frames. The function $Dissimilarity(.,.)$ calculates the four dissimilarity measures used in this work and finally also in this loop we calculate the phase correlation.

Algorithm 4.1: Dissimilarity vector calculation

```
Data: Video frames
Result: Dissimilarity d vectors
foreach frame t in the video do
    Hist[t] = Color histograms;
    Shape[t] = Shape descriptors;
    Colormom[t] = Color moments;
    Proj[t] = Projection histograms;
end
for t = 1 to Video size - 1 do
    d[0,t] = Dissimilarity(Hist[t], Hist[t+1]);
    d[1,t] = Dissimilarity(Shape[t], Shape[t+1]);
    d[2,t] = Dissimilarity(Proj[t], Proj[t+1]);
    d[3,t] = Phase Correlation between frame t and frame t+1;
end
```

Once we have the dissimilarity feature vector, we are able to detect the ATs. In the stage of classification we adopt a supervised classification method. As we use a consecutive pairwise dissimilarity, the number of dissimilarity vectors is one less than the number of video's frames, e.g., if we have a video of L frames, the number of dissimilarity vectors is $L-1$. The dissimilarity vectors are classified into two classes: "cut" and "non cut", i.e., we have a binary classification. Other characteristic of the data is that the number of dissimilarity vectors that correspond for "cut" is much smaller than ones that correspond for "non cut". This means that the data available is unbalanced.

Based on the characteristics of the data, we choose SVM as our classifier. SVM is a learning machine that can perform binary classification. The two key features of SVM are the generalization theory and kernel functions. Under the premise of zero empirical risk, SVM guarantees the correct classification of the whole training set and obtains the best generalization performance by maximizing the classification margin. SVM can obtain global optimal solution in theory, especially suitable to solve the classification problems with small samples. SVM solves linearly inseparable problem by non-linearly mapping the vector in low dimension space to a higher dimension feature space (thanks to kernel functions) and constructing an optimal hyperplane in the higher dimension space.

We will focus on SVMs for classification. Basically, SVM methods project data to classify in a space of large (possibly infinite) dimension, where a linear criterion is used. For any training set, one can choose an appropriate projection Φ so that linear separability may be achieved. Computation is done without an explicit form of the projection, but only with the kernel corresponding to the scalar product between projections.

The model is thus specified by choosing the kernel K:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

and a function f which sign is the predicted class:

$$f(x) = \mathbf{w} \cdot \Phi(x) + b.$$

Given training data $x_1, x_2, \ldots, x_n$ that are vectors in some space $\mathcal{X} \subseteq R^d$. Also given their labels $y_1, y_2, \ldots, y_n$ where $y_i \in \{-1, 1\}$. We will denote $\mathcal{T} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ a training set generated independent and identically distributed according to $(\mathcal{X}, \mathcal{Y})$. The computation of $\mathbf{w}$ is achieved by minimizing $||\mathbf{w}||$ under correct classification of the training set, i.e. $\forall_i$ $y_i f(x_i) \geq 1$. This is equivalent to maximizing the margin between training points and the separating hyperplane.

It can be proven Boser et al. [1992] that $\mathbf{w}$ is of the form $\sum_i \alpha_i y_i \Phi(s_i)$, where the α_i come from the following quadratic optimization problem:

Maximize

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(s_i, x_j)$$

subject to

$$0 \leq \alpha_i \leq C, \forall_i \text{ and } \sum_{i=1}^{N_s} \alpha_i y_i = 0,$$

where C is a misclassification cost used in order to tolerate noisy configuration (soft margin). The s_i with non-zero α_i are called *support vectors*.

Finally, the decision function g in SVM framework is defined as:

$$\begin{aligned} g(x) &= \operatorname{sgn}(f(x)) && (4.18)\\ f(x) &= \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b && (4.19) \end{aligned}$$

where $b \in \mathbb{R}$ and α_i parameters are computed, considering the SVM optimization.

Several common kernel functions are used to map data into high-dimensional features space:

Linear:

$$K(x_i, x_j) = x_i \cdot x_j \tag{4.20}$$

Polynomial kernel:

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{4.21}$$

Gaussian radial basis kernel :

$$K(x_i, x_j) = e^{-||x_i - x_j||^2/2\sigma^2} \tag{4.22}$$

Gaussian kernel with χ^2 distance (Gauss-χ^2):

$$K(x_i, x_j) = e^{-\chi^2(x_i, x_j)/2\sigma^2} \tag{4.23}$$

Triangular kernel Fleuret and Sahbi [2003]:

$$K(x_i, x_j) = -||x_i - x_j|| \tag{4.24}$$

Each kernel function results in a different type of decision boundary.

The SVM problem is convex whenever the used kernel is a Mercer one (c.f. Appendix A). The convexity ensures the convergence of the SVM algorithm towards a unique optimum. The uniqueness of the solution is one of the main advantages of the SVM compared to other learning approaches such as neural networks Boughorbel et al. [2004]. See Appendix A for further details.

In Figure 4.7, we present our training framework. The dissimilarity vectors are used for training our SVM classifier, the training data is constitute by one or more videos. As a result of the training, SVM computes the support vectors, which are the data points that lie closest to the decision surface. Therefore, thanks to support vectors we have a trained machine. Figure 4.8 shows the test framework based on the support vectors computed in the training stage. We are now able to detect when a "cut" occurs.

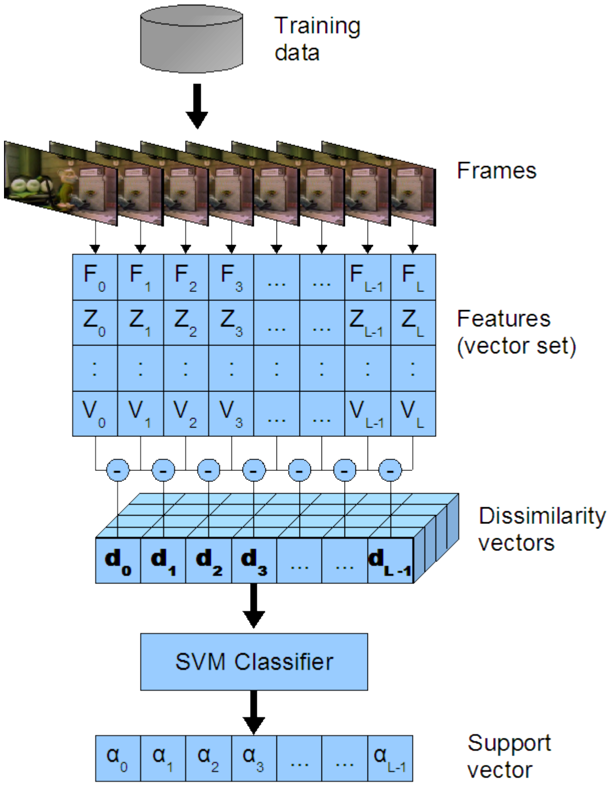

Training data
Frames
F_0 F_1 F_2 F_3 ... ... F_{L-1} F_L
Z_0 Z_1 Z_2 Z_3 ... ... Z_{L-1} Z_L
V_0 V_1 V_2 V_3 ... ... V_{L-1} V_L
Features (vector set)
-
d_0 d_1 d_2 d_3 ... ... d_{L-1}
Dissimilarity vectors
SVM Classifier
α_0 α_1 α_2 α_3 ... ... α_{L-1}
Support vector

Figure 4.7: **Learning-based approach for video cut detection: Training**. Feature vectors $F_i, Z_i, \dots C_i$ represent Fourier Mellin moments, Zernike moments, Color histogram, and the other features detailed in Section 4.2 and d_t is the dissimilarity between consecutive frames.

Another key in classification is the normalization of the input data. The objective of normalization is to equalize ranges of the features removing statistical error. The normalization methods tested in our work are the statistical normalization and the min-max normalization Ortega-Garcia et al. [2002].

4.5 Experiments

In this section we present the experiments conducted in order to choose the better parameters for our system and also compare our method with other methods in TRECVID evaluation.

4.5.1 Data Set

The *training set* consists of a single video of 9078 frames (5mins. 2 secs.) with 128 "cuts" and 8950 "non cuts". This video is captured from a Brazilian TV-station and is composed by a segment of commercials. The training video was labeled manually by ourselves. The *test set* is composed by two data sets of TRECVID evaluation. The first test set is the TRECVID-2002 data set that was used to define the best parameters, i.e., feature combinations, dissimilarity

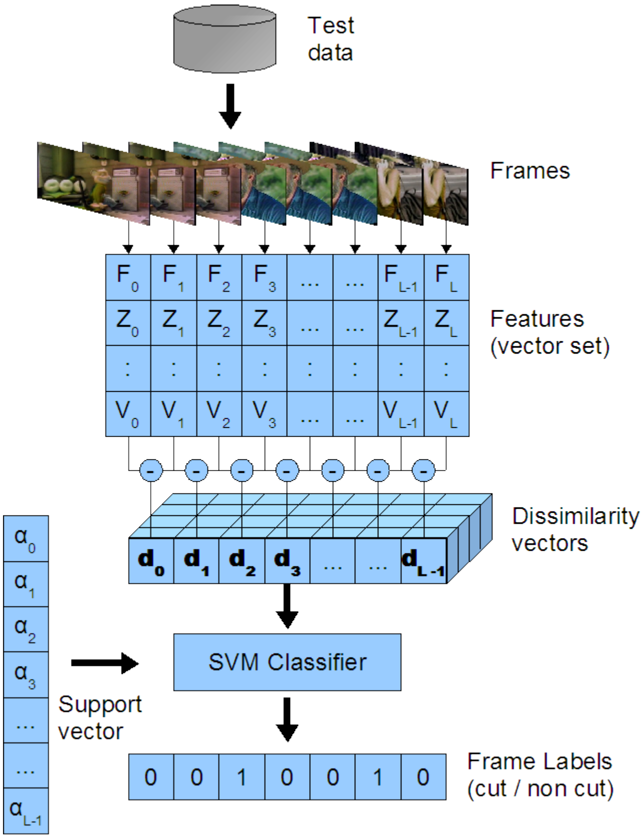


Figure 4.8: **Learning-based approach for video cut detection: Test**. Feature vectors $F_i, Z_i, \ldots C_i$ represent Fourier Mellin moments, Zernike moments, Color histogram, and the other features detailed.

measures and kernel functions. The second test set, TRECVID-2006 data, was used to compare the performance of our system with other methods proposed by teams that participate in the evaluation.

We strictly follow the TRECVID protocol in our tests. We use the precision, recall and $F1$ statistics defined in TRECVID protocol.

$$Recall = \frac{correct}{correct + missed};$$

$$Precision = \frac{correct}{correct + false}.$$

A good detector should have high precision and high recall. $F1$ is a commonly used metric that combines precision and recall values. If both values are high then $F1$ is high.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4.25}$$

4.5.2 Features

As our objective is to avoid pre-processing and post-processing steps we combine distinctive features. In the case of global color histograms we use three different color spaces:

RGB_h	in RGB space, 2 bits for each channel (64 bins)
HSV_h	in HSV space, 2 bits for each channel (64 bins)
$R - G_h$	in opponent color space, we use the second channel $(R-G)$, 64 bins

In the case of RGB_h and HSV_h we use 64 bins (2 bits per channel). In shot boundary detection the number of bits per channel is set to 2 or 3 in order to reduce sensitivity to noise, slight light and object motion as well as view changes Lienhart et al. [1997a]; Santos [2004]. We use opponent color space (brightness-independent chromaticities space) in order to make our set of features more robust to illuminance changes. The advantage of this representation is that the last two chromaticity axes are invariant to changes in illumination intensity and shadows. Thus, we use the second channel $(Red - Green)$ and divide it in 64 bins.

For shape descriptors we use Fourier-Mellin and Zernike moments:

Z_h	moments of order 5
F_h	moments of order 4

For Zernike moments we select moments of order 5 arranged in a vector of 12 elements. Greater orders are not necessary, since the content of consecutive frames that belongs to the same shot is very similar. Toharia et al. [2005] compare moments of order 3, 5 and 10. The performance between the three orders are similar. In the case of Fourier-Mellin moments, we choose moments of order 4 arranged in a vector of 24 elements. For representing an image over the same level of quality is always less than the order of other moments Kan and Srinath [2002].

Other features used in our framework are the projections histograms in X-axis and Y-axis direction (horizontal and vertical), phase correlation, computed in the frequency domain and the luminance variance (color moments):

V_h	vertical projection histograms, the size depends on the number frame's columns
H_h	horizontal projection histograms, the size depends on the number frame's rows
PC	32×32 blocks
Var	luminance variance

For phase correlation we choose a block size of 32×32. Porter et al. [2003] suggest the use of the maximum correlation value as a measure for each block. A drawback with this method is that we do not have information of the neighbors of the maximum correlation value. We propose the use of the entropy of the phase correlation blocks as the *goodness-of-fit* measure. The entropy give us global information of the block, not only information for a single element of the block. Although, the phase correlation feature is particularly relevant in presence of illumination changes, it provides false positive cuts for "black" frames due to MPEG-1 artifacts. In order to overcome this limitation, we add the luminance variance (Var). Indeed, two "black" frames phase correlation will be high like for non-similar images while variance will be little in the first case and high in the second. Indeed, the phase correlation feature of two successive "black" frames will be high like in case of two non-similar frames while variance will allow us to discriminate these configurations. In the case of projection histograms, they depend on the size of the frame.

Since our framework is tested using TRECVID data sets, we strictly follow the TRECVID protocol in our tests. We can provide up to 10 different runs (10 different choices of parameters, features or kernels). In Table 4.1, we present the visual feature vectors used in our tests. The combinations for each run were selected empirically, evaluating all possible combinations and choosing the best ones.

Run	Features
1	HSV_h, F_h, Z_h, H_h, PC, Var
2	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, PC, Var
3	$R-G_h$, HSV_h, RGB_h, F_h, H_h, PC, Var
4	HSV_h, RGB_h, F_h, Z_h, PC, Var
5	HSV_h, RGB_h, F_h, Z_h, H_h, PC, Var
6	RGB_h, F_h, Z_h, V_h, PC, Var
7	RGB_h, F_h, Z_h, V_h, H_h, PC, Var
8	HSV_h, RGB_h, F_h, Z_h, V_h, H_h, PC, Var
9	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, H_h, PC, Var
10	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, H_h, V_h, PC, Var

Table 4.1: Combination set of visual features used in our tests.

4.5.3 TRECVID 2002

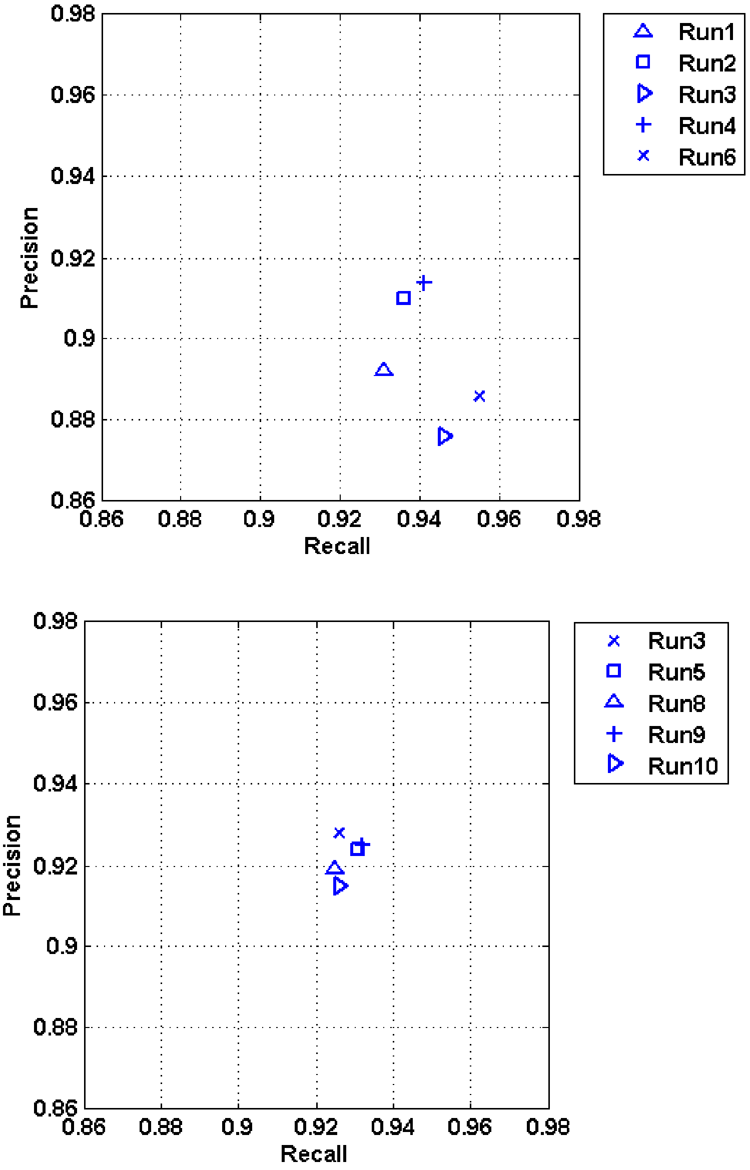
The shot boundary test collection of TRECVID-2002 contains 4 hours and 51 minutes of video. The videos are mostly of a documentary/educational nature, but very varied in age, production style, and quality. At a total, there are 18 videos in MPEG-1 with a total size of 2.88 gigabytes. The videos contain 545,068 total frames and 2,090 shot transitions with 1,466 cuts. For all videos, shot segmentation reference data has been manually constructed by the National Institute of Standards and Technology (NIST).

Table 4.2 shows the best three results for each run evaluation in terms of recall, precision and $F1$ measures. We also present the kernel functions and the dissimilarity distances used for comparing the feature vectors. We can observe that the run with the best recall has the worst precision. Something similar occurs with the run with the best precision, it has the worst recall. Now observing the best $F1$ measures, the run that achieves the highest value uses all the feature set (*Run 10*). In fact, this run has a more equilibrated recall/precision, i.e., both values are high. This means that using the hold data set we can reduce the number of false detection and missing transitions. Something that we have to take into account is that the results are very closer. When we refer to a run with the worst result, we mean that it is worst compared with the other results and not because the result is poor. Now, analyzing the other factors, kernel function and dissimilarity measure, the kernel function that best performs is the Gauss-χ^2 kernel. In the case of the dissimilarity measures, we can not conclude anything definitely because the results are very heterogeneous.

Run	Recall	Precision	F1	Kernel	Diss. Measure
1	0.929	0.923	0.926	Gauss-χ2	χ^2
	0.881	**0.950**	0.914	Gauss-χ2	Cos
	0.931	0.892	0.911	Gauss-χ2	L1
2	0.944	0.909	0.926	Gauss-χ2	χ^2
	0.936	0.910	0.923	Gauss-χ2	L1
	0.930	0.902	0.916	Gauss-χ2	Hist.Int.
3	0.926	0.928	0.927	Gauss-χ2	Cos
	0.927	0.916	0.922	Gauss-χ2	Hist.Int.
	0.934	0.898	0.916	Gauss-χ2	χ^2
4	0.941	0.914	0.927	Gauss-χ2	L1
	0.930	0.915	0.923	Gauss-χ2	Hist.Int.
	0.933	0.911	0.922	Gauss-χ2	χ^2
5	0.931	0.924	0.927	Gauss-χ2	Cos
	0.927	0.923	0.925	Gauss-χ2	Hist.Int.
	0.947	0.889	0.917	Gauss-χ2	χ^2
6	0.945	0.911	0.928	Gauss-χ2	Hist.Int.
	0.926	0.914	0.920	Gauss-χ2	Cos
	0.955	0.886	0.919	Gauss-χ2	L1
7	0.936	0.919	0.928	Gauss-χ2	Hist.Int.
	0.922	0.916	0.919	Gauss-χ2	Cos
	0.955	0.877	0.915	Gauss-χ2	χ^2
8	0.936	0.921	0.928	Gauss-χ2	Hist.Int.
	0.925	0.919	0.922	Gauss-χ2	Cos
	0.951	0.881	0.915	Gauss-χ2	χ^2
9	0.932	0.925	0.929	Gauss-χ2	Cos
	0.924	0.916	0.920	Gauss-χ2	Hist.Int.
	0.944	0.892	0.918	Gauss-χ2	χ^2
10	0.936	0.923	**0.930**	Gauss-χ2	Hist.Int.
	0.926	0.915	0.920	Gauss-χ2	Cos
	0.923	0.911	0.917	Triangle	Hist.Int.

Table 4.2: Measure of performance for each run.

Through Table 4.2, it is not possible to extract a conclusion with respect to the dissimilarity measures. Therefore, we make an analysis with the best results for each type of dissimilarity measure, see Figure 4.9 and 4.10. The performance is evaluated in function of recall and precision. Even though the performance of the dissimilarity measures are similar, we can see that histogram intersection and cosine dissimilarities outperform lightly the results of $L1$ and χ^2 dissimilarities. In almost all the cases the kernel function with best performance is the Gaussian-χ^2. In Figure 4.5.3, the triangle marker inside the circle is the only run where Triangle kernel function outperforms other kernels. The best recall has the worst precision (*Run*6 in Figure 4.5.3), this means that experiment run detects almost all the transitions but it also has various false positives. This also occurs with the best precision, it also presents the worst recall (*Run*8 in Figure 4.5.3). This means that almost all the transitions detected by the experiment run are true, but it misses various transitions. The experiment *Run*10, see Figure 4.5.3, is the one that has a more equilibrate recall/precision and also has the best F1 measure.



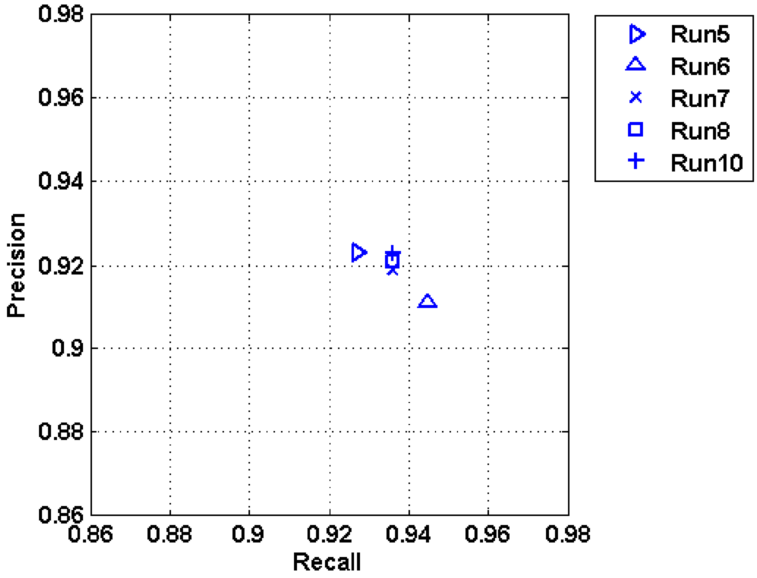
[$L1$ norm dissimilarity]

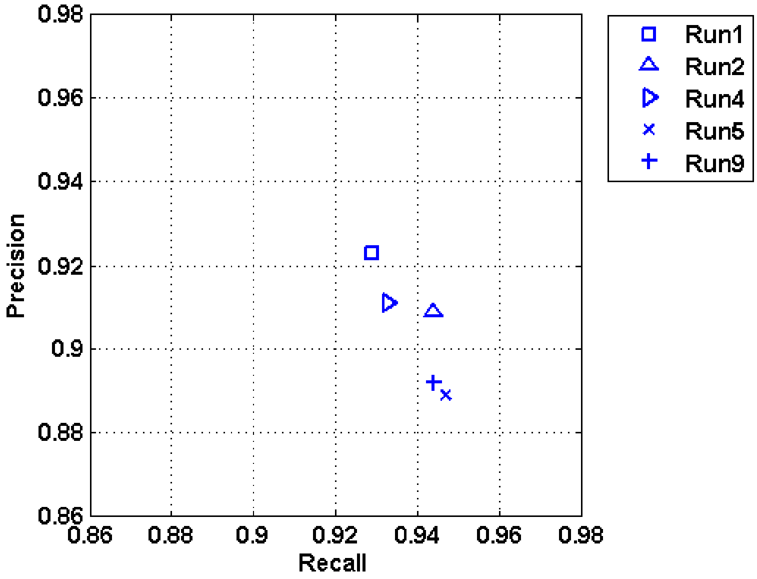
[Cosine dissimilarity]

Figure 4.9: Precision/Recall measure performance for $L1$ norm and cosine dissimilarity.

Since we already have selected the ten different combinations of the features, our objective is to find the other parameters: kernel function and dissimilarity measure. Figures 4.11 and

[Histogram intersection dissimilarity]

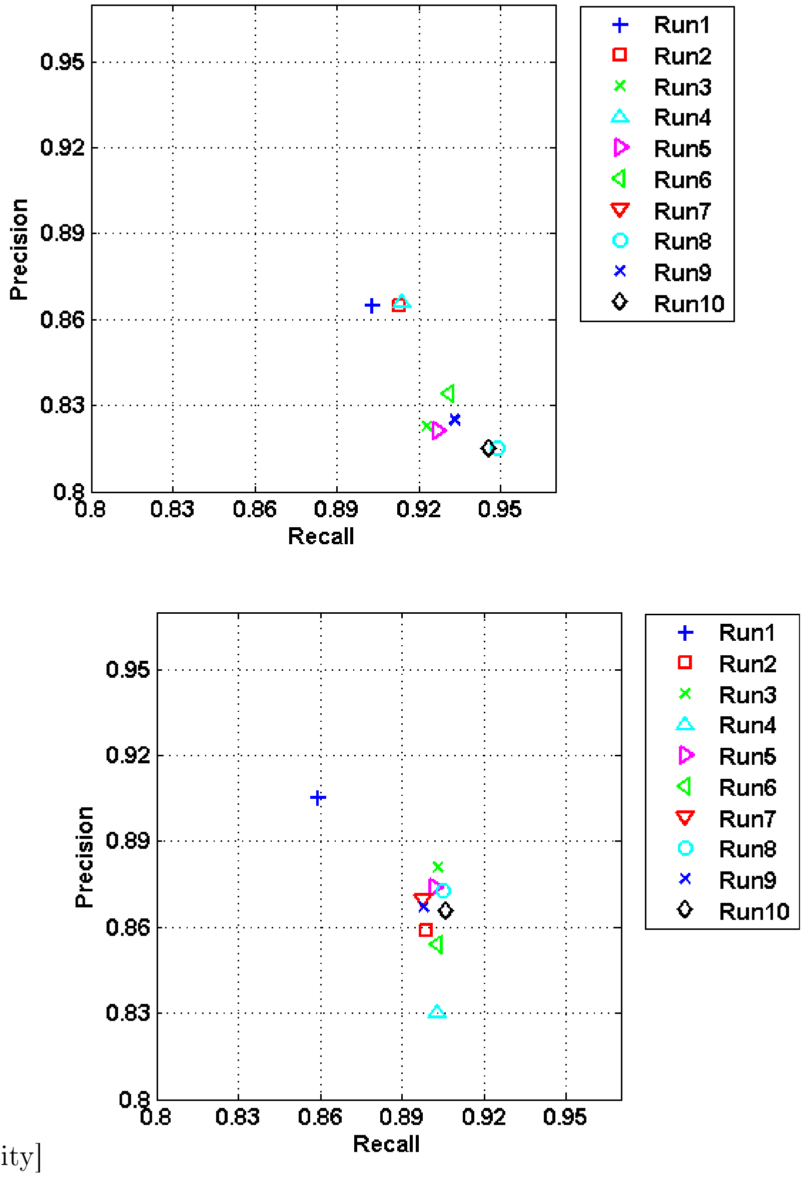




[χ^2 dissimilarity]

Figure 4.10: Precision/Recall measure performance for histogram intersection and χ^2 dissimilarity.

4.12 shows the performance for all the experiment runs using the linear kernel function and different dissimilarity measures. We find the best recall results in Figure 4.5.3 where the $L1$ norm is used as a dissimilarity measure. But unfortunately the precision results are the worst. We see that this behavior is the same for all precision values, i.e., the $L1$ norm has the worst precision values. The cosine dissimilarity (Figure 4.5.3) and histogram intersection (Figure 4.5.3) have a more equilibrate relation of recall and precision. Other characteristic is that the experiment runs are close together, this means that independent of the experiment run, the performance of the system is almost the same. The χ^2 dissimilarity (Figure 4.5.3) also shows a good performance, but comparing with the cosine dissimilarity and histogram intersection is a little bit worst. In conclusion, the dissimilarity measures that seem better than the linear kernel function are the cosine dissimilarity, histogram intersection and finally the χ^2 dissimilarity.

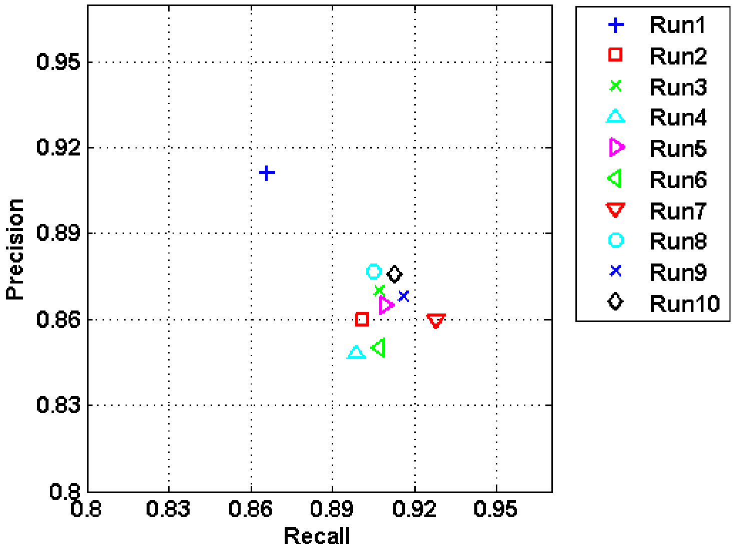


[L1 norm]

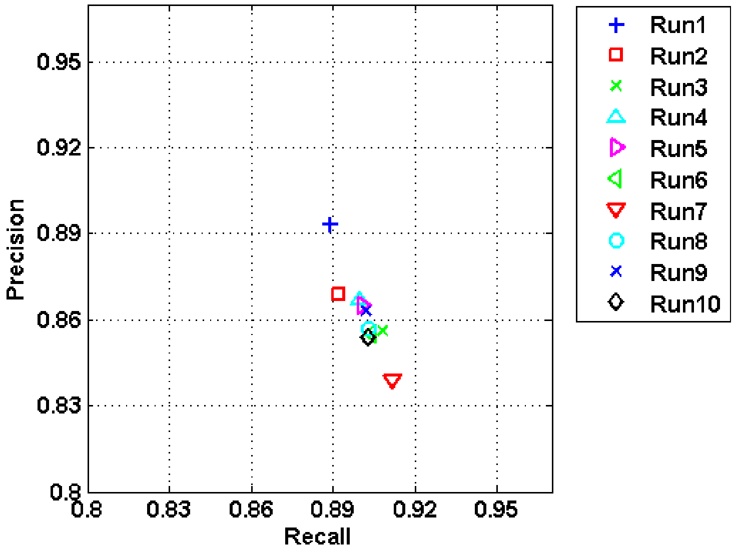
[Cosine dissimilarity]

Figure 4.11: Precision/Recall measure for all runs using the *Linear* kernel function.

Figures 4.13 and 4.14 show the performance for all the experiment runs using the polynomial kernel function and different dissimilarity measures. In Figure 4.5.3, we can see that the performance of the system is increased using the polynomial kernel instead of the linear kernel (see Figure 4.5.3). The relation recall/precision is also better with polynomial kernel and the results are closer between them, i.e., they are more or less the same. Again the cosine dissimilarity (Figure 4.5.3) and histogram intersection (Figure 4.5.3) have a more equilibrate



[Histogram intersection]

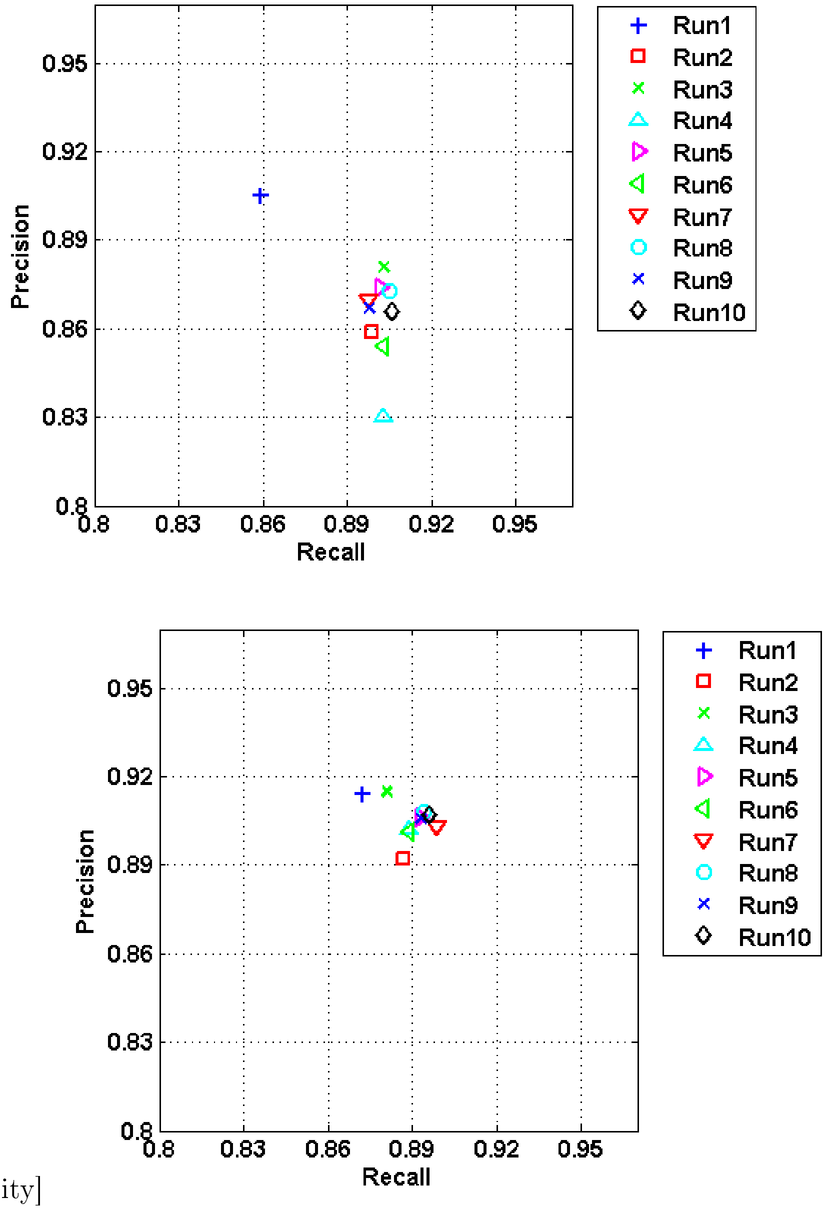


[χ^2 dissimilarity]

Figure 4.12: Precision/Recall measure for all runs using the *Linear* kernel function.

relation of recall/precision and experiment runs are also close together. The χ^2 dissimilarity (Figure 4.5.3) also shows a good performance. This dissimilarity has the best recall, but unfortunately the precision become worst. An interesting fact that we can notice is that in all the cases (the four dissimilarities) the recall increase while the precision decrease. Again the best dissimilarities are cosine and the histogram intersection. In conclusion, the performance really increases when the system uses the $L1$ norm with the polynomial kernel function. Another interesting fact is that the relation recall/precision are more stable in all the cases. As it was established for linear kernel, the cosine and the histogram intersection also show the best performance when the polynomial kernel function is used .

Then the next kernel function to be tested is the Gaussian-$L2$. Figures 4.15 and 4.16

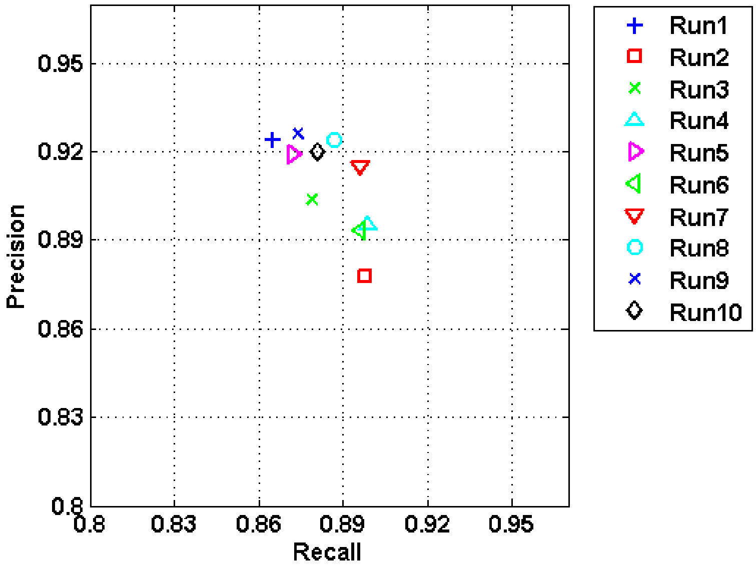


[L1 norm]

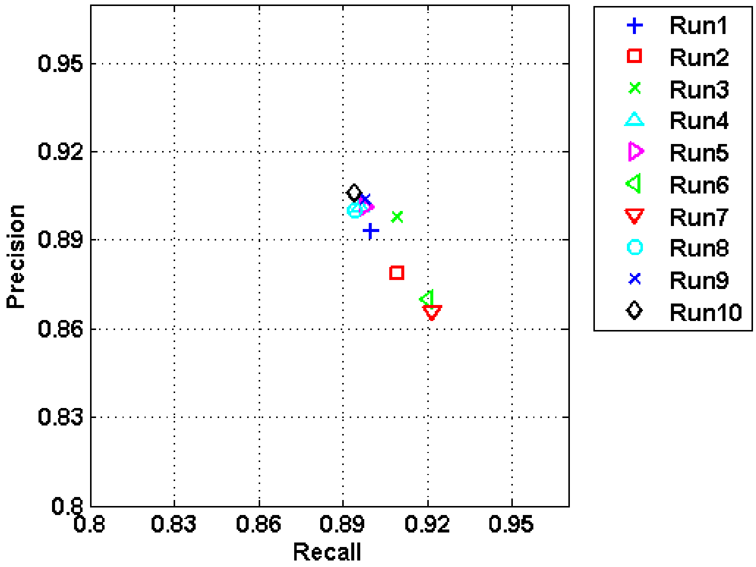
[Cosine dissimilarity]

Figure 4.13: Precision/Recall measure for all runs using the *Polynomial* kernel function.

present the performance for all the experiment runs using the Gaussian-$L2$ kernel function and different dissimilarity measures. As it occurs with the polynomial kernel function, the Gaussian-$L2$ outperforms the linear kernel. In the case of the $L1$ norm, see Figure 4.5.3, the behavior of the system with Gaussian$-L2$ is similar to the behavior of the system with the linear kernel. In both cases the results are spread and have high recall values, but low precision values. As it occurs with linear and polynomial kernel, the cosine dissimilarity (Figure 4.5.3) and histogram intersection (Figure 4.5.3) have a more equilibrate relation of recall/precision and experiment runs are also close together. The χ^2 dissimilarity (Figure 4.5.3) increases a little the recall, but it lose performance in precision. Compared to linear kernel the results are better, but when compared with the polynomial kernel, the recall maintains almost the



[Histogram intersection]

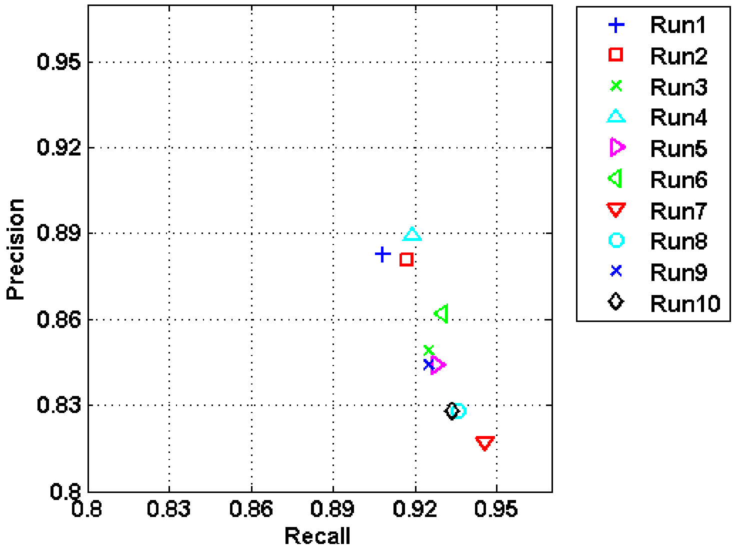


[χ^2 dissimilarity]

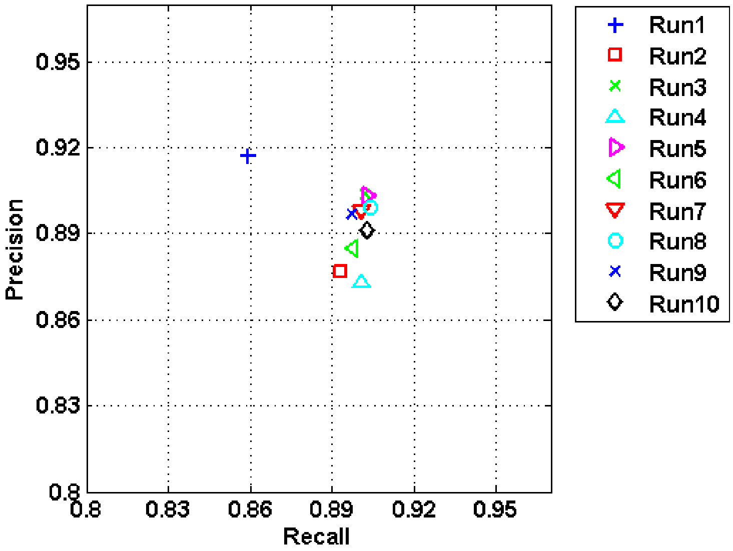
Figure 4.14: Precision/Recall measure for all runs using the *Polynomial* kernel function.

same values but it decreases in precision. We can conclude again that cosine dissimilarity and histogram intersection are the best dissimilarity measures. An interesting fact is that with the Gaussian-$L2$ kernel it was possible to outperform, in recall and precision, the quality of the results compared to linear kernel.

Now, we evaluate another Gaussian kernel, but instead of using the $L2$ norm distance, we use the χ^2 distance. When we presented the best results for each experiment run, we saw that the best kernel was the Gaussian-χ^2. Figures 4.17 and 4.18 present the performance for all the experiment runs using the Gaussian$-\chi^2$ kernel function and different dissimilarity measures. With the four dissimilarity measures the system gain in performance using the Gaussian$-\chi^2$ kernel function. The gain is not only in a better precision but also in a better recall, but



[L1 norm]

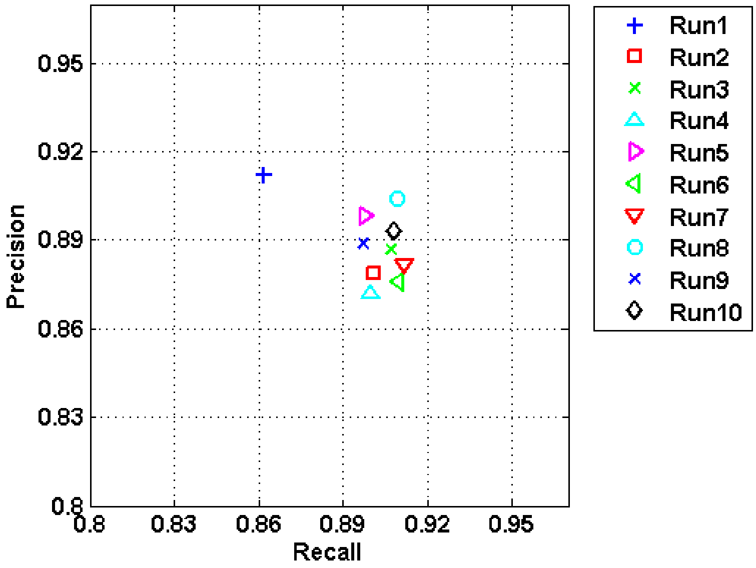


[Cosine dissimilarity]

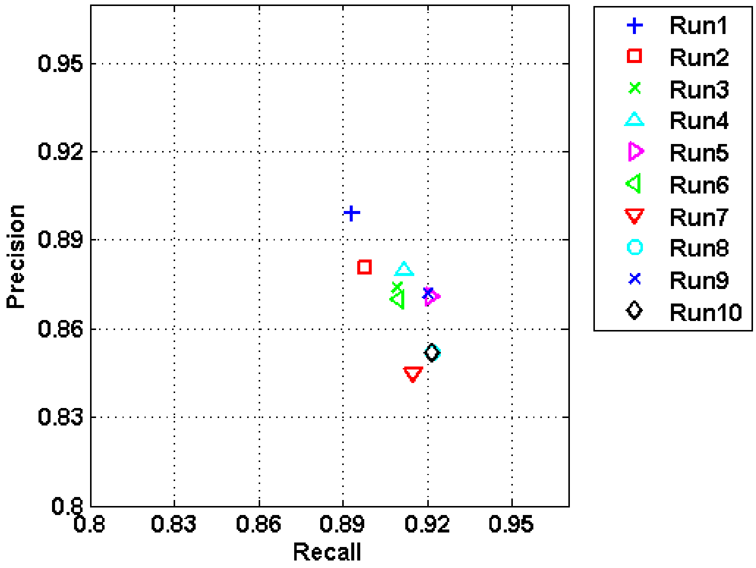
Figure 4.15: Precision/Recall measure for all runs using the *Guassian-L2* kernel function.

also in equilibrate relation recall/precision and similar results, i.e., the system shows similar performance independent of the experiment run. The $L1$ norm (Figure 4.5.3) and the χ^2 dissimilarity (Figure 4.5.3) have high recall values and lower precision values compared to recall. The behavior of cosine dissimilarity (Figure 4.5.3) and histogram intersection (Figure 4.5.3) are the same behavior that we saw with other kernels, i.e., high recall and high precision and all the results are similar. In conclusion, the Gaussian$-\chi^2$ outperforms the results of other kernel functions and again we get the best results using cosine dissimilarity and histogram intersection.

Finally, we evaluate the triangle kernel function. Figures 4.19 and 4.20 present the performance for all the experiment runs using the triangle kernel function and different dissimilarity



[Histogram intersection]

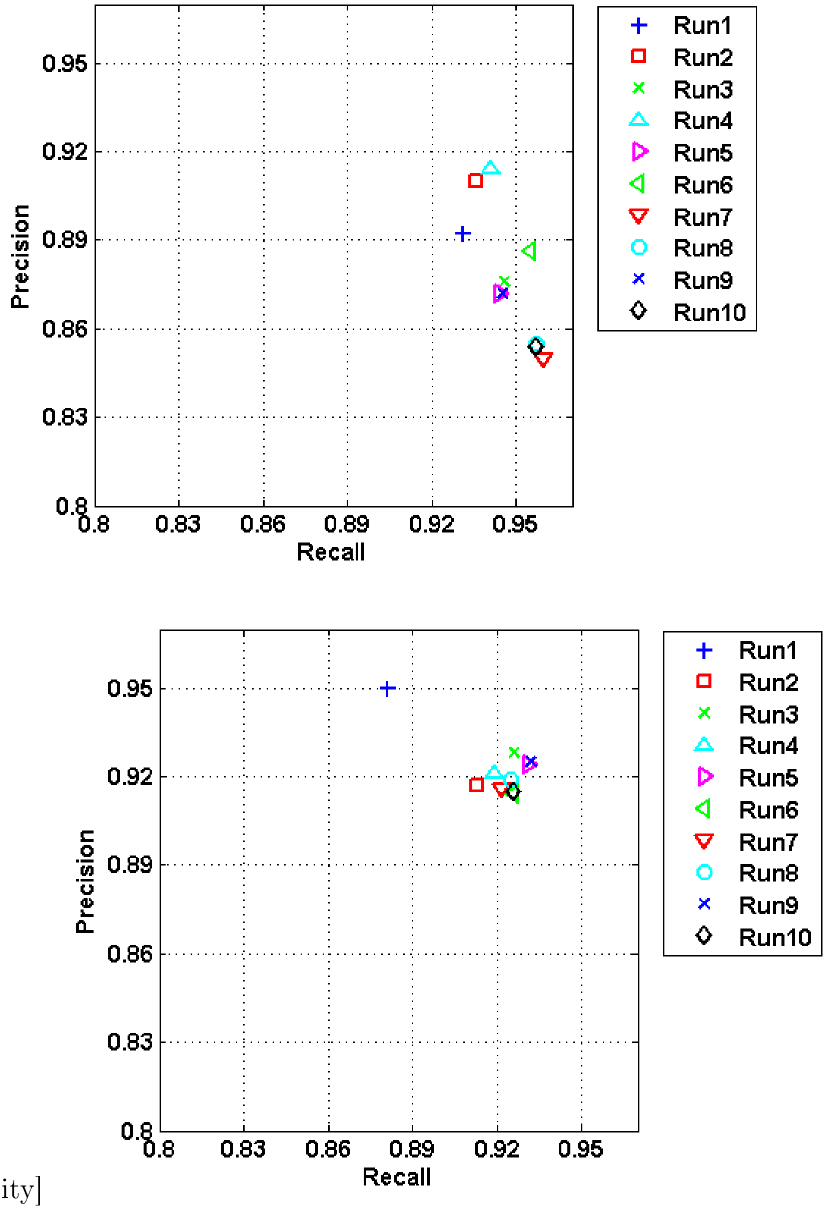


[χ^2 dissimilarity]

Figure 4.16: Precision/Recall measure for all runs using the *Gaussian-L2* kernel function.

measures. Comparing the results with the other kernels function, the triangle kernel function is the second best in performance. When the $L1$ norm is used (Figure 4.5.3), the results are spread and the recall/precision values are better than linear, polynomial and Gaussian-$L2$ kernels. Again the two best dissimilarity measures are the cosine dissimilarity (Figure 4.5.3) and the histogram intersection (Figure 4.5.3). The χ^2 dissimilarity (Figure 4.5.3) has a similar performance than cosine dissimilarity and histogram intersection. In conclusion, the triangle kernel function outperforms the results of linear, polynomial and Gaussian-$L2$ kernel functions. The only one that has a better performance is the Gaussian-χ^2 kernel and the best dissimilarity measures are the cosine dissimilarity and the histogram intersection.

Learning support is robust since with training sets from different camera, from different



[L1 norm]

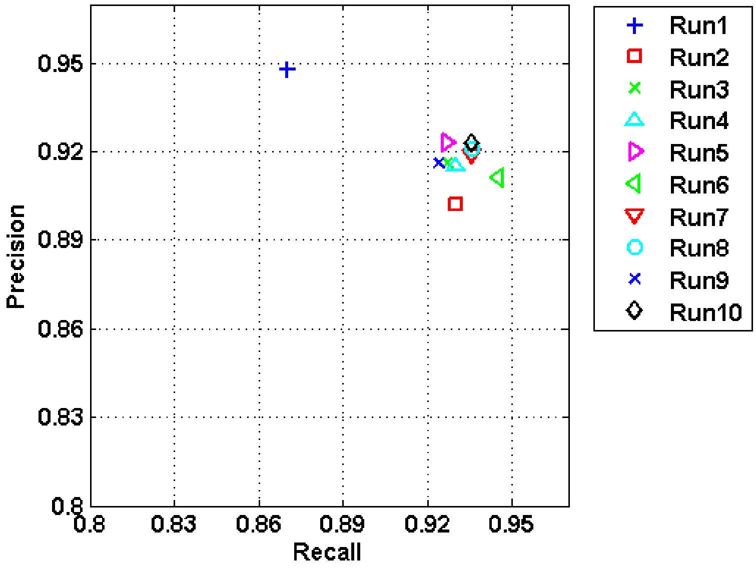
[Cosine dissimilarity]

Figure 4.17: Precision/Recall measure for all runs using the $Gaussian-\chi^2$ kernel function.

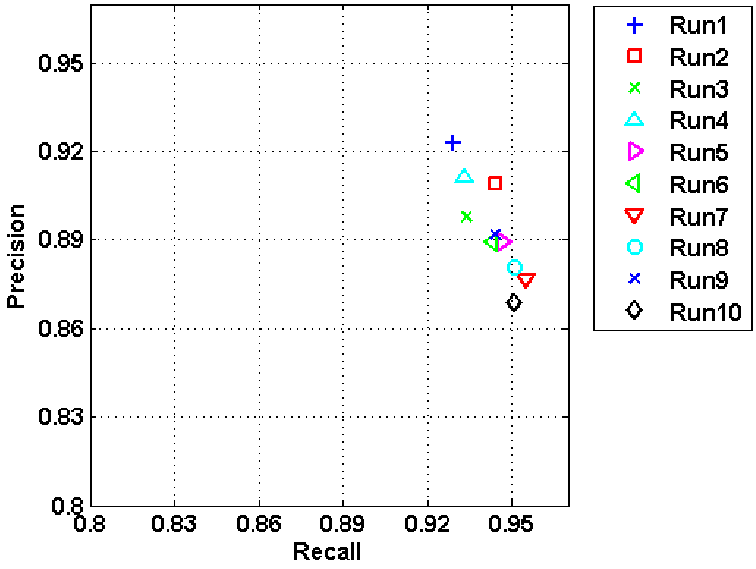
compress format, coding, from different country, situation, the features keep being relevant and stable to detect cuts in different context and environment. We realized different experiments and optimization processes:

Optimization of Kernel Functions

We conducted numerous experiments that provide interesting and meaningful contrast. Table. 4.3 shows the recall, precision and $F1$ measures for the three best similarity measures for each kernel function, we also present the dissimilarity distance used for comparing the feature vectors and the features that were used in each run. The Gaussian-χ^2 kernel provides



[Histogram intersection]



[χ^2 dissimilarity]

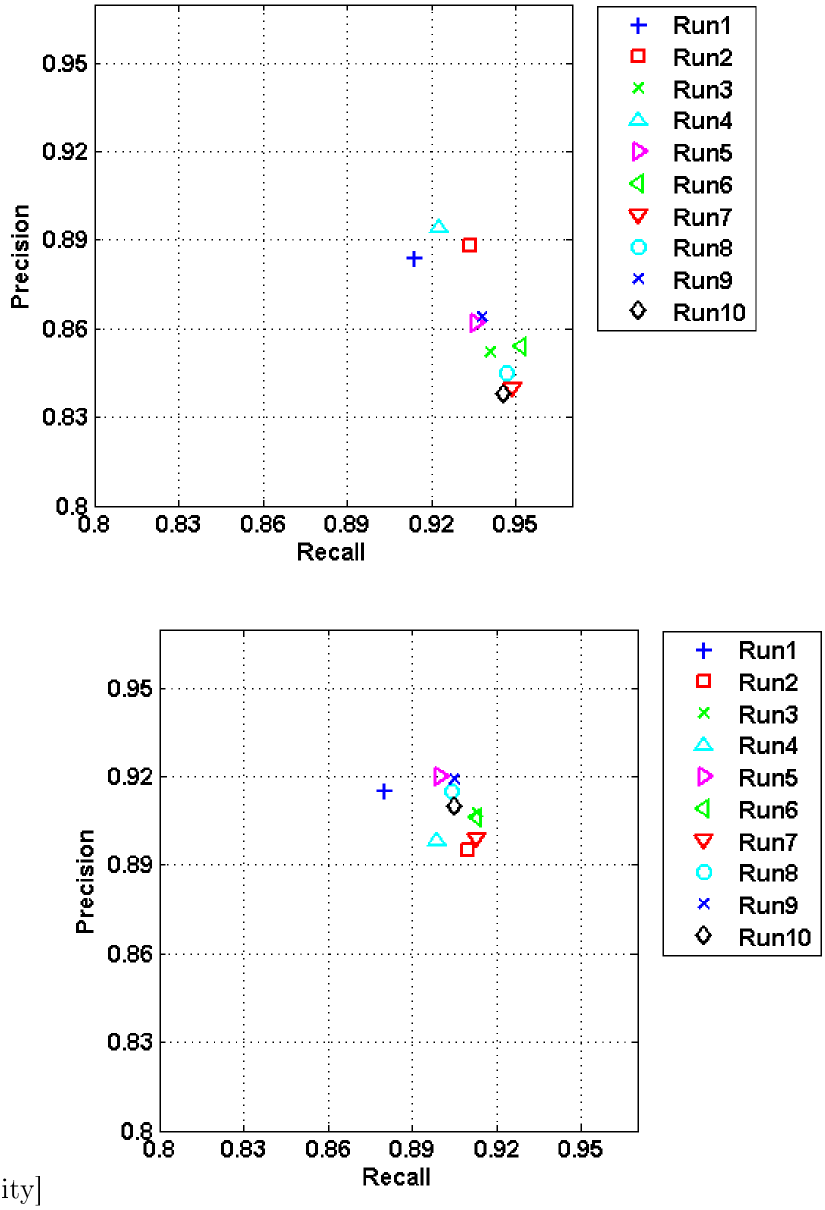
Figure 4.18: Precision/Recall measure for all runs using the *Gaussian*$-\chi^2$ kernel function.

the best results over all the other kernel functions.

Thus, our evaluation of kernel functions confirms that when distributions are used as feature vectors, a Gaussian kernel gives excellent results in comparison to distance-based techniques Gosselin and Cord [2004b].

Optimization of Training Set

In order to reduce the number of support vectors and decrease the time consumed for training and testing we reduce our training set. Instead of using the 5 min. video (c.f. Section 4.5.1) we segment it and train our classifier with a 2 min. video that contains 50 cuts.



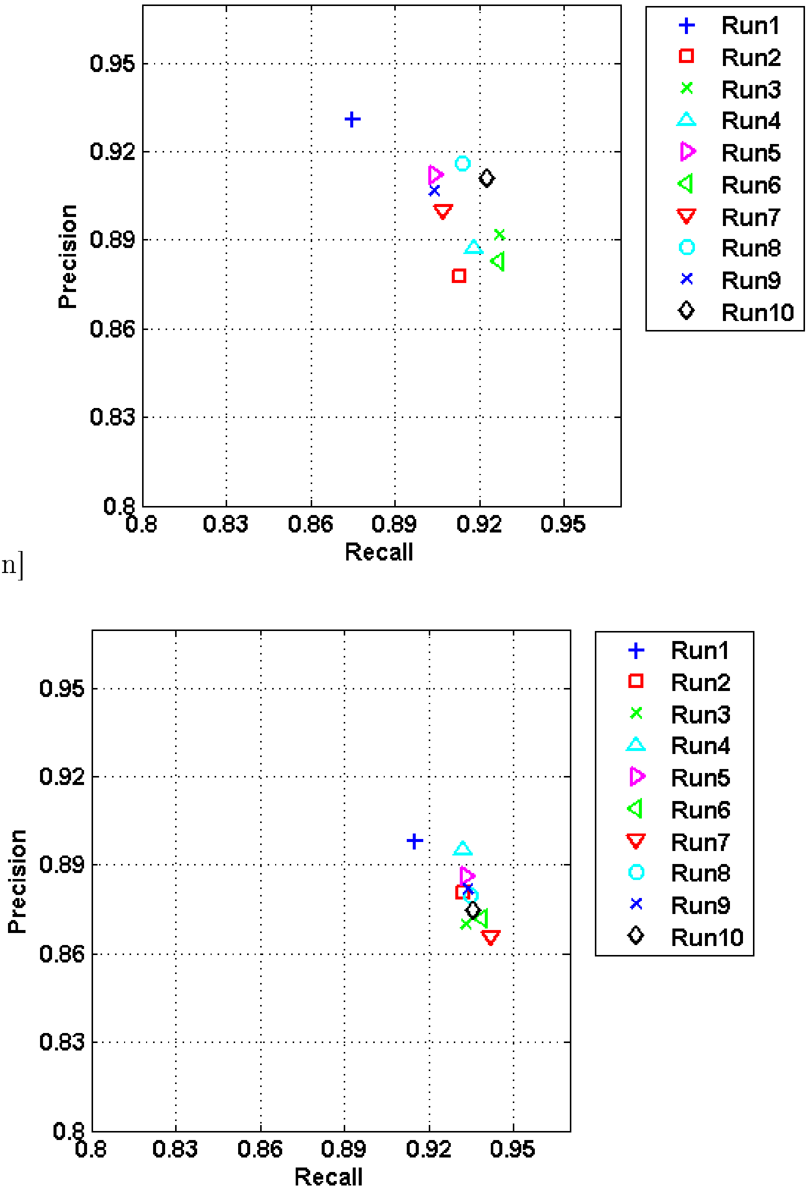
[L1 norm]

[Cosine dissimilarity]

Figure 4.19: Precision/Recall measure for all runs using the *Triangle* kernel function.

The performance of our system maintains its accuracy with the advantage that the steps of training and testing are very fast. In Table 4.4 we show the recall, precision and $F1$ statistics using seven different feature sets.

Based in our previous experiments, we are able to set the choice for the kernel function and the dissimilarity measure. The choice for kernel function is the Gaussian-χ^2 (as it is shown in our experiments, it executes the best performance). The choice of cosine dissimilarity is based on the results of our experiments, this only confirms what Cabedo and Bhattacharjee [1998] have shown in their experiments and they also demonstrate the better performance of cosine dissimilarity. Therefore, we test the performance of our detector using the Gaussian-χ^2 kernel function and the cosine dissimilarity. We evaluate our system with TRECVID-2002 data set,



[Histogram intersection]

[χ^2 dissimilarity]

Figure 4.20: Precision/Recall measure for all runs using the *Triangle* kernel function.

i.e., our ten experiment runs are compared in terms of recall/precision with the results of the teams that participate in the TRECVID-2002 evaluation.

TRECVID 2002 Evaluation

In Table 4.5, we show the performance of our system. All these results, the best ones, are obtained using the Gaussian-χ^2 kernel. We present the recall and precision, its respective variance and the $F1$ measures. The small values of variance show the stability of our system. In Figure 4.5.3, we show the results that were obtained in the official TRECVID-2002 evaluation and compare them with the results of our ten runs, Figure 4.5.3. As shown in the

Kernel	Recall	Prec.	F1	Diss.	Run
Linear	0.913	0.876	0.894	Hist.Int.	10
	0.928	0.860	0.892	Hist.Int.	7
	0.903	0.881	0.892	Cos.	3
Poly	0.896	0.915	0.905	Cos.	7
	0.887	0.924	0.905	Hist.Int.	8
	0.909	0.898	0.903	χ^2	3
Gauss-$L2$	0.909	0.904	0.906	Hist.Int.	8
	0.919	0.889	0.904	$L1$	4
	0.903	0.903	0.903	Cos.	5
Gauss-χ^2	**0.936**	0.923	**0.930**	Cos.	10
	0.932	**0.925**	0.929	Cos.	9
	0.936	0.921	0.928	Hist.Int.	8
Triangle	0.923	0.911	0.917	Cos.	10
	0.914	0.916	0.915	Hist.Int.	8
	0.932	0.895	0.914	χ^2	4

Table 4.3: Measure performance for each kernel function (in Table 4.1, we present the features used in the runs).

Complete Train Set 128			Reduced Train Set 50			
Recall	Prec.	F1	Recall	Prec.	F1	Features
0.92	**0.92**	0.92	0.90	0.93	0.92	HSV_h, Z_h, H_h, Var, PC
0.92	0.92	0.92	0.91	**0.93**	0.92	HSV_h, V_h, H_h, Var, PC
0.93	0.90	0.92	0.93	0.91	0.92	HSV_h, RGB_h, F_h, H_h, Var, PC
0.93	0.91	0.92	0.92	0.92	0.92	HSV_h, Z_h, V_h, H_h, Var, PC
0.94	0.90	0.92	0.93	0.91	0.92	$R-G_h$, HSV_h, F_h, H_h, Var, PC
0.95	0.90	**0.93**	0.93	0.91	**0.92**	HSV_h, RGB_h, F_h, Z_h, H_h, Var, PC
0.94	0.90	0.92	**0.93**	0.91	0.92	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, H_h, Var, PC

Table 4.4: Comparison of performance for 7 feature sets using all training set videos and the reduced training set videos.

figure the accuracy and robustness of our approach is very efficient. Hence, the capacity of generalization of our classifier is proven and the combination of the selected features performs good results without any pre-processing or post-processing.

4.5.4 TRECVID 2006

The test data is composed by news video in Arabic, Chinese and English. The data were collected by Linguistic Data Consortium (LDC) during November and December of 2005, digitized and transcoded to MPEG-1. The test collection comprises about 7.5 hours, including 13 videos for a total size of about 4.64 Gb. The total number of frames is 597,043 and the number of transitions is 3785. The collection contains 1844 abrupt transitions, that represents 48.7% of the total transitions. The reference data was created by a student at NIST whose

Run	Recall	σ_{recall}	Prec.	$\sigma_{\text{prec.}}$	F1	Diss. meas
1	0.929	0.004	0.923	0.010	0.926	χ^2 test
2	0.944	0.003	0.909	0.014	0.926	χ^2 test
3	0.926	0.003	**0.928**	0.007	0.927	Cos
4	0.941	0.003	0.914	0.009	0.927	L1
5	0.931	0.003	0.924	0.007	0.927	Cos
6	**0.945**	0.003	0.911	0.007	0.928	Hist.Int.
7	0.936	0.004	0.919	0.008	0.927	Hist.Int.
8	0.936	0.004	0.921	0.009	0.928	Hist.Int.
9	0.932	0.003	0.925	0.007	0.928	Cos
10	0.936	0.005	0.923	0.007	**0.929**	Cos

Table 4.5: Performance of our system with Gaussian-χ2 kernel function

task was to identify all transitions.

The nomenclature used for the features is as follows: RGB color histogram (RGB_h), HSV color histogram (HSV_h), opponent color histogram ($R-G_h$), Zernike moments (Z_h), Fourier-Mellin moments (F_h), Horizontal project histogram (H_h), Vertical projection histogram (V_h), Phase correlation (PC) and Variance (Var). In Table 4.6, we present the visual feature vectors for cut detection used for the 10 runs.

The experiment runs are compound by the election of the features, kernel function and dissimilarity measure. In the case of kernel function we select the Gaussian-χ^2 our choice for the dissimilarity measure is the cosine dissimilarity.

In Table 4.7, we show the performance of our system for cut detection, measured in recall and precision. We present the recall and precision and its respective variance. The small values of variance shows again the stability of our system.

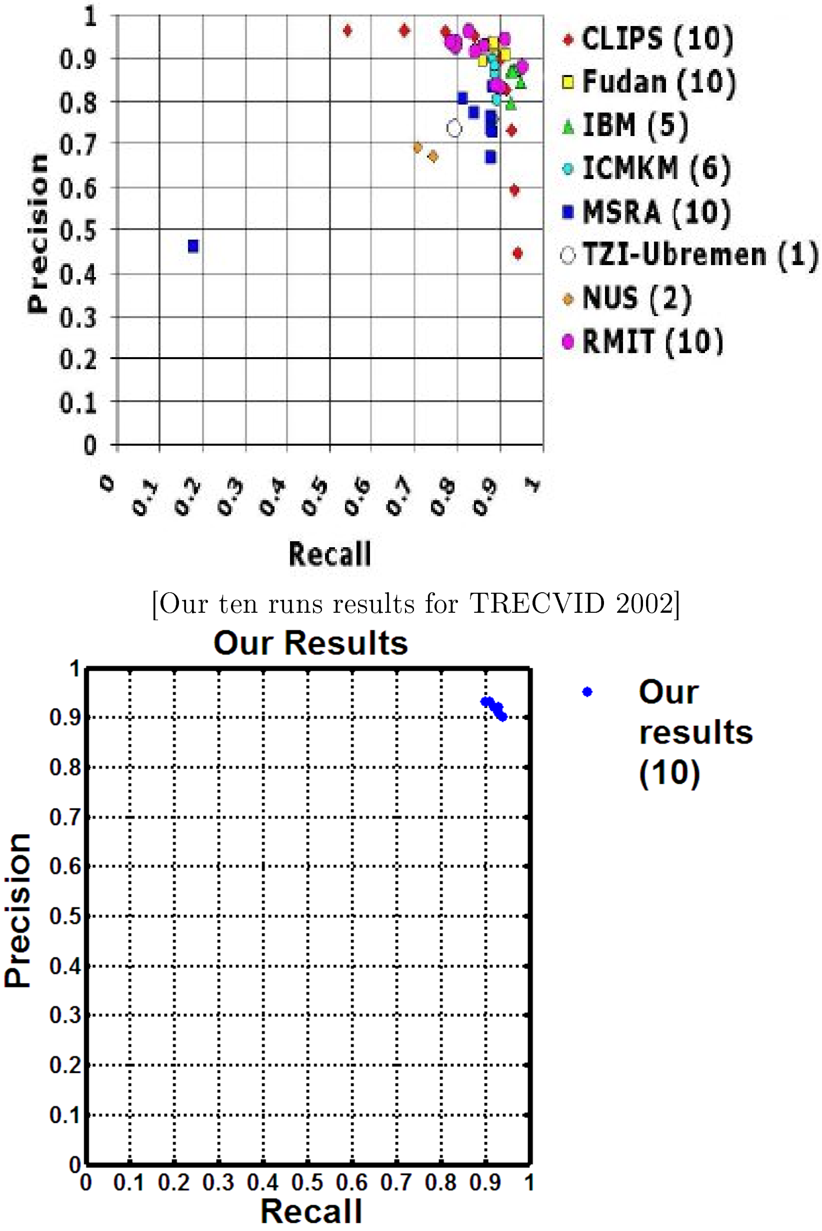
The factor that influence the precision and recall values is related to GTs. In GTs we have three classes: "dissolve", "fade out-in" and "other" transitions. In the case of dissolves, more or less half of them are extremely short (less than 6 frames) and are considered as ATs. Fade-in, fade-out, wipe, "black" frames separating consecutive shots and other type of special effects are included in "other" transitions category. Now, let see how these GTs affect the performance of our AT detector. As short dissolves are considered as ATs, the recall of our system decreases since the recall count the detected transitions from all possible transitions (cuts and short dissolves). In Figure 4.22, we see some examples of "other" class transitions. Our system detects false cuts in the abrupt changes. Thus, the precision values are affected by the false positives detected by our system.

Trecvid 2006 Participants

We made a classification based on the approach used for the participants of TRECVID 2006 Evaluation:

Machine learning approach

[1] Values are calculated in function of $F1$ measure



[Official results for TRECVID 2002 Smeaton and Over [2002]]

[Our ten runs results for TRECVID 2002]

Figure 4.21: Precision/Recall measure of performance

– *AT&T*: cut detector is a finite state machine. For each frame a set of visual features are extracted, these can be classify into two types: intra-frame and inter-frame. The intra-frame features are: color histograms (RGB and HSV), edge and related statistical features (mean, variance, skewness and flatness). The inter-frame features capture the motion compensated intensity matching errors and histograms changes. The HSV color histogram is quantize into 256 bins. Motion features are extracted based on 24 blocks, each with the size 48 × 48 pixels. The search range of motion vector for each block is set to 32 × 32. The motion features include the motion vector, the matching error and the matching ratio. The dissimilarities are

Run	Features
1	HSV_h, Z_h, H_h, Var, PC
2	HSV_h, V_h, H_h, Var, PC
3	HSV_h, RGB_h, F_h, Z_h, Var, PC
4	RGB_h, Z_h, V_h, H_h, Var, PC
5	$R-G_h$, HSV_h, RGB_h, F_h, H_h, Var, PC
6	HSV_h, RGB_h, F_h, Z_h, H_h, Var, PC
7	RGB_h, F_h, Z_h, V_h, H_h, Var, PC
8	HSV_h, Z_h, V_h, H_h, Var, PC
9	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, H_h, Var, PC
10	HSV_h, RGB_h, F_h, Z_h, H_h, V_h, Var, PC

Table 4.6: 10 best combinations of visual features for cut detection

Run	Recall	σ_{recall}	Prec.	$\sigma_{\text{prec.}}$
1	0.821	0.012	0.909	0.003
2	0.825	0.013	0.889	0.003
3	0.818	0.015	0.908	0.003
4	0.827	0.013	0.886	0.003
5	**0.832**	0.012	0.876	0.003
6	0.828	0.012	0.876	0.004
7	0.827	0.014	0.886	0.003
8	0.821	0.014	0.879	0.004
9	0.813	0.014	**0.911**	0.002
10	0.803	0.021	0.868	0.002
Mean Trecvid [1]	0.729	-	0.722	-
Max Trecvid [1]	0.868	-	0.943	-

Table 4.7: Performance of our system with $\chi 2$ kernel function

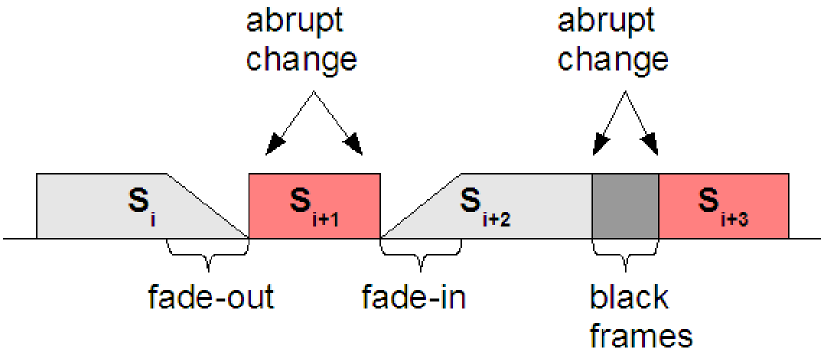


Figure 4.22: Shot transitions

computed between consecutive frames and frame distance of 6 frames. SVM is applied to cut detector to further boost the shot boundary performance.

– *Chinese Academy of Sciences / JDL (CAS/JDL)*: uses a two pass approach, first selects the suspicious transition candidates using a low threshold method and then judges the candidates by using the SVM base method. The features used are histograms and mutual information. Due to the low threshold, the method does not need to extract complex features. The dissimilarity measure used is the $L1$ norm. The drawback of this method is that sometimes the system is not able to differentiate between GTs and object motion.

- *FX Palo Alto Laboratory (FXPAL)*: uses dissimilarity features within the particular temporal interval as the input for kNN classifier. Color histograms in YUV space are extracted, global image histograms and block histograms using a uniform 4×4 spatial grid. The dissimilarity measure used is the χ^2 distance and is computed using a frame distance of 5 and 10 frames.
- *Helsinki University of Technology (HelsinkiUT)*: extracts feature vectors from consecutive frames and project them onto a 2D self-organizing map (SOM). The features extracted are the average color, color moments, texture neighborhood, edge histogram and edge co-occurrence. The frame features are calculated for five spatial zones for each frame of the video. These results are averaged over the frames contained within each one of the five non-overlapping temporal video slices. By this way, a final feature vector that describes the changes of the frames descriptors over time in different spatial areas of the video is calculated. The average color feature vector contains the average RGB of all the pixels within the zone. The color moments feature treats the HSV color channels as probability distributions, and calculates the first three moments. The texture neighborhood feature is calculated from the Y (luminance) component of the YIQ. The 8-neighborhood or each inner pixel is examined, and a probability estimate is calculated for the probabilities that the neighbor pixel in each surrounding relative position is brighter than the central pixel. The feature vector contains these eight probability estimates. Edge histogram, is the histogram of four Sobel edge directions. Edge co-occurrence gives the co-occurence matrix of four Sobel edge directions. Finally the system detects ATs from the resulting SOM. Computationally the most expensive (because of SOMs).
- *Indian Institute of Technology at Bombay (IIT.Bombay)*: proposes a method that reduces the number of false positives caused by dramatic illumination changes (flashes) and shaky camera and fire/explosions. They use a multi-layer filtering to detect candidates based on correlation of intensity features and is further analyzed using a wavelet transform. The correlation used is a normalized mean centered correlation. A high correlation signifies similar frames, probably belonging to the same shot; a low value is an indication of a shot break. To overcome the problem of threshold setting, the system considers the continuity of correlation values rather than the correlation values themselves. The system achieves this using the Morlet wavelet. The Morlet wavelet is a complex sine wave modulated with a Gaussian. The characteristic of this wavelet is that the number of positive and negative values are equal and the area sums zero. When there is no or little change in

the correlation sequence, the wavelet transform returns zero value. If there is a AT, there is a discontinuity in the correlation value, which results in a distinctive PPNN pattern (two positives values followed by two negatives) in the lowest scale. A final filtering step is executed by a trained SVM. The features used in the training SVM are: pixel differences which includes average pixel difference and Euclidean pixel difference, histograms differences (average histogram difference, histogram intersection and χ^2 distance), edge difference, average intensity value, correlation, cross-correlation and maximum of the correlation values, presence of PPNN pattern in the lowest level of the wavelet transform and the lowest wavelet coefficient.

- KDDI and R&D Laboratories (KDDI): compressed domain approach for detecting ATs and short dissolve. Feature parameters are judged by SVM. The features uses are: the number of in-edges and out-edges in divided regions, standard deviations of pixel intensities in divided regions, global and block histograms with Ohata's and RGB color spaces and edge change ratio. The system uses a 2-stage data fusion approach with a SVM. The overview of the data fusion approach is as follows: At the first stage, every adopted feature is judged by a specific SVM. This means the number of feature types is equal to the number of SVMs at the first stage. And the SVM at the second stage synthesizes the judgments from the first stage.
- Tsinghua University (Tsinhgua): cut detector uses 2nd order derivatives of color histogram and pixel-wise comparisons. Features vectors for ATs are constructed based on the graph partition, and then are used to train a SVM. It also has a post-processing module for flashlight detection. The features used are: color histograms of 48 bins in RGB space (16 bins per channel), histogram intersection is adopted to calculate the dissimilarity of two histograms, pixel-wise difference is used as a supplement to color histograms because it introduces spatial information. A threshold method, called second order derivative, is proposed to boost the precision of AT candidates. This scheme eliminates the false positives. To detect flashlight effect and monochrome frame, the mean value and standard deviation of each frame's pixel intensities are also calculated. Abrupt change of illumination can be detected by tracking the variation of mean gray value. Moreover, stable intensities, a prominent characteristic of monochrome frame, can be reflected by small standard deviation feature.
- University of Marburg (Marburg): proposes an unsupervised kmeans clustering for ATs. Two different frame dissimilarity measures are used: motion-compensated pixel differences and color histograms. To detect cuts, two different frame dissimilarity measures are applied: motion-compensated pixel differences of subsequent DC-frames and the histogram dissimilarity of two frames within a predefined temporal distance of 2. A sliding window technique similar is used to measure the relative local height of a peak value. For cut detection, the best sliding window size is estimated by evaluating the clustering quality of "cut clusters" for several

window sizes. Thus, the minimum and maximum sliding window size serves as a parameter for both dissimilarity metrics. Several ranges for this parameter are tested in the experiments for both dissimilarity measures. For cut detection, the unsupervised approach is optionally extended by two classifiers in order to build an ensemble of classifiers. An Adaboost and an SVM classifier is incorporated in that ensemble of classifiers. The features uses are: motion compensated pixel differences, histogram differences, luminance mean and variance, edge histograms of Sobel-filtered DC-frames, local histogram differences and ratio of the second largest dissimilarity value divided by the local maximum for several sliding window sizes.

- Tokyo Institute of Technology (TokyoInstTech): proposes a supervised SVM classifier for AT and short GT detection. For the cut detection, two linear kernel SVMs (one for ATs and the other for short GT) with different feature sets are used. The features for a AT detection are activity ratio (the ratio of "dynamic" pixels to all pixels, where each dynamic pixel has larger difference than a predetermined threshold), the optical flow, the change in the Hue-Saturation color histogram and edge. The features for short GT detection are the activity ratio and the change in the Hue-Saturation color histogram. Linear kernel functions are used for both systems.

Threshold-based approach

- *Artificial Intelligence and Information Analysis (AIIA)*: uses mutual information as a similarity metric. The mutual information between two frames is calculated separately for each of the RGB color components. The mutual information corresponds to the probability that a pixel with gray level i in frame f_t has gray level j in frame f_{t+1}. The mutual information is not calculated between all pair of frames, because relations between frames, which are far apart are not important for the AT detection. Thus, the method uses only mutual information calculated between frame in a sliding temporal window (30 frames). Then a cumulative measure which combines information from all these frame pairs is calculated. Mutual information calculated between consecutive frames provides easily detectable peaks. The threshold for detection of the transition is set empirically.
- *Chinese University of Hong Kong (CityUHK)*: applies adaptive thresholding on color histograms (RGB and HSV color spaces) and gray-level histogram differences. The system uses Euclidean distance, color moments, and Earth Mover's Distance (EMD) measures to calculate color differences. The former two performed rather poorly as they are under-sensitive to true positives but over-sensitive to false-positives. The EMD method, however, is able to produce better results, as it is sensitive to most transition-like changes. Though it also produce more noise than the other two measures, this is not problematic when adaptive thresholding is applied. The adaptive threshold is calculated within a sliding window of 11 frames.

- *Communication Langagière et Interaction Personne-Système (CLIPS)*: detects ATs by image comparisons after motion compensation. Pre-process operations like motion compensation and filtering process like photographic flash are applied. The system has several thresholds that have to be tuned for an accurate detection. Direct image difference is the simplest way for comparing two images and then to detect ATs. Such difference however is very sensitive to intensity variation and to motion. This is why an image difference after motion compensation is being used. Motion compensation is performed using an optical flow technique which is able to align both images over an intermediate one. This particular technique has the advantage to provide a high quality, dense, global and continuous matching between the images. Once the images have been optimally aligned, a global difference with gain and offset compensation is computed. Since the image alignment computation is rather costly, it is actually computed only if the simple image difference with gain and offset compensation alone has a large enough value (i.e. only if there is significant motion within the scene). Also, the differences are computed on reduced size images. A possible cut is detected if both the direct and the motion compensated differences are above an adaptive threshold. Filtering process like photographic flash are applied. The flash detection is based on an intensity peak detector which identify 1- or 2-frame long peaks on the average image intensity and a filter which uses this information as well as the output of the image difference. A flash is detected if there is a corresponding intensity peak and if the direct or motion compensated difference between the previous and following frames are below a given threshold.

- *European Cooperation in the Field of Scientific and Technical Research (COST292)*: transitions are detected by merging the results of two shot boundary detectors. The first detector is based on the extraction of the relevant features from spatiotemporal image blocks and modeling those features to detect and identify a vast range of transition and an abundance of graphical effects. The extracted features are mainly related to the behavior of luminance values of pixels in the blocks. Further, as the features used and the processing steps performed are rather simple, the proposed method is computationally inexpensive. Video data is defined as a three dimensional discrete function of luminance values: horizontal and vertical frame dimensions and the length of the video. To perform a 3D analysis on the data, overlapping spatiotemporal data blocks are define. There exists a temporal overlap factor. Some statistics are computed from this blocks, if these values are bigger than a threshold an AT is detected. The second detector works directly on compressed video only in I/P resolution. The shot boundary detector works separately on I-frames and P-frames. The detection on P-frames is based on the temporal difference of intra-coded macroblocks and the variation of global motion parameters. The detection method for I-frames reuses the global motion models of the shot boundary detection on P-frames. It is used to calculate the histogram intersection of the DC image of the current I-frame and the motion compensated DC image of the previous I-frame. In order to detect an AT, the values of the histogram intersection are thresholded. The merging is performed under the basic assumption that the first detector achieves a higher precision and recall, since the second works in the compressed domain only in I/P resolution. For each detector, the shot boundary detection results are characterized by a confidence measure. In the merging process, both confidence measures are used and privilege the first detector.

- *Dokuz Eylol University (Dokuz)*: is based on color histograms differences (RGB color space) for AT detection. Color histograms are quantize into 27 bins. Then a Euclidean distance of histogram belonging to two consecutive frames are calculated. The method uses a threshold value for AT detection and a *skip frame interval* to skip ahead 5 frames for eliminating consecutive frames that have much redundant information. The detection is based on a threshold.

- *Institute of Informatics and Telecommunications National Center for Scientific Research "Demokritos" (ITT/NCSR Demokritos)*: a two step process is executed in order to detect ATs and eliminate false detections produces by flashlights. The feature set consists of combinations of RGB color, adjacent RGB color, center of mass and adjacent gradients. In the first step candidate ATs are detected applying a threshold, the second step is a flashlight filter. A modeling of an AT in terms of the Earth Mover's Distance (EMD) is introduced. For any candidate boundary a set of similarities based on EMD between the current frame and each of the 5 previous frames are computed. The objective is to get a spatiotemporal template in order to express a linear dissimilarity that decreases in time.

- *RMIT University (RMIT)*: the system consists of a two-pass implementation of a moving query window algorithm. The content of each frame is represented by two types of histograms: local and global. Local color histograms in the HSV color space are extracted from 16 equal-sized regions in each frame. For each region, separate histogram with 32 bins per color component is computed. Two, three-dimensional global HSV histograms, where each color is represented as a point in a three-dimensional space. For both type of histograms the Manhattan distance is used as dissimilarity measure. For AT detection the system uses the techniques of query-by-example and ranked results. The moving window extends equally on either side of the current frame, but not including the current frame itself. The, the current frame is used as a query on the collection of frame within the window. The frames forming the preceding half window are referred as pre-frames, and the frames that following the current frame are referred as post-frames. The behavior of the algorithm is controlled through the following parameters: *half window size*, number of frames on one side of the query window; *demilitarized zone depth*, specifies the number of frames (size of the gap) on each side of the current frame which are not evaluated as part of the query window; *lower bound*, this is the lower threshold used for AT detection and *upper bound*, this upper threshold is used for AT detection in connection with the lower bound.

To detect and AT, the number of pre-frames are monitored in the $N/2$ results as each frame is examined, where N is the size of the window. When the sliding window goes closer to an AT, the number of pre-frames rises quickly and passes the upper bound. Once it pass the transition, the number of frames falls sharply below the lower bound. The slop reflects this by taking on a large positive value, followed quickly by a large negative. The drawback with the system consists in determinate the size of the window, which is critical. They use a dynamic threshold based on the information of previous frames.

– *University of Modena (Modena)*: examine frame differences behaviors over time to see if it corresponds to a linear transformation. The approach is strictly focus on GTs with linear behavior, including ATs. The detection is based on the fitness of the data to a linear model. The length of the transition distinguishes an AT from a GT.

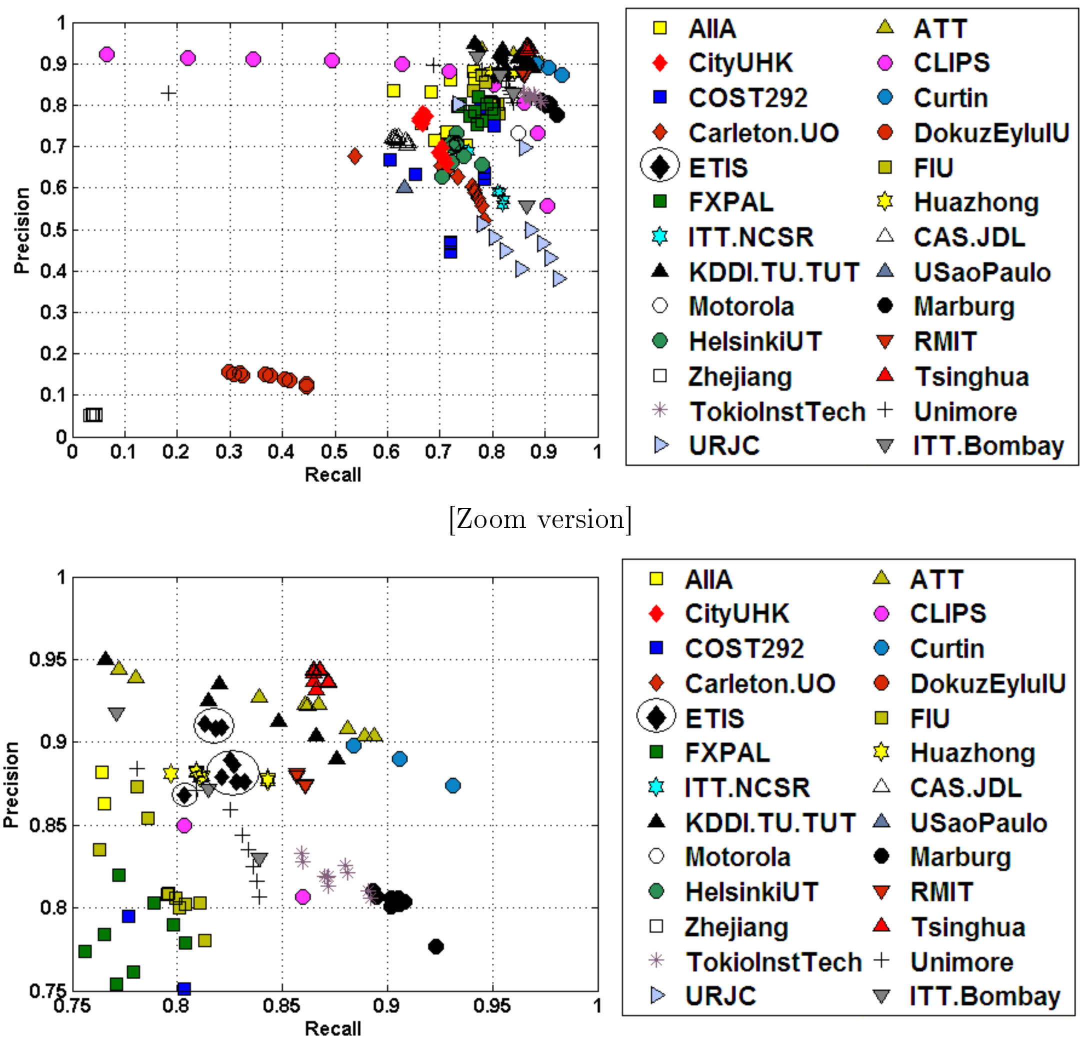
– *Carleton University (Carleton.UO)*: approach based on tracking image features across frames. ATs are detected by examining the number of features successfully tracked (and lost) in adjacent frames, refreshing the feature list for each comparison. The features used are corners of edges on gray scale frames and requires registration of corner features across frames. In the case of a cut at frame f, all features being tracked should be lost from frame f_{t-1} to f_t. However, there are often cases where the pixel areas in the new frame coincidentally match features that are being tracked. In order to prune these coincidental matches, the minimum spanning tree of the tracked and lost feature sets are examined. The inter-frame difference metric is the percentage of lost features from frame f_{t-1} to f_t. This corresponds to changes in the minimum spanning tree. The system needs automatic thresholding to adjust to video type. The auto-selection of a threshold will be done by examining the density distribution of the lost features over the entire sequence.

– *University of São Paulo (USP)*: propose a two step process. First, compute absolute pixel differences between adjacent frames and detect any type of large discontinuity or activity in pixels. Frames are considerate as gray scale images. If the difference is bigger than a threshold then it is consider as an event point. The second one is a refinement, looking for shot boundaries only. Parameters (window size and thresholds) are set experimentally. Designed for AT detection only.

– *University Rey Juan Carlos (URJC)*: use color histogram (16 bins) and shape descriptors as Zernike moments of order 3. They vary the weighed combinations and find a fusion approach that improve the accuracy on the independents in isolation. The confidence is measured based on the difference computed between the current frame and a window of frames. A candidate for AT is detected when the values are higher than a dynamically computed threshold.

There are no information available for *Curtin University*, *Florida University(FIU)*, *Huazhong University of Science and Technology*, *Motorola* and *Zhejiang University* systems.

In Figure 4.23 we show the results that were obtained in the official TRECVID-2006 evaluation. Hence, the capacity of generalization of our classifier is proven and the combination of the selected features performs good results without any pre-processing or post-processing step. The data of TRECVID-2006 are more complex, making more difficult the task of shot boundary detection, this can be seen comparing with the results of previous years.


[All results]

[Zoom version]

Figure 4.23: Precision/Recall measure of performance on the TRECVID 2006 for cut detection

The best results are achieved by AT&T, Tsinghua, Curtin and KDDI systems. The first, second and fourth are machine learning approaches. Unfortunately, we do not have any information about Curtin system. These teams are being participating in Trecvid evaluation for many years. Thanks to that, they could improve year by year their methods. The two best systems (ATT and Tsinghua) based their approaches in finite state machines and the results are improved by SVM. Tsinghua also has several post-processing filters, which let the system eliminate false positives. KDDI system is a SVM-based detector, for cut detection they use 6 SVMs for different type of features and then combine them at the end. This method is a type of bagging technique[2] Breiman [1996], which aloud to improve machine learning methods. Other machine learning methods that have similar performance to our method are ITT.Bombay and Marburg systems. Even though ITT.Bombay (SVM-based method) has post-processing operations, their results are similar to ours with no pre-processing and post-processing operations. And the other machine learning approaches, CAS/JDL, FXPAL and HelsinkIUT show worst performance than our system.

We can conclude that the best methods for shot boundary detection in TRECVID Evaluation are the machine learning approaches. They can deal with many features, eliminate the threshold setting and can also use an ensemble of classifiers in order to improve its accuracy.

4.6 Conclusion

In this Chapter, we addressed the problem of temporal video segmentation. Classical methods like static thresholding approaches have the drawback of manual fine tuning of detection parameters, i.e., select an appropriate threshold for different kind of videos. These methods only performs well if video content exhibits similar characteristics over time. Methods with adaptive thresholds were proposed in order to overcome the problem of threshold setting, but these approaches add new problems like defining the size of sliding windows where the adaptive threshold is evaluated. Thus, in order to overcome this problem we consider AT detection from a supervised classification perspective.

Previous detecting cut classification approaches consider few visual features because of computational limitations. As a consequence of this lack of visual information, these methods need pre-processing and post-processing steps, in order to simplify the detection in case of illumination changes, fast moving objects or camera motion. We evaluate different visual features and dissimilarity measures with the objective to build an automatic and free-parameter AT detector.

We focus on improving existing algorithms for AT detection. We evaluate the characteristics of different visual features. Since our objective is to avoid pre-processing and post-processing steps, we consider features that let our system deal with abrupt illumination changes and motion compensation. Features like phase correlation and color histograms in the opponent color space are more robust to abrupt illumination changes. As color histograms does not consider spatial information are more robust to camera/object motion. Therefore, we consider different features with the objective to use the capabilities of the features and also to overcome the weakness of them and our system let us merge the features.

[2] Ensemble of classifiers has better accuracy than the single classifier that composes the ensemble

We improve the accuracy of phase correlation method and propose to use entropy as the *goodness-of-fit* measure in block-based correlation coefficients. The advantage of our method is that it considers the global information of the block instead of a single element of the block as it was proposed by other author. We also evaluate different dissimilarity measures: $L1$ norm, cosine dissimilarity, histogram intersection and χ^2 test. In our approach the cosine dissimilarity and histogram intersection show the best performance. Kernel functions were also evaluated by our kernel-based system. We consider 5 different kernel functions: linear, polynomial, Gaussian-$L2$, Gaussian-χ^2 and triangular kernels. The Gaussian-χ was the kernel that showed the best performance followed by triangle kernel. Both kernel functions show a equilibrate relation recall/precision, getting high values both measures.

We used the TRECVID-2002 and TRECVID-2006 data sets. The former was used to compare, evaluate and define the different feature sets, dissimilarity measures and kernel functions. The latter was used to compare our approach with other approaches, i.e., we participated in the TRECVID Evaluation of 2006. Even though the performance of our AT detector was affected by some type of GTs, we can claim that we are among the best teams in shot boundary task.

Chapter 5

Gradual Transition Detection

Gradual transition detection could not be based on the same assumption of ATs (high similarity between frames corresponding to the same shot and low similarity between to frames corresponding to two successive shots), since similarity is also high in GTs. Unlike ATs, the inter-frame difference during a GT is small. The main problem of detecting GTs is the ability to distinguish between GTs and changes occurred by motion of large objects or to camera operations. GTs are often used at scene boundaries to emphasize the change in content of the video sequence. The purpose of this chapter is to present our approach for GTs, specifically for fade out-in and dissolve detection.

5.1 Introduction

There has been a small amount of work on detecting GTs, because it is a much harder problem. Usually, GTs manifest themselves as gradual increase in the frame differences over a relatively long sequence of frames. Different methods have been created to detect the prolonged increase in frame difference during a GT. However, false detections due to camera operations or object motions need to be prevent because they are also characterized by similar increase in the frame differences. All of these approaches have relied directly on intensity data.

The number of possible GTs is quite large. Well-known video editing programs such as Adobe Premiere[1] or Ulead MediaStudio[2] provide more than 100 different and parameterized types of edits. In practice, however, 99% of all edits fall into one of the following three categories: cuts, fades, or dissolves Lienhart [1999]. Therefore, in the following, we concentrate on fades and dissolves in the case of GT detection.

In Figure 5.1, we present an overview of our method for GT detection. We adopt a hierarchical approach, where in a first stage we detect the boundaries of the ATs. We also need to detect the boundaries of fade transitions. This first stage is important because we search for dissolves in the video once the sequence is segmented into cut-free and fade-free segments.

[1] Available: http://www.adobe.com/products/premiere/

[2] Available: http://www.ulead.com/msp/runme.htm

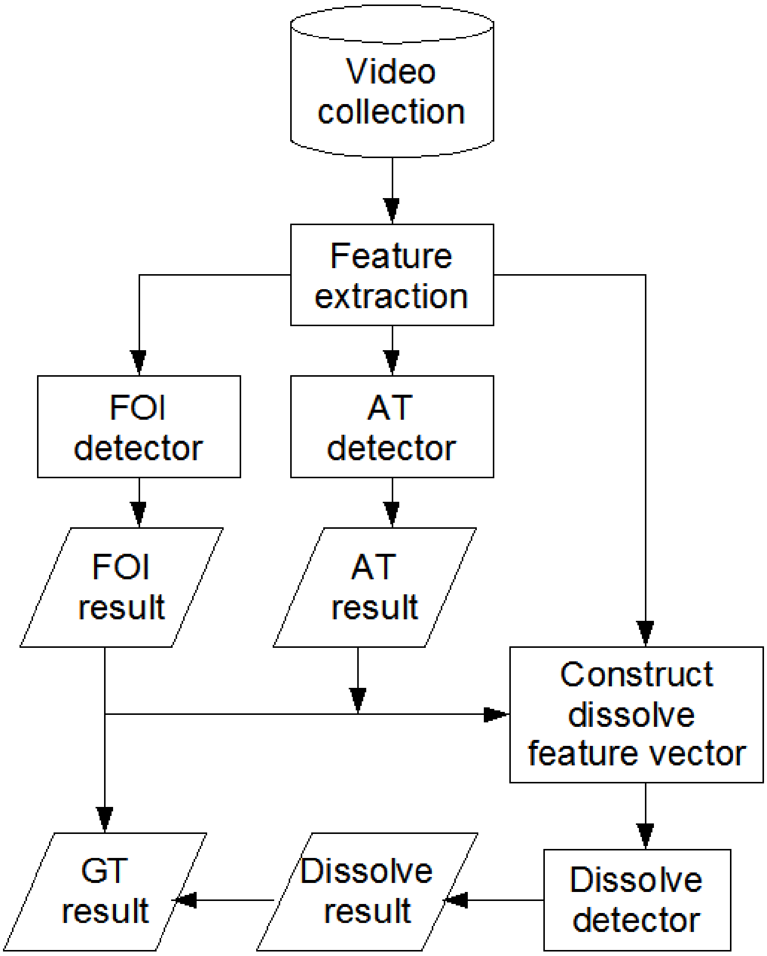


Figure 5.1: General framework for GT detection.

Before we present an overview of our approaches for dissolve and fade out-in detection, let remember the definitions of dissolves and fades (for simplification we omit frame coordinates). The dissolve is characterized by a progressive change of a shot P into a shot Q with non-null duration,

$$f(t) = \alpha(t) \times P(t) + (1 - \alpha(t)) \times Q(t) \qquad t_1 \leq t \leq t_2 \tag{5.1}$$

where α is a decreasing function during the gradual scene change with $\alpha(t_1) = 1$ and $\alpha(t_2) = 0$, t represents temporal dimensions and $t_2 - t_1$ is the duration of the transition.

A fade-out is characterized by a progressive darkening of a shot P until the last frame becomes completely black,

$$f(t) = \alpha(t) \times P(t) + (1 - \alpha(t)) \times G \qquad t_1 \leq t \leq t_2 \tag{5.2}$$

where G is a monochromatic frame and α has the same characteristics that in dissolve transition.

A fade-in is characterized by a progressive appearing of shot Q. The first frame of the fade-in is a monochromatic frame G,

$$f(t) = \alpha(t) \times G + (1 - \alpha(t)) \times Q(t) \qquad t_1 \leq t \leq t_2 \tag{5.3}$$

We can observe that fade-out (Eq. 5.2) and fade-in (Eq. 5.3) are special cases of dissolve transition (Eq. 5.1). We base our GT detector in the fact that fade-in and fade-out transitions are special cases of dissolve transitions.The approach we use is based on the detection by modeling Brunelli et al. [1999], that consists in formulating mathematical models of edited transitions and use these models to design the feature vector and identifying them within the video. These models use the luminance variance for characterizing the dissolve and the fade out-in transitions.

Our dissolve detector consists of the following steps:

1. *Features for dissolve modeling*: We present the luminance variance and the gradient magnitude of the frame, both features show a similar behavior, i.e., the transitions can be approximate by a parabola, see Figure 5.2. We present these features in Section 5.2.1;

2. *Candidate dissolve regions detection*: In this stage, we detect all the intervals where the previous features describe a downward parabola. This include true dissolves and object/camera motion that produce the same effect of dissolves. We present this stage in Section 5.2.2;

3. *Verification of candidate regions*: We filter most of the false dissolves using the dissolve modeling error that we present in Section 5.2.3;

4. *Dissolve features*: In this stage, we extract different features from the candidate regions that lately will be used to train a classifier. We present different methods for dissolve detection in Section 5.2.4 and we also improve a well-known method.

5. *Machine Learning*: In this last stage, we train a SVM classifier with features extracted in the previous stage. We present our machine learning approach in Section 5.2.5.

For the fade out-in detection we exploit the fact that fades are special cases of dissolve transition and propose a method based on the improved method that we used for dissolve detection. In Section 5.3, we present our method for fade out-in detection. We use a threshold-based approach for this method since we only need to set an unique parameter.

As we did with AT detection we test our GT detector on TRECVID data sets of 2002 and 2006. The first data set (2002) was used to test the different kernel functions of our classifier. The second data set (2006) was used to compare the performance of our method with other methods. These results are presented in Section 5.4. Finally, we discuss our conclusion in Section 5.5.

5.2 Dissolve Detection

In Zhang et al. [1993], they use a twin threshold mechanism based on histogram difference metric. Zarih et al. [1996] have used a measure based on the number of edge changes for detecting editing effects, also for cut detection. This method requires global motion compensation before computing dissimilarity. Low precision rate and time-consuming are the drawbacks of this technique. Another feature that is commonly used for dissolve detection is intensity (luminance) variance. During a dissolve transition, the intensity curve forms a downwards-parabolic shape, see Figure 5.2. Alattar [1993] proposes a variance-based approach, many other researchers have used this feature to build their dissolve detectors Hanjalic [2002]; Truong et al. [2000a]. Alattar [1993] suggests to take the second derivative of intensity variance, and then check for two large negative spikes. Again object/camera motion and noise make difficult the dissolve detection (spikes are not too pronounced due to motion and noise). Truong et al. [2000a] propose an improved version with more constraints. Won et al. [2003] suggest a method based on the analysis of a dissolve modeling error that is the difference between an ideally modeled dissolve curve without any correlation and an actual variance curve with a correlation. Other researches based on correlation are Campisi et al. [2003]; Han and Kweon [2003]. Nam and Tewfik [2005] use B-spline polynomial curve fitting technique to detect dissolves. The main drawback of these approaches lies in detecting different kind of transitions with a unique threshold. We want to get rid of the threshold setting as much as possible.

First, we present the dissolve model in more details because we are going to use it in the next sections. The dissolve is characterized by a progressive change of a shot P into a shot Q with non-null duration. Each transition frame can be defined by

$$f(x,y,t) = \alpha(t) \times P(x,y,t) + \beta(t) \times Q(x,y,t) \qquad t_1 \leq t \leq t_2 \tag{5.4}$$

where α is a decreasing function during the gradual scene change with $\alpha(t_1) = 1$ and $\alpha(t_2) = 0$; and $\beta(t)$ is a increasing function with $\beta(t_1) = 0$ and $\beta(t_2) = 1$; x, y and t are continuous variables that represent the horizontal, vertical and temporal dimensions, respectively. In the following discussion, we made two assumptions:

$$\alpha(t) + \beta(t) = 1 \tag{5.5}$$

$$P(x,y,t) = P(x,y) \text{ and } Q(x,y,t) = Q(x,y). \tag{5.6}$$

The second assumption Eq.(5.6) is that, during those transitions, no violent object and camera motion happen. In fact, most GTs satisfy this assumption.

5.2.1 Features for Dissolve Modeling

We use luminance variance and the effective average gradient for modeling a dissolve transition. In both cases, we search for a pronounced downward parabola. In the following sections we demonstrate that both features performs a parabola effect when a dissolve transition occur.

Luminance Variance Sequence

Considering Eqs. (5.4)(5.5) and (5.6), the mean of image sequence during dissolve can be expressed as:

$$E(f(t)) = \alpha(t)E(P) + \beta(t)E(Q) \tag{5.7}$$

and (5.6) the variance of $f(x,y,t)$ within the dissolve region can be expressed as the following equation:

$$\begin{aligned} \sigma_f^2(t) &= E(f-\bar{f})^2 \\ &= E[\alpha(t)P(x,y) + \beta(t)Q(x,y) - \alpha(t)\overline{P(x,y)} - \beta(t)\overline{Q(x,y)}]^2 \\ &= \alpha^2(t)\sigma_P^2 + \beta^2(t)\sigma_Q^2 + 2\alpha(t)\beta(t)E[(P(x,y) - \overline{P(x,y)})(Q(x,y) - \overline{Q(x,y)})] \end{aligned} \tag{5.8}$$

where $\alpha(t) + \beta(t) = 1$.

If $P(x,y)$ and $Q(x,y)$ are assumed to be statically independent with variances σ_P^2 and σ_Q^2, respectively, then the covariance between $P(x,y)$ and $Q(x,y)$ is zero. Therefore Eq. (5.8) is approximated as following:

$$\begin{aligned} \sigma_f^2(t) &\approx \alpha^2(t)\sigma_P^2 + \beta^2(t)\sigma_Q^2 \\ &= [\sigma_P^2 + \sigma_Q^2]\alpha^2(t) - 2\sigma_Q^2\alpha(t) + \sigma_Q^2 \end{aligned} \tag{5.9}$$

Eq. (5.9) shows that the variance $\sigma_f^2(t)$ for dissolve can be approximated by a parabola, see Figure 5.2(a). Considering the middle of the parabola ($\alpha(t) = 0.5$) in Eq. (5.9), the variance of an ideal parabola model $\tilde{\sigma}_{center}^2$ is defined as:

$$\tilde{\sigma}_{center}^2 = \frac{\sigma_p^2 + \sigma_q^2}{4}. \tag{5.10}$$

In this subsection, we describe the first feature used for finding candidate regions. The candidate regions are extracted using the first and second derivatives of the variance curve.

Effective Average Gradient (EAG)

The local edge magnitude can be computed by

$$G^2(t) = (G_x^2(t) + G_y^2(t)) \tag{5.11}$$

where G_x is the gradient on horizontal direction and G_y is the gradient on vertical direction.

Using the video edit model Eq. (5.4)

$$G_x(t) = \alpha(t)G_x(p(x,y)) + \beta(t)G_x(q(x,y))$$

$$G_y(t) \quad = \quad \alpha(t)G_y(p(x,y)) + \beta(t)G_y(q(x,y)) \tag{5.12}$$

Let $TG^2(t) = \sum_{x,y} G^2(t)$, then

$$\begin{aligned} TG^2(t) &= \sum_{x,y}\Big((\alpha(t)G_x(p(x,y)) + \beta(t)G_x(q(x,y)))^2 \\ &\quad +(\alpha(t)G_y(p(x,y)) + \beta(t)G_y(q(x,y)))^2\Big) \\ \beta(t) &= 1 - \alpha(t) \end{aligned} \tag{5.13}$$

Considering $\sum_x G_x(p)G_x(q) \approx 0$, $\sum_y G_y(p)G_y(q) \approx 0$

$$TG^2(t) \approx (TG^2(p) + TG^2(q))\alpha^2(t) - 2TG^2(q)\alpha(t) + TG^2(q) \tag{5.14}$$

Thus, as for intensity variance, the gradient magnitude of image sequence during dissolve also shows parabolic shape.

It is notable that effective average gradient (EAG) can be used for the same purpose. EAG is defined by the following equation:

$$EAG = \frac{TG}{TP} \tag{5.15}$$

where $TG = \sum_{x,y} G(x,y)$ is the total magnitude of the gradient image, and $TP = \sum_{x,y} F(x,y)$ is the total number of pixels with non-zero gradient values, as $F(x,y)$ is defined by

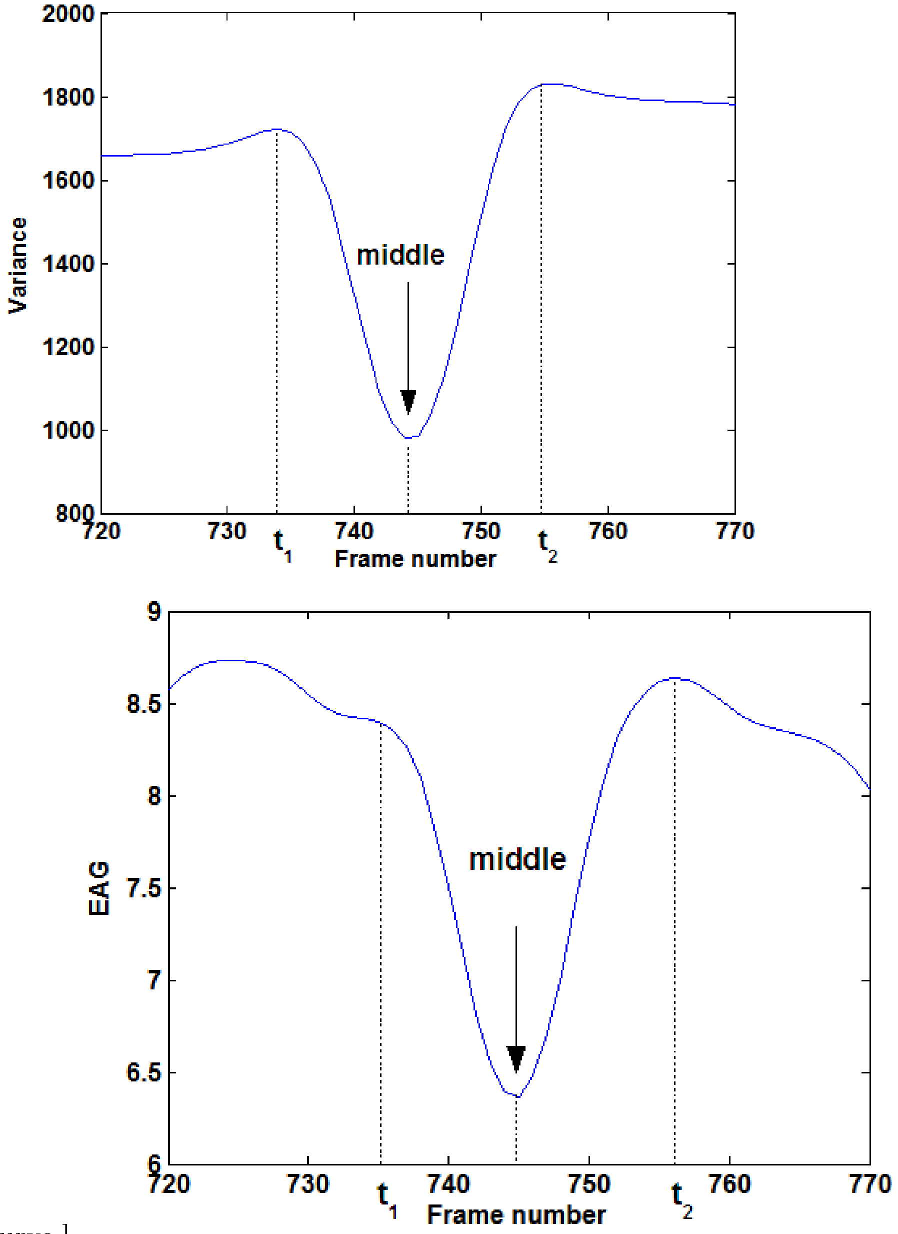
$$F(x,y) = \begin{cases} 1 & \text{if} \quad |G(x,y)| > 0 \\ 0 & \text{if} \quad |G(x,y)| = 0 \end{cases} \tag{5.16}$$

As the EAG also shows a parabolic shape in presence of dissolve (see Figure 5.2)(b), it is possible to extend Eq. (5.10) and to consider again the middle of the parabola, ($\alpha(t) = 0.5$) in order to define the variance of an ideal parabola model $\widetilde{EAG^2}_{center}$ is defined as:

$$\widetilde{EAG^2}_{center} = \frac{EAG^2(p) + EAG^2(q)}{4}. \tag{5.17}$$

Clearly, when variance or gradient magnitude of an image situated at the beginning or at the end of the transition is low, the valley of parabola becomes less distinct, i.e., the parabola becomes less pronounced. We present an example in Figure 5.3 where the luminance variance curve and the gradient magnitude curve of the same interval of a video sequence is presented. The parabolic valleys in Figure 5.2.1 are less pronounced and difficult to identify while in Figure 5.2.1 we present the EAG in the same interval and both parabolic are easily distinct. This phenomena also occurs with EAG, some dissolves are not easily detected by EAG but they could be found by the variance. Thus, based on this criteria we adopted both features as a possible dissolve indicator.

In this subsection, we describe the second feature used for finding candidate regions. The candidate regions are extracted using the first and second derivatives of the effective average



[Luminance variance curve.]

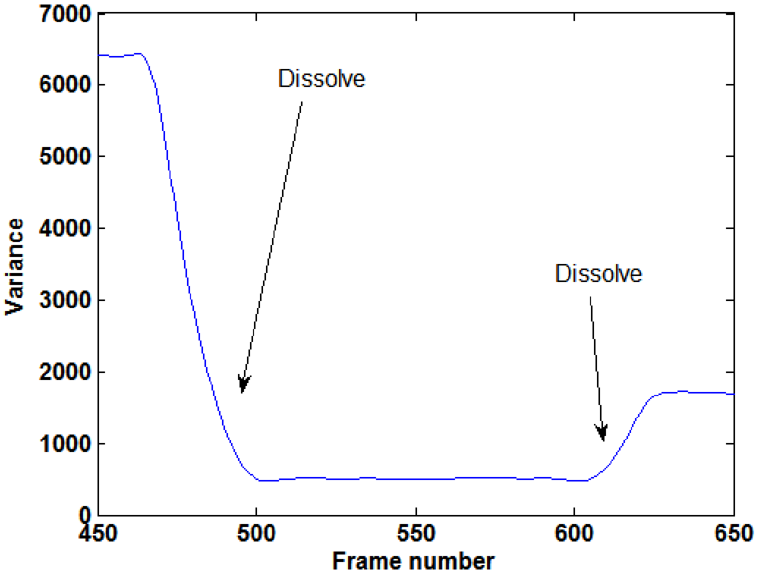
[EAG curve.]

Figure 5.2: Downward parabolic shape described by a dissolve.

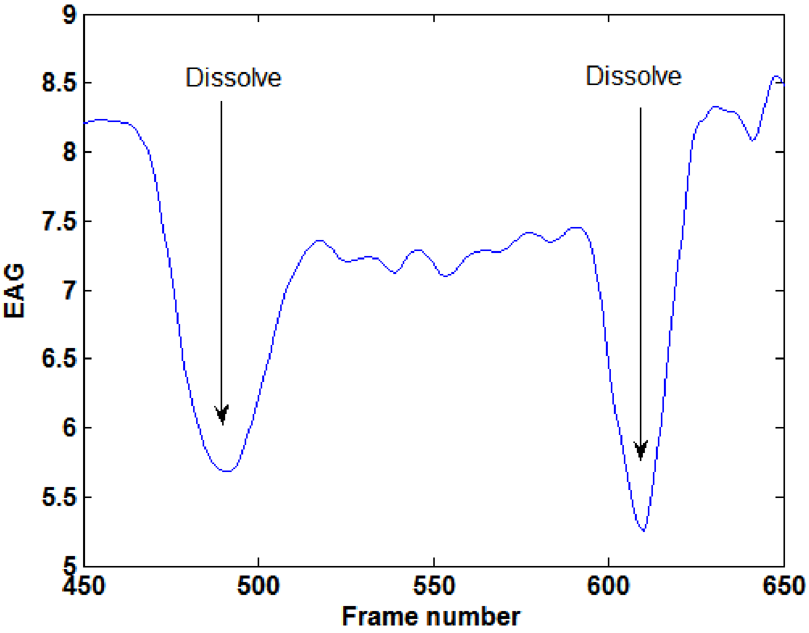
gradient curve. All processes are executed over luminance and edges, thus, when we talk about variance, mean or standard deviation we are talking about luminance of the frame.

5.2.2 Dissolve Regions Detection

The candidate region is identified using the characteristics of first and second derivative of the luminance variance curve. The same process followed in luminance variance curve will be applied to EAG curve. Figure 5.4 shows the procedure used for detecting a candidate region using the luminance variance.



[Variance curve.]



[EAG curve.]

Figure 5.3: Downward parabolas described by two dissolves.

In Figures 5.2.2 and 5.2.2, we present a sequence of luminance variance where we can find a dissolve (the pronounced parabola) and the zoomed version of this dissolve, respectively. The candidate region extraction begins by identifying the search region in the first derivative of the variance/EAG curve, see Figure 5.4(c). To determinate the search region in the first derivative of the variance/EAG curve, the zero crossing point from negative to positive is first identified and used as the center of the search region. Then, the starting point of the search region is determined as the first position to the left of the zero crossing point where the value of the first derivative is zero. The end point is determined as the first position to the right of the zero crossing point where the first derivative is zero. The area between the first point and the zero crossing point is referred as the left size of the search region, and the region between the zero crossing point and the end point is called as right size of the region (see Figure 5.4(c) which shows the search region).

Then, a candidate region is extracted from the search region using the second derivative. We search in the left side of the search region for the minimum local value position of the second derivative. This position is set as the starting point of the candidate region. We do the same process in the right size of the search region and look for the minimum local value position of the second derivative, this position is set as the ending point of the candidate region. Figure 5.4(d) shows the candidate region.

Candidate regions identification are based on the analysis of characteristics of first and second derivative of variance/EAG curve, i.e., searching a downward parabola. Other edition effects also produce the same behavior. This means that the number of interval detected are big, we use the dissolve modeling error to eliminate most of these effects, specially the object/camera motion.

5.2.3 Verification of Candidate Regions

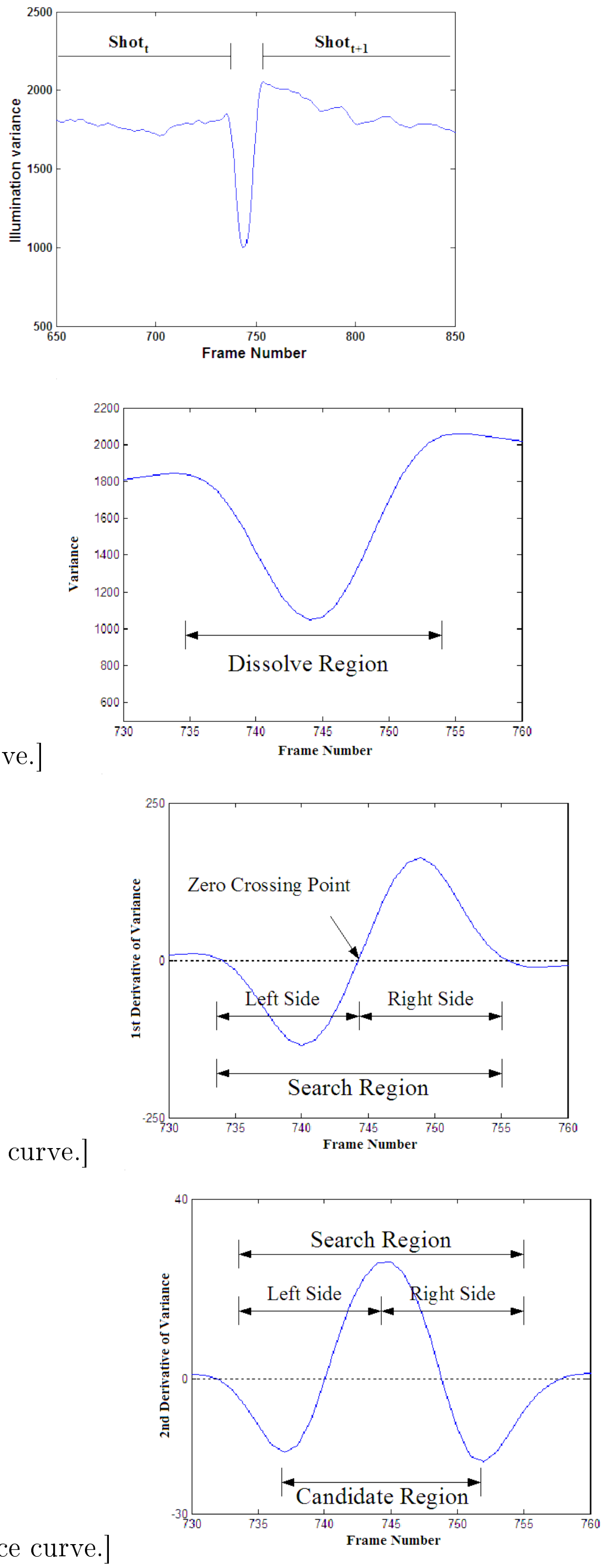
Early researches in dissolve detection based their methods on the characteristics of an ideal model without any correlation between neighbor scenes, i.e., they are based on the assumption that neighboring scenes are independent. However, in most of real cases exists a certain correlation between different scenes that affects the accuracy of the dissolve detection methods. Consequently, dissolve can be missed in a video sequence with high correlation or low luminance variance between adjacent scenes, moreover scene including object/camera motion can be falsely detected as a dissolve. Won et al. [2003] demonstrate the effect of correlation between neighbor scenes. This correlation must be taken into account for the precise detection of a dissolve.

The dissolve modeling error Won et al. [2003] is the difference between an ideal dissolve that starts at t_1 and ends at t_2, and the actual variance curve. Let $\sigma^2_{real}(t)$ be the actual variance curve including a correlation and $\sigma^2_{ideal}(t)$ be the ideal dissolve model curve without any correlation in the region $[t_1, t_2]$. The actual variance curve can be expressed by Eq. (5.8) and the ideal dissolve model by Eq. (5.9). As consequent, the dissolve modeling error can be given by

$$\begin{aligned} f(t) &= 2\alpha(t)\beta(t)E[(P(x,y) - \overline{P(x,y)})(Q(x,y) - \overline{Q(x,y)}) \qquad (5.18) \\ &= 2\alpha(t)\beta(t)\sigma_{PQ} \qquad (5.19) \end{aligned}$$

where σ_{PQ} is the covariance between scene P and scene Q. The covariance can be normalized by the standard deviations at t_1 and t_2:

$$\rho_{PQ} = \frac{\sigma_{PQ}}{\sigma_P \sigma_Q}. \qquad (5.20)$$



[Variance curve.]

[Zoom of variance curve.]

[First derivative of variance curve.]

[Second derivative of variance curve.]

Figure 5.4: Processes for detecting a possible dissolve.

where ρ_{PQ} is the covariance normalized by $\sigma_P\sigma_Q$, i.e., the correlation at t_1 and t_2. If Eq. (5.18) is substituted by Eq. (5.20), the dissolve modeling can be expressed as Won et al. [2003]:

$$f(t) = 2\alpha(t)\beta(t)\sigma_P\sigma_Q\rho_{PQ}. \tag{5.21}$$

At the center of a dissolve, $\alpha(t) = 0.5$, the dissolve modeling error is proportional to the correlation. The maximum dissolve modeling error D_{max} can be defined as Won et al. [2003]:

$$D_{max} = \frac{\sigma_P\sigma_Q\rho_{PQ}}{2}. \tag{5.22}$$

If a correlation c is defined in the region $[t_1, t_2]$, the maximum dissolve modeling error D_{max_c} becomes

$$D_{max\_c} = \frac{\sigma_P\sigma_Q c}{2}. \tag{5.23}$$

A dissolve is detected if the maximum dissolve modeling error D_{max} is less than D_{max_c}, this region can be identify as a dissolve with a correlation smaller than c. Hence, the maximum dissolve error D_{max_c} with correlation c becomes an adaptive threshold determined by the characteristics of each region, where c is the target correlation.

Figure 5.5 shows a flow chart for verifying the dissolve region. For each candidate region, the maximum dissolve modeling error D_{max_c} (c.f. Eq. 5.23)between a dissolve model with a given target correlation c and an ideal dissolve model with no correlation is estimated with variances at the start and end points of each candidate region and the given target correlation c. Won et al. [2003] propose a value of c between 0.15 and 0.45. Then D_{max} becomes the adaptive threshold to verify each candidate region as a dissolve.

The maximum dissolve modeling error D_{max} in each candidate is defined by the difference between the variance σ^2_{center} at the center of each candidate region and the variance $\widetilde{\sigma}^2_{center}$ at the center of an ideal dissolve model estimated by Eq. 5.10. If the maximum dissolve modeling error D_{max} in the current region is less than the target modeling error D_{max_c}, the region is determined to be a dissolve region.

5.2.4 Dissolve Features

After the first filtering of possible dissolves, there still persist some edition effects that cannot be detected by the dissolve modeling error. Most of these are produced by fast motion, continues changes of the frame's content like the motion of water, smoke, fire, etc. Sometimes there exist a dissolve only on a portion of the frame and in that case the region is considered as a false dissolve. Therefore, due to many factors that influence the quality of the detection, it is necessary a second filtering using other features extracted from the interval. Next, we present the features used for a final dissolve filtering.

5.2.4.1 Double Chromatic Difference

Another confirmation test used to distinguish between true dissolves and false alarms caused by object and camera motion is the double chromatic difference (DCD) test proposed by Yu

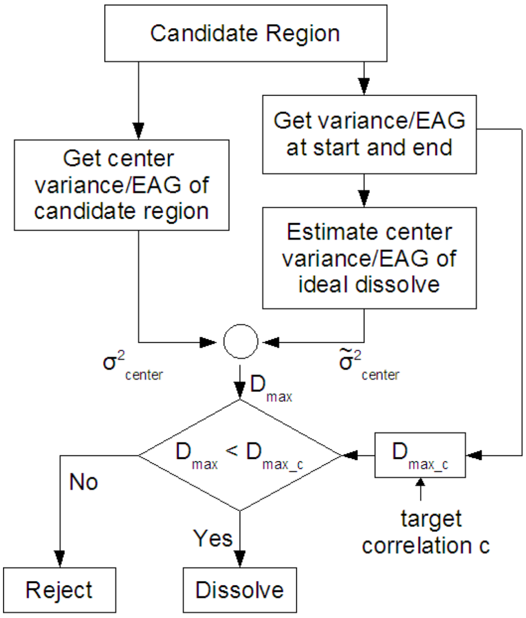

Candidate Region
Get variance/EAG at start and end
Get center variance/EAG of candidate region
Estimate center variance/EAG of ideal dissolve
σ^2_{center}
$\tilde{\sigma}^2_{center}$
D_{max}
$D_{max} < D_{max\_c}$
D_{max_c}
No
Yes
target correlation c
Reject
Dissolve

Figure 5.5: Flow chart for verifying dissolve region Won et al. [2003].

et al. [1997]. The DCD confirmation test defines a synthetic dissolve per potential-dissolve segment, beginning and ending at the first and last frame of the segment, respectively. From these starting and ending frames, the center frame of the synthetic dissolve is formed and compared to the real dissolve shape. If the shape of the comparison error over time is bowl shaped, the potential-dissolve segment is accepted, see Figure 5.7(a).

We refine the dissolve detection obtained with *dissolve modeling error* using a modification of the DCD test. The feature can discriminate dissolve from zoom, pan and wipe. The DCD of frame f_t of a moving image sequence is thus defined as the accumulation of pixel-wised comparison between this average and the intensity of frame $f(x, y, t)$, where $f(x, y, t)$ is a frame in the possible segment of dissolve. This follows the results using Eq. (5.4) and assumptions (5.5) and (5.6).

$$\begin{aligned} DCD(t) &= \sum_{x,y} \left| \frac{f(x,y,t_1) + f(x,y,t_2)}{2} - f(x,y,t) \right| \qquad (5.24) \\ &= \sum_{x,y} \left| \frac{\alpha(t_1) + \alpha(t_2)}{2} \right| |P(x,y) - Q(x,y)| \qquad (5.25) \\ &= \left| \frac{\alpha(t_1) + \alpha(t_2)}{2} - \alpha(t) \right| \sum_{x,y} |P(x,y) - Q(x,y)| \end{aligned}$$

where $t_1 \leq t \leq t_2$, t_1 and t_2 define the starting point and ending frames of a dissolve period. Because $\alpha(t)$ is a decreasing function, $DCD(t)$ is approximately a parabola. Under the assumption of $\alpha(t) + \beta(t) = 1$, there always exists a frame t_m, $t_1 < t_m < t_2$, where

$$f(x,y,t_m) = \frac{f(x,y,t_1) + f(x,y,t_2)}{2} \qquad (5.26)$$

i.e., $DCD(t_m) = 0$.

From Eq. (5.25), it is possible to see that for any t_1, t_2 satisfying $0 \leq t_0 < t_n \leq T$, $DCD(t)$ will always show approximate parabolic shape. That is, the positions of starting point and ending point of dissolve are not essential in DCD calculation. Actually, it is difficult to find out starting point and ending point of dissolve accurately.

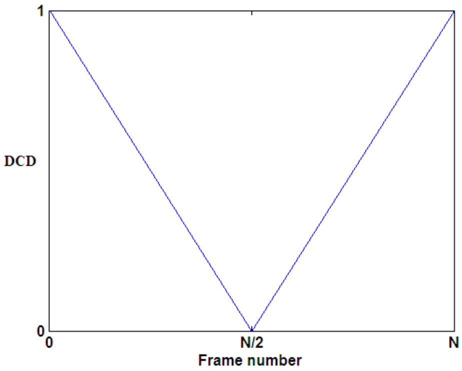


Figure 5.6: Ideal DCD of an ideal dissolve.

Figure 5.6 shows the plot of an ideal DCD of an ideal dissolve transition. Here, we assume an ideal dissolve transition is a dissolve with neither camera motion nor object motion present during any part of the dissolve transition.

We propose to use a one dimensional descriptor that preserves illumination and spatial information instead of the frame content (2 dimensions) in Eq. (5.24). A descriptor that has these characteristics is the projection histogram Trier et al. [1996]. This descriptor also has a successful performance in abrupt transition detection Cámara-Chávez et al. [2006a]. Based in the characteristics of this descriptor and good performance in shot boundary detection, we decide to use it instead of the frame content. Thus, we reduce the computational complexity, from a 2D descriptor to a 1D descriptor, preserving the performance of the DCD. For our modified DCD, the formulation Eq. (5.24) remains the same if $f(x, y, t)$ represents projection histogram. Figure 5.7 shows a comparison between the shape generated by DCD and the modified DCD.

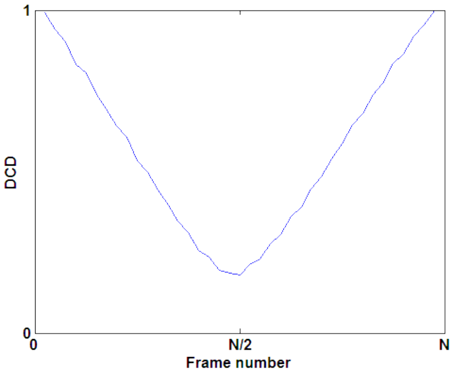
The modified double chromatic difference (SD) of frame f_t of a moving image sequence is thus defined by the comparison between average projection histograms:

$$SD(t) = \sum_x \left| \frac{M_v(x, t_0) + M_v(x, t_N)}{2} - M_v(x, t) \right| \tag{5.27}$$

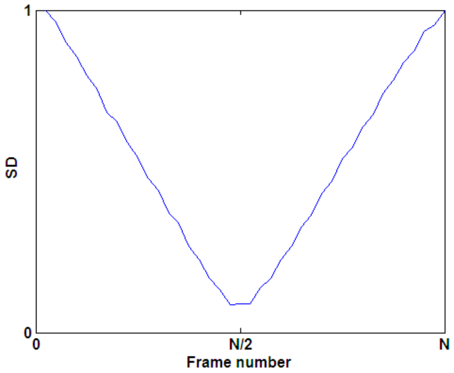
where $t_0 \leq t \leq t_N$, t_0 and t_N define the starting point and ending frames of a dissolve period.

Ideally, there exists a projection histogram $M_v(x, t)$, where

$$M_v(x, t) = \frac{M_v(x, t_0) + M_v(x, t_N)}{2} \tag{5.28}$$



[DCD curve.]



[Modified DCD curve.]

Figure 5.7: Comparison between shapes generated by DCD and modified DCD.

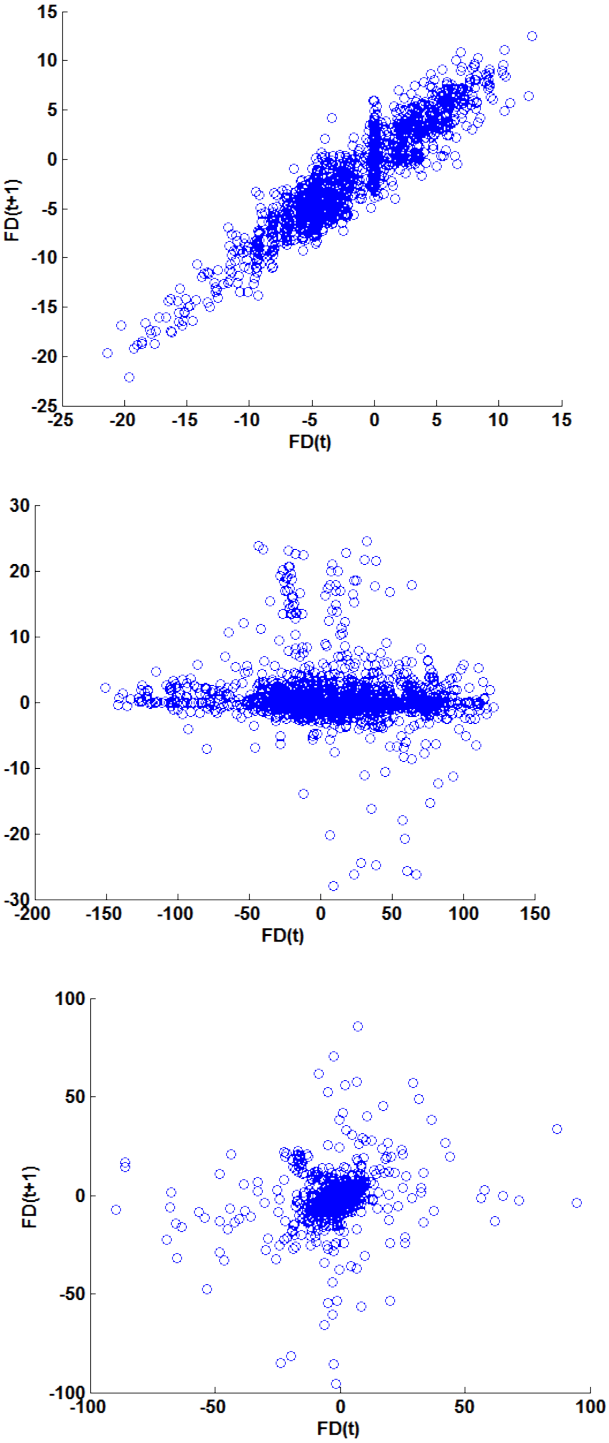
5.2.4.2 Correlation coefficients

The first-order frame differences $FD(t) = f_t - f_{t+1}$ remain constant during the transition of a dissolve. Consider a 2D scatter space spanned by two subsequent frame differences, $X = FD(t)$, and $Y = FD(t+1)$. The points (X, Y) tend to scatter linearly during a dissolve transitions in spite of noise, as shown in Figure 5.2.4.2. Observe the non-linearity in Figure 5.2.4.2 and 5.2.4.2 corresponding to the frames with cut and large motion. Therefore, the correlation coefficient as a measure of linearity between X and Y can be used to distinguish dissolve frames from others.

Han and Kweon [2003] propose a method based on the correlation of the difference sequence. First, the image is divided into blocks of 8×8 $B_t(j)$, where $1 \leq j \leq J$ and J is the number of blocks in frame f_t. Each block is represented by its own average intensity. Then, the blocks of interest (BOI) are selected or inliers among the blocks for frame f_t, according to the following criterion:

$$BOI_t(j) = B_t(j) \qquad \text{if} \quad v_j < V_t / \log V_t, \tag{5.29}$$

where v_j is the variance of block j and V_t the global variance of frame f_t. The role of denominator $\log V_t$ is to normalize the order of V_t. The objective of using BOI instead of B is the reduction of motion artifacts (rejecting outliers) in dissolves. Finally, the BOI differences $BD_t(j)$ between frame f_t and f_{t+1} are used to compute the correlation coefficients ρ_t given



[Dissolve.]

[Cut.]

[Large motion.]

Figure 5.8: Scatter plot of $X = FD(t)$ and $Y = FD(t+1)$.

by

$$\rho_t = \frac{\sigma_{BD_t, BD_{t+1}}}{\sigma_{BD_t}\sigma_{BD_{t+1}}}, \tag{5.30}$$

where σ_{BD_t} is the standard deviation of BD_t and $\sigma_{BD_t,BD_{t+1}}$ is the covariance of BD_t and BD_{t+1}.

5.2.4.3 Other Visual Features

Here we use features presented in the previous section, extract information from that features, specifically for DCD and modified DCD, and some features computed for AT detection. In Figures 5.2.4.3 and 5.2.4.3, we present the luminance variance sequence where a dissolve occurs in the interval $[735, 755]$, and the zoom version of the dissolve. In both figures we can see the position where the dissolve begins (t_1) and the position where it ends (t_2). The computation of some of the features we used here are calculated in these positions. Another important position is the center of the dissolve region, when we said center of region, we are talking about the position, along the luminance variance curve, with the lowest value in the interval (candidate region). That is, in the interval $[t_1, t_2]$ the position with the minimum variance value is searched. This position is defined by C. More formally, we define C as follows:

$$C = argmin_t Var(t), t \in [t_1, t_2] \tag{5.31}$$

where Var is the luminance variance curve.

1. *Ddata*: different information extracted from the dissolve region, the features used are:
   a) 2 correlation values : the correlation between frames in t_1 and C, the other between frames in C and t_2;
   b) 2 color histogram differences : color histogram difference, here we use the $L1$ norm, between histograms of frames at t_1 and C, the other histogram difference is computed between histograms of frames at C and t_2;
   c) correlation by blocks of interest in the sequence: this feature is computed only on the target intervals and use the dissolve descriptor Han and Kweon [2003] (cf. Section 5.2.4.2). The median of the correlation coefficients is calculated, i.e., $median(\rho_t)$, $t \in [t_1, t_2]$.
2. *DCD* features: we extract information from DCD curves
   a) the quadratic coefficient of the parabola approximating the DCD curve at best Lienhart [2001b].
   b) The "depth" of the DCD curve (downward parabola). In Figure 5.2.4.3, we present the DCD feature generated from frames within the interval $[t_1,\ t_2]$. From the DCD curve, we find the "depth" of the parabola as the height difference between $DCD(m)$ and $DCD(0)$ (or $DCD(N)$) Hanjalic [2002].

$$\psi(i) = \begin{cases} 1 - \frac{min(DCD_i(m), DCD_i(N))}{max(DCD_i(m), DCD_i(N))}, \text{ if } R \leq 0 \\ 1 - \frac{min(DCD_i(m), DCD_i(0))}{max(DCD_i(m), DCD_i(0))}, \text{ if } R > 0 \end{cases} \tag{5.32}$$

where $R = |DCD_i(m) - DCD_i(N)| - |DCD_i(m) - DCD_i(0)|$ and m is the position with the lowest value in the DCD, N is the size of the DCD cruve and i is the interval (region) number.

3. *SD* features: the modified DCD, here we extract the same features presented in the DCD features (previous item),

4. *VarProj*: difference of the projection histograms extracted in the first step (cut detection), i.e., the dissimilarity of consecutive projection histograms during the dissolve interval (from t_1 to t_2). This difference is normalized in size with the objective that all the projection intervals have the same size.

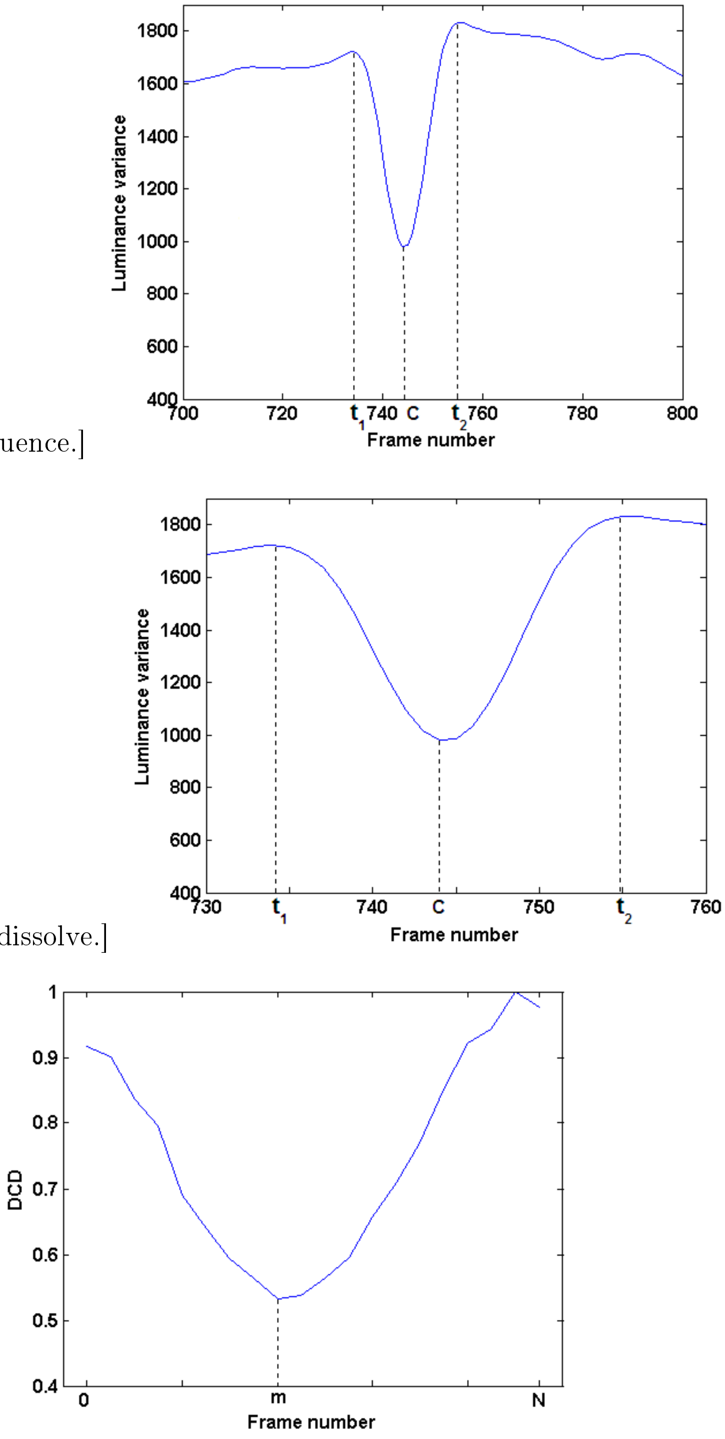
5. *Motion*: motion vectors are also extracted in the first step, when the phase correlation method is computed, for each block we compute the magnitude of the motion vector.

We concatenate them in one feature vector given as input to our kernel-based SVM classifier in order to determine "dissolves" and "non-dissolves" video segment.

5.2.5 Machine learning approach

The classification problem can be restricted to a two-class problem Cord et al. [2007]. The goal is, then, to separate the two classes with a function induced from available examples. We hope to produce, hence, a classifier that will properly work on unknown examples, i.e. which generalizes efficiently the classes defined from the examples. Therefore we consider dissolve as a categorization task and classifying every possible dissolve interval into "dissolve" and "non dissolve".

Figure 5.10 shows the steps of our approach. The first step is the detection of possible dissolves, this step is based on three processes. The first process consists in the computation of luminance variance and the EAG curve. Then in the second process, for each type of curves we find the candidate regions using the first and second derivatives of the luminance variance and EAG curves, respectively. The third process is the first filtering of possible dissolves in our approach which consists in eliminating the false dissolves generated by object/camera motion. For this purpose we use the dissolve modeling error. When we presented the luminance variance and EAG curves (cf. Section 5.2.1), we said that some dissolves that not appear clearly define in luminance variance curve may appear clearly in EAG curve and vice versa. With the previous three process we have a list of possible dissolve regions computed from luminance variance curve and we also have other list computed from EAG curve. We merge both lists in order to have a single list of regions from we are going to extract the features that we will use for a final classification. The last stage of our approach consists in extracting the features from these interval. We compute the DCD and modified DCD features (Section 5.2.4.1), the correlation coefficients (Section 5.2.4.2) and the other visual features (Section 5.2.4.3). The concatenation of all these features correspond the input to our SVM classifier. Finally, these intervals are been classified into "dissolves" and "non dissolves".



[Luminance variance sequence.]

[Zoom version of possible dissolve.]

[DCD curve.]

Figure 5.9: Dissolve features.

We use the same kernels functions presented in Section 4.4: linear, polynomial, Gaussian-$L2$, Gaussian-χ^2 and triangular. For further details on SVM and kernel functions see Appendix A.

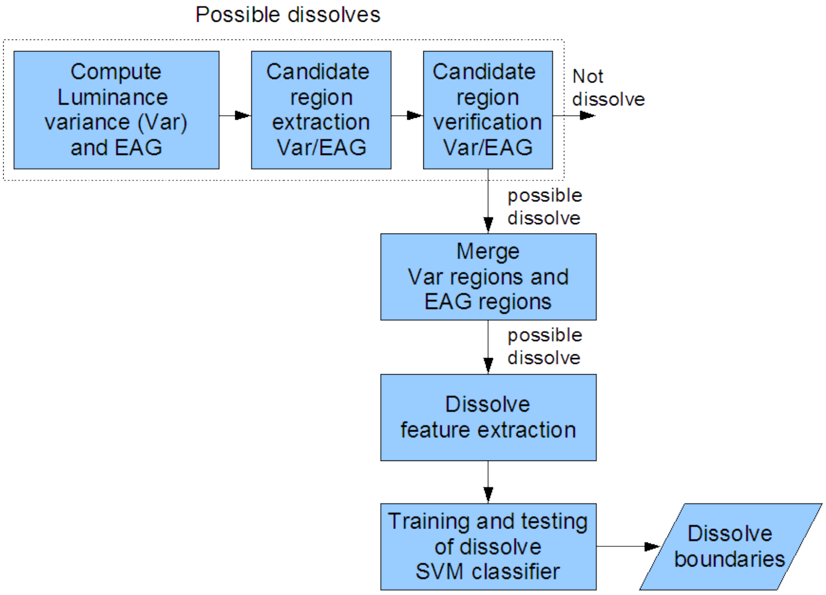

Possible dissolves
Compute Luminance variance (Var) and EAG
Candidate region extraction Var/EAG
Candidate region verification Var/EAG
Not dissolve
possible dissolve
Merge Var regions and EAG regions
possible dissolve
Dissolve feature extraction
Training and testing of dissolve SVM classifier
Dissolve boundaries

Figure 5.10: Proposed model for dissolve detection.

5.3 Fade detection

A fade process is a special case of dissolve process. During a fade, a video sequence gradually darkens and is replaced by another image which either fades in or begins abruptly. Alattar [1993] detects fades by recording all negative spikes in the second derivative of frame luminance variance curve. The drawback with this approach is that motion also would cause such spikes. Lienhart [1999] proposes detecting fades by fitting a regression line on the frame standard deviation curve. Truong et al. [2000a] observe the mean difference curve, examining *the constancy of its sign* within a potential fade region. We present further extensions to these techniques.

A fade-out process is characterized by a progressive darkening of a shot P until the last frame becomes completely black. A fade-in occurs when the picture gradually appears from a black screen. The fades can be used to separate different TV program elements such as the main show material from commercial blocks.

Fade-in and fade-out occur together as a fade group, i.e., a fade group starts with a shot fading out to a color $\mathcal{C}$ which is then followed by a sequence of monochrome frames of the same color, and it ends with a shot fading in from color $\mathcal{C}$.

As a fade is a special case of a dissolve we can explore some of the features used for dissolve detection. The salient features of our fade detection algorithm are the following:

1. The existence of monochrome frames is a very good clue for detecting all potential fades, these are used in our algorithm. In a quick fade, the monochrome sequence may be compound by a single frame while in a slower fade it would last up to 100 frames Truong et al. [2000a]. Therefore, detecting monochrome frames (candidate region) is the first step in our algorithm.

2. In this second step we are going to use a descriptor that characterizes a dissolve, our improved double chromatic difference. The variance curves of fade-out and fade-in frame sequences have a half-parabolic shape independent of $\mathcal{C}$. Therefore, if we compute the modified DCD feature in the region where the fade-out occurs we will have a parabola shape, the same principle is applied for the fade-in. Figure 5.11 shows the half-parabolic formed in the fade-in and fade-out regions. Therefore, if we compute the modified DCD feature in the region where the fade-out occurs we will have a parabola shape, the same principle is applied for the fade-in. In Figures 5.12 and 5.13, we have the parabolas generated using the modified DCD feature in the fade-out and fade-in regions, respectively.

3. We also constrain the variance of the starting frame of a fade-out and the ending of a fade-in to be above a threshold to eliminate false positives caused by dark scenes preventing them from being considered as monochrome frames.

Some of the techniques used for detecting fades are not tolerant to fast motion, which produces the same effect of a fade. DCD feature is more tolerant to motion and other edition effects or combinations of them. Our modified double chromatic difference feature preserves all the characteristics of the feature presented in Yu et al. [1997], with the advantage that we reduce the size complexity of the feature, from 2D to 1D.

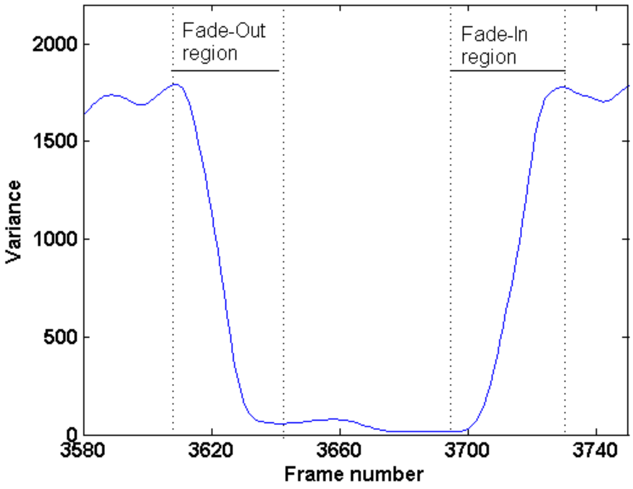


Figure 5.11: Variance curve in the fade-out and fade-in interval.

5.4 Experiments

In this section we present the experiments conducted in order to choose the better parameters for our system and also compare our method with other methods in TRECVID evaluation.

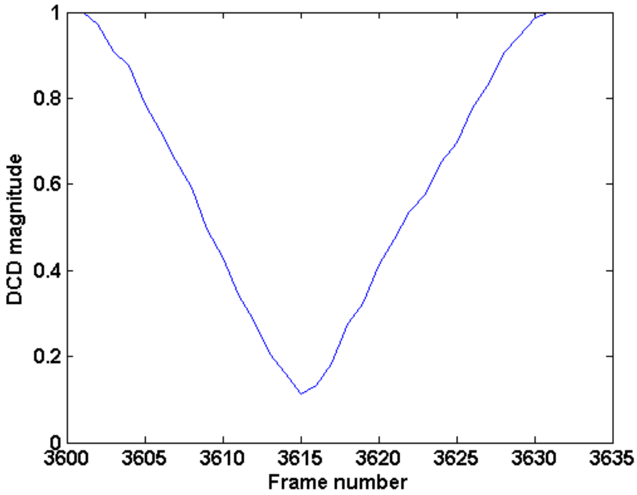


Figure 5.12: Modified DCD generated in the fade-out region.

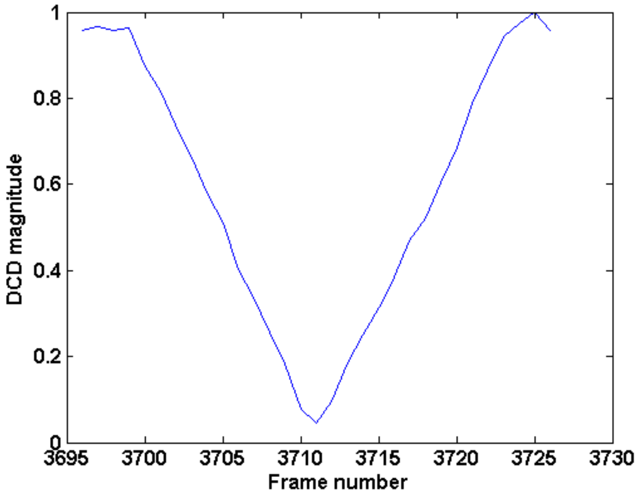


Figure 5.13: Modified DCD generated in the fade-in region.

5.4.1 Data set

We test our system with two different data sets. For the first experiment, the training set consists of a single video of 20,000 frames with 20 dissolves. This video is captured from a Brazilian TV-station and is composed by a segment of a soccer match. The training video was labeled manually by ourselves. The test set consists of TRECVID-2002 data set. For the second experiment, the training set consists of TRECVID-2002 data and the test set consists of TRECVID-2006 data. The TRECVID data sets are described later in Sections 5.4.3 and 5.4.4, respectively.

5.4.2 Features

As our objective is to avoid pre-processing and post-processing steps we combine distinctive features. Next we are going to present the features used for our fade out-in detector and dissolve detector.

For fade detection we choose a threshold of 200 for the variance of each frame, if the variance is lower than that value we consider it as a monochrome frame and a possible fade. After that is necessary to see if the interval has two downward parabolas, one for fade-in and other for fade-out.

For dissolve detection, variance and edge average gradient curves are smoothed by B-spline smooth method in order to reduce the noise influence. After computing all possible dissolve intervals from first and second derivative of both curves, we make the first filter process eliminating intervals through verification candidate region method. Won et al. [2003] propose a value of c between 0.15 and 0.45 (cf. Section 5.2.3). In our case we use a value of 0.8 because our intention is that the classifier make the decision based on the characteristics of the candidate region.

From candidate region, we compute a set of features that describe the characteristics of the interval (cf. Section 5.2.4.3):

DCD	compute double chromatic difference for each interval, and quadratic coefficient and parabola depth is computed
SD	compute modified double chromatic difference for each interval, and quadratic coefficient and parabola depth is computed
$VarProj$	vertical and horizontal projection differences are used
$Motion$	magnitude of the motion vectors

The values of $Ddata$ are computed from the candidate region, features are extracted between the beginning of the region and the "center" of the downward parabola formed in luminance variance curve and between the "center" and the ending of the candidate region.

$Ddata$	identify the frame comparison positions on luminance variance curve, i.e., the begging (t_1), the "center" (C) and the ending (t_2) of the candidate region, where $C = argmin_t Var(t)$, $t \in [t_1, t_2]$. Features computed are color histogram difference, correlation between frames and the median of correlation coefficients computed using the correlation by blocks of interest inside the interval (cf. Section 5.2.4.2).

5.4.3 TRECVID 2002

The training set consists of a single video of Brazilian soccer match, which contains 20,000 frames with 20 dissolves. We use a SVM classifier and train it with different kernels: linear, polynomial, Gaussian with χ^2, Gaussian with L2 norm and triangular.

We use the corpus of TRECVID-2002 Video Data Set. The shot boundary test collection contains 4 hours and 51 minutes of video. The videos are mostly of documentary/educational nature, but very varied in age, production style, and quality. The total size of data is 2.88 Gigabytes of MPEG-1 video. The collection used for evaluation of shot boundary contains 624 gradual transitions with the following breakdown:

- 511 dissolves
- 63 fades out-in
- other

We use the following features for dissolve detection: $Ddata$ and SD. We conduct numerous experiments that provide interesting and meaningful contrast. Table.5.1 shows the mean recall/precision measure and the variance for each kernel function. The five kernels present similar results, thus the quality of the features selected are good. Seeing the variance results, it is also possible to conclude that the classification's results of all the videos are more or less the same.

Kernel	Recall	Var. Recall	Precision	Var. Precision	F1
Linear	0.819	± 0.032	0.886	± 0.011	0.832
Polynomial 3	0.746	± 0.044	**0.908**	± 0.006	0.800
Gauss-$L2$	0.837	± 0.026	0.901	± 0.010	0.851
Gauss-χ^2	**0.850**	± 0.025	0.905	± 0.009	**0.877**
Triangle	0.821	± 0.032	0.901	± 0.010	0.840

Table 5.1: Performance measure in mean precision and mean recall for each kernel function.

We want to emphasize with these results that our system is very robust to training data set. Indeed, the training data set used here is Brazilian TV videos which are very different in terms of quality, format and length from TRECVID videos we used for testing our system.

In this second experiment, we use 11 videos from TRECVID 2002 corpus. We take one of these videos for training and testing our system on the 10 others. We repeat this experiment to explore all the possible combinations and present the results in Table.5.2. As it occurs with ATs the best kernel function is the Gaussian-χ^2, then followed by the Gaussian-$L2$. In ATs the worst kernel function was the linear one, but in GT detection the polynomial kernel is the one the performs worst (here worst does not mean bad results) compare to the other kernels.

We can see through these results the stability of our system. Our system is among the most efficient ones since best methods offer average precision and recall between 0.7 and 0.9.

5.4.4 TRECVID 2006

The test data are composed by news video in Arabic, Chinese and English. The data were collected by Linguistic Data Consortium (LDC) during November and December of 2005,

	Linear		Polynomial		Gaussian-$L2$		Gaussian-χ^2		Triangle	
Video	Re.	Pr.	Re.	Pr.	Re.	Pr.	Re.	Pr.	Re.	Pr.
1	0.65	0.87	0.72	0.84	0.69	0.88	0.72	0.90	0.67	0.89
2	0.66	**0.91**	0.66	**0.92**	0.73	0.92	0.73	0.92	0.65	0.92
3	0.77	0.90	0.23	0.77	0.82	0.87	0.85	0.90	0.79	0.89
4	0.77	0.88	0.37	0.92	0.76	**0.93**	0.80	**0.95**	0.77	**0.93**
5	0.97	0.78	0.91	0.81	**1.0**	0.71	**1.0**	0.71	0.97	0.79
6	**1.00**	0.75	**0.99**	0.73	**1.0**	0.74	**1.0**	0.73	**0.99**	0.76
7	0.85	0.86	0.91	0.83	0.95	0.79	0.95	0.83	0.93	0.86
8	0.94	0.86	0.95	0.87	0.95	0.81	0.96	0.84	0.94	0.84
9	0.94	0.82	0.81	0.79	0.96	0.78	0.97	0.81	0.95	0.83
10	0.98	0.75	0.75	0.76	0.96	0.77	0.98	0.79	0.92	0.81
11	0.69	0.87	0.79	0.87	0.7	0.92	0.74	0.93	0.79	0.91

Table 5.2: Performance measure for each kernel function.

digitized and transcoded to MPEG-1. The test collection comprises about 7.5 hours, including 13 videos for a total size of about 4.64 Gb. It comprised 13 videos for a total size of about 4.64 Gb. The reference data was created by a student at NIST whose task was to identify all transitions. The distribution of GTs is as follows

- 1509 dissolve (39.9%)
- 51 fades out-in(1.3%)
- 381 other (10.1%)

The training data used was the TRECVID 2002 data set, see Section 5.4.3 for more details. We use Gaussian-χ^2 kernel function for our SVM classifier, the selection of this kernel is based on the excellent performance on ATs detection.

In Table 5.3, we present the visual feature vectors for dissolve detection used for the 10 runs. The feature vector *Ddata* is computed from DCD features, except for Etis7, Etis8 and Etis9 runs, that is computed from SD features. The objective is to compare the performance of both features (DCD and SD), see if the modification we are proposing works well, i.e., reduce the complexity and preserve the accuracy. Each run is tested with the hold data set (13 videos).

In Table 5.4 we show the performance of our system for gradual transition detection (dissolves and fades), measured in recall and precision. The recall and precision are computed from the GT boundaries, i.e., we fusion the boundaries detected by the dissolve detector and the fade out-in detector. It is important to notice that our framework detects only dissolves and fades, but in the data set there is another class named “other”, which includes other types of transitions. It is possible to find wipe, fade in, fade out (notice that fade in and fade out are not merged as a compound GT), black frames that separate two consecutive shots and other kind of transitions. The experiment run that show the best performance in terms of $F1$ measure is the $Run4$, that also has the best precision and the second best

Run	Features
1	Ddata, VarProj
2	Ddata, Motion
3	Ddata, DCD
4	Ddata, DCD, SD
5	Ddata, DCD, VarProj
6	Ddata, DCD, Motion
7	Ddata, SD
8	Ddata, SD, VarProj
9	Ddata, SD, Motion
10	Ddata, DCD, SD, VarProj, Motion

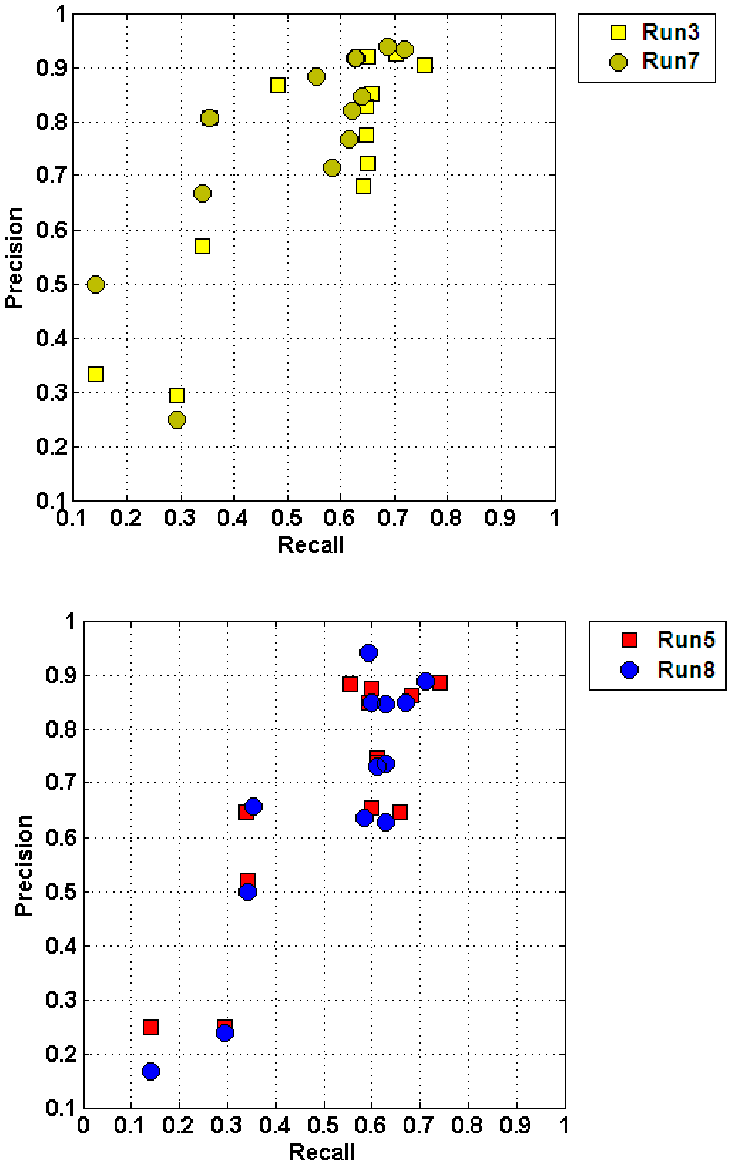
Table 5.3: 10 best combinations of visual features for gradual transition detection.

recall. This experiment run combines the histograms differences, frames correlation, median of correlation coefficients, DCD and modified DCD features. If we compare *Run*3 and *Run*7 the performance in terms of $F1$ measure is very similar: 0.716 and 0.711, respectively. The former use the DCD feature while the latter the modified DCD. We see the same behavior with *Run*5 and *Run*8 with $F1$ equal to 0.678 and 0.672, respectively. In the former we include the DCD feature and in the latter the modified DCD feature. In both comparisons the performance is almost the same. In next figures we are going to present the performance using recall/precision measure for each video for these four runs.

Run	Recall	σ_{recall}	Precision	$\sigma_{\text{prec.}}$	F1
1	0.585	0.031	0.771	0.060	0.665
2	0.602	0.029	0.798	0.061	0.686
3	**0.632**	0.037	0.825	0.044	0.716
4	0.621	0.033	**0.853**	0.042	**0.719**
5	0.607	0.032	0.769	0.049	0.678
6	0.581	0.029	0.807	0.051	0.676
7	0.612	0.032	0.849	0.040	0.711
8	0.604	0.031	0.758	0.058	0.672
9	0.586	0.030	0.837	0.048	0.689
10	0.583	0.031	0.757	0.059	0.659
Mean TRECVID	0.533	-	0.626	-	0.576
Max TRECVID	0.775	-	0.858	-	0.814

Table 5.4: Detailed results for all runs of gradual transition.

In Figure 5.14, we compare the accuracy of the double chromatic difference method (*Run3*, *Run5*) and our modified double chromatic difference method (*Run7*, *Run8*), respectively. The former is represented by square marker and the latter by round marker. We can see that in both figures the results are very similar, i.e., the results produced by the double chromatic difference features are similar than results produced by modified double chromatic difference. We reduce the computational complexity, from a 2D descriptor to a 1D descriptor, preserving the performance of the DCD method. We can see in both figures that there are two videos where our system shows a bad performance. This is due to extremely short dissolves, the transitions take only 2 frames. In one of the videos has only one dissolve bigger than 2 frames and our system detected it, the rest of dissolves take 2 frames of duration. The second video is from a TV journal, it has the same characteristics of the previous video, i.e., many (almost all) extremely short dissolves and edition effects that made our system misclassify these edition effects as dissolves. These effects consist in a portion of the frame disappears slowly, exactly as a dissolve does, and produce the same downward parabola effect in luminance variance and DCD/SD curves. We present an example in Figure 5.15. These effects are very difficult to identify, a probably solution could be separate the frame in blocks and analyze if the effects occur in all or almost all the blocks.



[Run3 and Run7]

[Run5 and Run8]

Figure 5.14: Comparison between double chromatic difference method square marker)and our modified double chromatic difference method (round marker)

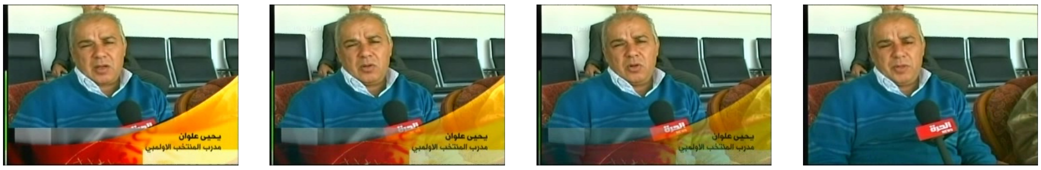


Figure 5.15: An example of a false dissolve.

In Figure 5.16, we show the performance of our system. We can see that we have high values for precision and lower values for recall. For the recall values, let first remember that our system only detects dissolve and fade out-in transitions. But there also exists another class which involves wipes, fade-in, fade-out, black frames between shots and other special effects that separate two consecutive shots. These class represent more or less the 20% of the GTs and also exists some dissolves with fast motion embedded. As the recall count the detected transitions from all possible transitions, our system detects only a percentage of all GTs. Another reason that affect the performance is that some transitions that they really exist are not considered in the ground truth. This omission is due to errors in the labeling process. On the other hand, we still want to compare the performance of our improved feature. In the figure, we represent the runs using the DCD with square marker and the runs using the SD with round marker. In the three cases the outcome are almost the same, thus we can conclude that our feature is as good as the original feature but with less computational complexity.

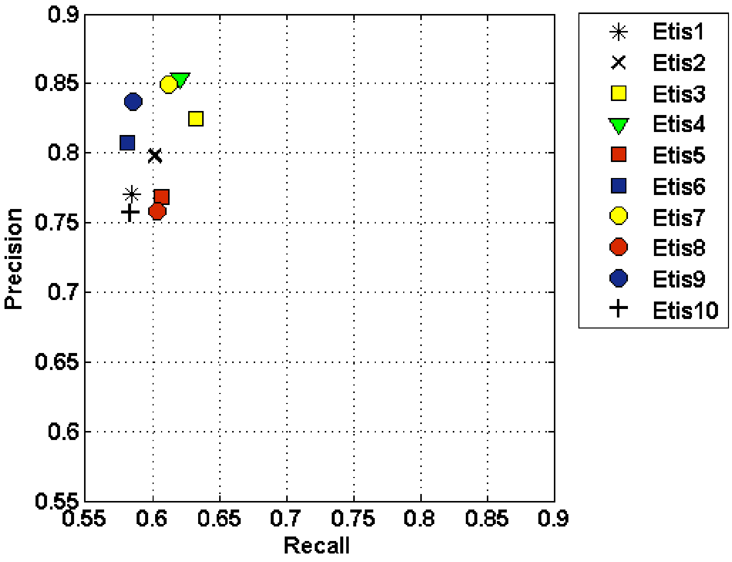


Figure 5.16: Performance measure in recall and precision for each of our runs

Another measure used for gradual transitions is the accuracy of the interval detected, i.e., how well the interval of the gradual transition detected matches with the real transition. Frame-precision and frame-recall are used to measure this:

$$frame{-}recall = \frac{\#overlapping\ frames\ of\ \text{``}detected\text{''}\ transition}{\#frames\ of\ detected\ reference\ transition} \tag{5.33}$$

$$frame{-}precision = \frac{\#overlapping\ frames\ of\ \text{``}detected\text{''}\ transition}{\#frames\ of\ detected\ submitted\ transition} \tag{5.34}$$

Figure 5.17 shows an example of how to compute the frame-recall and frame-precision. These measures are only computed for detected GTs. The reference transition is the interval where a GT occurs, i.e., the true interval (in Figure 5.17 from frame 40 to frame 70), submitted transition is the interval found by the GT detector (in Figure 5.17 from frame 50 to frame 75) and the overlapping frames are the intersection between reference transition and submitted transition (in Figure 5.17 from frame 50 to frame 70). Thus, the $frame{-}recall = 20/30$ and the $frame{-}precision = 20/25$.

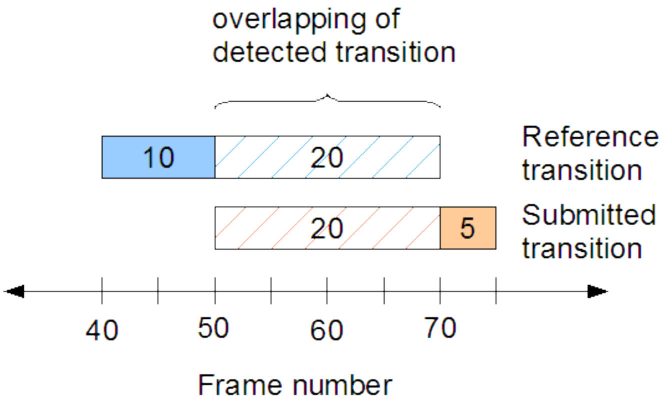


Figure 5.17: Elements for computing frame-recall and frame-precision of GTs.

Note that a system can be very good in detection and have poor accuracy, or it might miss a lot of transitions but still be very accurate on the ones it finds.

Table 5.5 shows the results of all runs measured in frame-recall and frame-precision. The values that measure how well the detected transition fits with the reference transition are more or less the same, independent of the run. The values of frame-recall and frame-precision are very close, this means that the accuracy for all runs is almost the same.

Trecvid 2006 Participants

We made a classification based on the approach used for the participants of TRECVID 2006 Evaluation:

Machine learning approach

Run	F-Recall	σ_{recall}	F-Precision	$\sigma_{\text{prec.}}$	F1
1	0.766	0.009	0.849	0.004	0.805
2	0.773	0.010	0.850	0.004	0.810
3	**0.775**	0.007	0.849	0.004	0.810
4	**0.775**	0.008	0.849	0.004	0.810
5	0.769	0.010	0.847	0.003	0.806
6	0.774	0.009	0.849	0.005	0.810
7	**0.775**	0.010	0.850	0.004	**0.811**
8	0.770	0.009	0.849	0.004	0.808
9	0.772	0.010	**0.851**	0.004	0.810
10	0.767	0.011	0.843	0.004	0.803
Mean Trecvid	0.674	-	0.768	-	0.718
Max Trecvid	0.889	-	0.921	-	0.905

Table 5.5: Detailed results for all runs for frame-precision and frame-recall

- *AT&T*: builds six independent detectors for cuts, fast dissolves (less than 4 frames), fade-in, fade-out, dissolve, and wipes. Each detector is a finite state machine. The dissolve detector is computed from luminance variance and use many features for dissolve verification. For fade-in and fade-out detectors use the intensity histogram variance. The dissimilarities are computed between consecutive frames and frame distance of 6 frames. SVM is applied to dissolve detector.

- *Chinese Academy of Sciences / JDL (CAS/JDL)*: presents two parts: fade out-in detection and other type of gradual transitions. For fade detection they use two features: image monochrome and joint entropy between two frames. For GTs, a sliding window of 60 frames is defined. It uses a two pass approach, first selects the suspicious transition candidates using a low threshold method and then judges the candidates by using the SVM base method. Needs to improve distinction between GTs and camera motion.

- *FX Palo Alto Laboratory (FXPAL)*: uses dissimilarity features within the particular temporal interval as the input for kNN classifier. The features used are global and local histograms. The same features used for AT detection. All possible pairwise comparisons between frames are visualized as a similarity or affinity matrix. Define two matrices, one for global and the other local histograms, with the (i, j) element equal to the similarity between frames i and j. Time, or the frame index, runs along both axes as well as the diagonal. The input is formulated as correlation of specific kernels along the main diagonal of the similarity matrix.

- *Helsinki University of Technology (HelsinkiUT)*: the system is based on a 2D self-organizing map (SOM). There is one classifier for each feature calculated from the frames, and each classifier has a weight value. The final decision is made by comparing the weighted vote result of the classifiers to a threshold value. ATs and GTs are detected using the same method. Computationally the most expensive (because of SOMs).

- *KDDI and R&D Laboratories (KDDI)*: proposes an extension of 2005 approach, a new additional feature (for long dissolve detection) and the combination of multi-kernels improve the accuracy of the detector. This approach works on the uncompressed domain (very fast execution time). The dissolve and fade detection use the frame activity which is the sum of the square difference. This frame activity also performs a downward parabola when a dissolve occurs. In the case of a fade-in or fade-out, activity curve shows monotonous increase/decrease. Then a temporal filtering is executed between the current frame and previous n frames, this feature produce a peak in a presence of a dissolve. For confirm the presence of a dissolve the system use both features (two shapes), the downward parabola and the peak. The system also has a dissolve detector based on edge histogram descriptor specified in MPEG-7 and is extracted from DC images[3].

- *Tsinghua University (Tsinhgua)*: two independent detectors for fade in-out and GTs. Fade in-out detector based on detecting monochrome frames using mean and the standard deviation of the intensities. Then search the fade-out boundary of the previous shots and the fade-in boundary of the next shot. For GT detection, is based on graph partition model. The graph is associated to a weight matrix which indicate the similarity between two nodes, the larger the more similar. The input is formulated as correlation of specific kernels along the main diagonal of the similarity matrix. The system uses different kernels in order to detect different types of transitions. The features used are global color histograms in RGB color space (16 bins per channel) in HSV color space and, local color histograms in RGB color space (2, 4×4 and 8×8 blocks) and HSV color space (2). Finally a SVM classifier is used to detect the transitions.
- *University of Marburg (Marburg)*: the main idea of the GT detection is to view a GT as an abrupt change at a lower temporal resolution and also proposes an unsupervised kmeans clustering for GTs. First, frame dissimilarities are computed based on histograms of approximated DC-frames. These dissimilarities are computed from different frame distances ($d = 6, 10, 20, 30, 40, 50$). The signal is filtered using a sliding window in order to detect isolate peaks. Finally, these features are clustered using a k-means algorithm.
- *Tokyo Institute of Technology (TokyoInstTech)*: for GT detection, a radial kernel function is used for the SVM classifier. The features used are the difference between consecutive frames, the optical flow, the change in the Hue-Saturation color histograms and edge.

Threshold-based approach

- *Indian Institute of Technology at Bombay (IIT.Bombay)*: the system attempt to detect dissolves using a simple method. Dissolves are detected analyzing the change

[3] Reduced images formed from the collection of scaled Discrete Cosine (DC) coefficients in intra-coded discrete cosine transformation compressed video retain “global” feature.

in the brightness value of frames. Within a shot, the total brightness remains predictable when a GT is encountered a cone-like pattern is produced.

- *Artificial Intelligence and Information Analysis (AIIA)*: mutual information is used as similarity measure. The accumulative mutual information shows a “V” pattern, i.e., in the first part of the GT the mutual information decreases while in the second part it increases. A threshold is used to identify the GT and first and second derivative to confirm the presence of GT and also to identify the boundaries. Parameters are set empirically.
- *Chinese University of Hong Kong (CityUHK)*: uses the same features and similarity measures of AT detection. A distinguishing characteristic between cut, long GTs, and false positives is the smoothness of their Earth Mover’s Distance values across time. Gradients from Earth Mover’s Distance are calculated and analyzed to determinate if as GT occurs.
- *Communication Langagière et Interaction Personne-Système (CLIPS)*: dissolves are the only GT effects detected by this system. The method is very simple: a dissolve is detected if the $L1$ norm of the first image derivative is larger enough compared to the $L1$ norm of the second derivative of the second image derivative, this checks that the pixel intensities roughly follows a linear but non constant function of the frame number. The method detects dissolves between constant or slowly moving shots. A sliding window of 11 frames is used and a filter is then applied. Parameters are set manually.
- *European Cooperation in the Field of Scientific and Technical Research (COST292)*: uses a general system for ATs and GTs. This system is described in the previous chapter.
- *Dokuz Eylol University (Dokuz)*: uses the same features of AT detection. GTs are detected on a second pass by computing the length of the consecutive cuts. It uses a threshold that holds the minimum number of frames that a shot holds. The minimum number is fixed to 10 frames.
- *Institute of Informatics and Telecommunications National Center for Scientific Research “Demokritos” (ITT/NCSR Demokritos)*: the method relies on spatial segmentation and a similarity measure based on Earth Mover’s Distance (EMD). The GT detection is based on the fit of a frame to a spatiotemporal template. The system uses the features described in the previous chapter.
- *RMIT University (RMIT)*: the approach is also based on the moving query window, but frames are note ranked (as it is done for AT detection). For each frame within the window, a similarity is computed for the current frame. Frames on either side of the current frame are then combined into two sets of pre-frames and post-frames. The ratio between average similarities of each set is used to determinate a GT.

- *University of Modena (Modena)*: the same model used for ATs. They work on determining the range (in frames) and nature of a GT and integrating AT and GT detectors. A window of 60 frames is used.

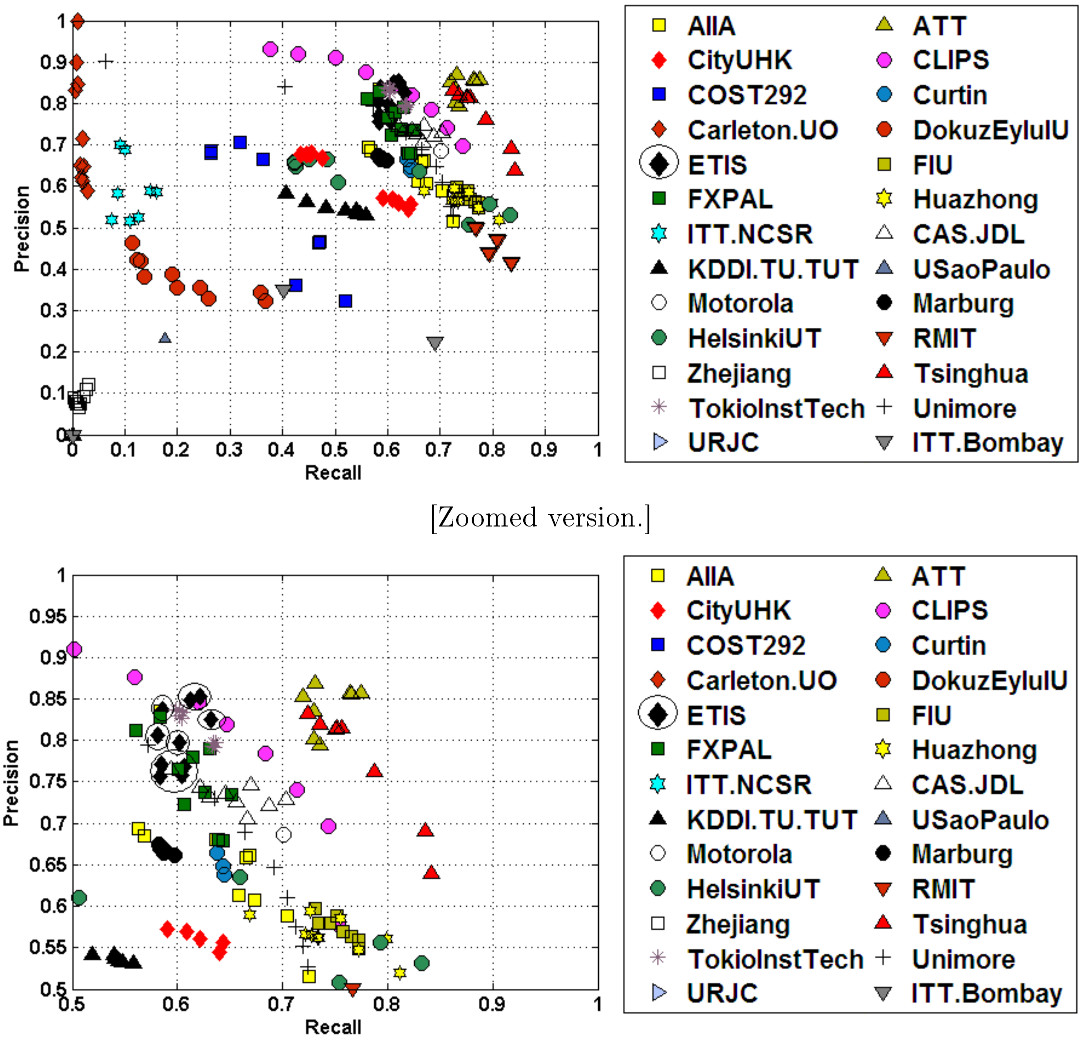
There is no information available for *Curtin University*, *Florida University (FIU)*, *Huazhong University of Science and Technology*, *Motorola* and *Zhejiang University* systems. Carleton University, University of São Paulo and University Rey Juan Carlos systems only detect ATs.

Figure 5.4.4 shows the performance of our system for gradual transition, measured in recall and precision, and Figure 5.4.4 is a zoomed version. We compare our results with all other submissions. The best two submissions are from AT&T and Tsinghua systems, both of them are SVM-based methods. In the case of AT&T, it has six independent detectors for cuts, fast dissolves (less than 4 frames), fade-in, fade-out, dissolve, and wipes. Thus, they can detect more types of transitions. We can see that the precision of our system is similar to the precision of AT&T. Tsinghua system has a similar structure than our system, i.e. a SVM-based detector for ATs and GTs, and a detector for fade out-in transitions. The difference with our system is that the features are constructed from a graph partition model and also the system has a post-processing module that detects short transitions (less than 4 frames). Therefore, they can detect the short dissolves that we missed. This is the reason why they have a higher recall, but if we consider the precision it is more or less the same of us. The other three teams that have almost the same performance of us, two of them are machine learning-based (TokioInstTech and FXPAL systems) and only one is threshold-based (CLIPS).

Figure 5.19 shows the frame-precision and frame-recall for all the runs submitted for each team. Again AT&T and Tsinghua systems are the best ones, not only have high values, but also all the runs are close between them. That is, despite the execution strategy the accuracy of their methods performs well. In the case of the three systems with similar outcome to us, all runs of TokyoInstTech are very similar (points are very close) with good precision but low recall. For CLIPS and FXPAL systems, we see that results are spread. This means that not necessarily their best GT detector is very accurate on the transitions it finds. All our runs have more or less the same accuracy. Our results are among the best ones.

In Table 5.6 we show the combination of all transitions, i.e., abrupt transitions and gradual transitions. In all cases, AT and GT detection and accuracy of GT, we are over the mean results of all the teams that participate in TRECVID Evaluation 2006. Even though it is the first time we participate in this Evaluation, the results obtained by our system are really encouraging. We have got better results than teams that have participated for many years in the evaluation.

Figure 5.20 shows all transitions compared with the results of other teams. Despite the problems occasioned for "other" gradual transitions the overall performance of our system is among the best teams. In TRECVID Evaluation 2O06 does not exist a ranking of the participants. Thus, it is not possible to refer to an official ranking and say in what position a team is positioned. With the objective to know in which position our system stays, we made



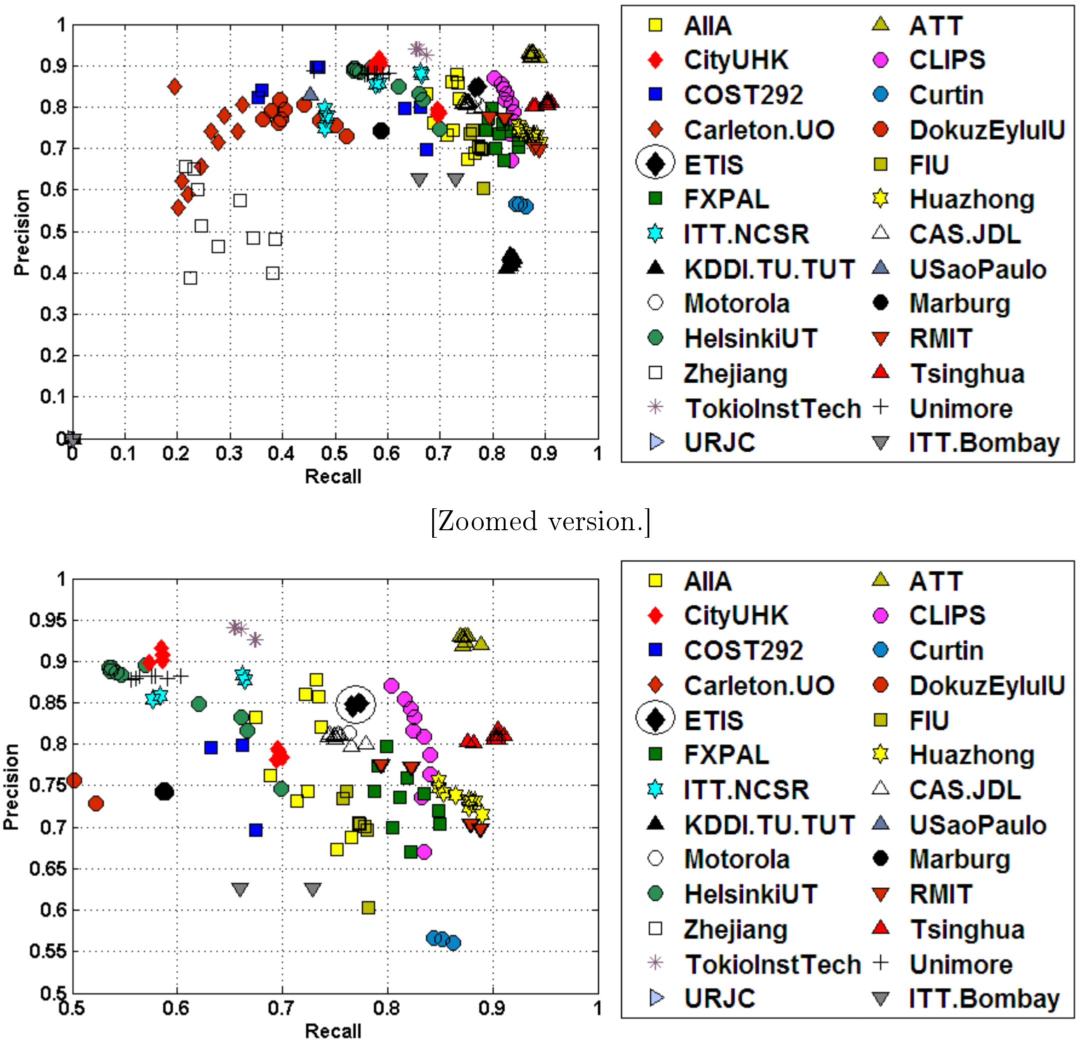
[All results.]

[Zoomed version.]

Figure 5.18: Precision/Recall measure of performance on the TRECVID 2006 for gradual transitions Smeaton and Over [2006].

an unofficial raking based on the $F1$ measure. In the case of AT detection we obtained the sixth best performance, in GT the fourth position, in accuracy of GTs we obtained the fourth position and in the overall performance we obtained the fourth position from a total of 26 participants. We insist in the fact that this is not an official ranking. For further information of the approaches of other teams visit `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`. You can also find the results in recall/precision of some of the teams.

In Figure 5.21, we show the runtime require for detecting ATs and GTs in the TRECVID 2006, i.e., the detection time of the shot boundaries for the 13 videos. Now a days the computer are getting faster and faster. It is better to have a method that requires less human adjusts than having something that it is computationally faster. So the computers can be fast and fast but human work is not so easy to get. Thus, that is the reason why we sacrifice computer cost to avoid human intervention (interaction).



[All results.]

[Zoomed version.]

Figure 5.19: Frame-Precision/Recall measure of performance on the TRECVID 2006 for gradual transitions Smeaton and Over [2006].

5.5 Conclusion

In this chapter we present our hierarchical system gradual transition detection. Our system is dedicated to detect dissolves and fades out-in transitions.

For dissolve detection, we use a pyramidal approach, i.e., we look for dissolves inside shots delimited by ATs and fades out-in boundaries. This means that in a first stage we need to detect ATs and fades. The hierarchical structure of our system allows us to reduce to two modalities of identification of GTs: fast motion or dissolve. Our approach consists in detect the possible dissolves using a modeling method, then extract features from the region of possible dissolve and finally use a SVM classifier to detect the dissolves. We investigate different features that characterize dissolve and improve a well-known method for dissolve detection. We reduce the size of the feature from 2D (frame content) to 1D (projection

Run	All transitions		
	Recall	Precision	F1
Etis1	0.757	0.876	0.812
Etis2	0.764	0.868	0.813
Etis3	0.768	0.888	**0.824**
Etis4	**0.771**	0.879	0.821
Etis5	0.771	0.851	0.809
Etis6	0.761	0.861	0.808
Etis7	0.769	0.878	0.820
Etis8	0.762	0.850	0.804
Etis9	0.751	**0.894**	0.816
Etis10	0.743	0.842	0.789
Mean Trecvid	0.668	0.687	0.677
Max Trecvid	0.855	0.892	0.873

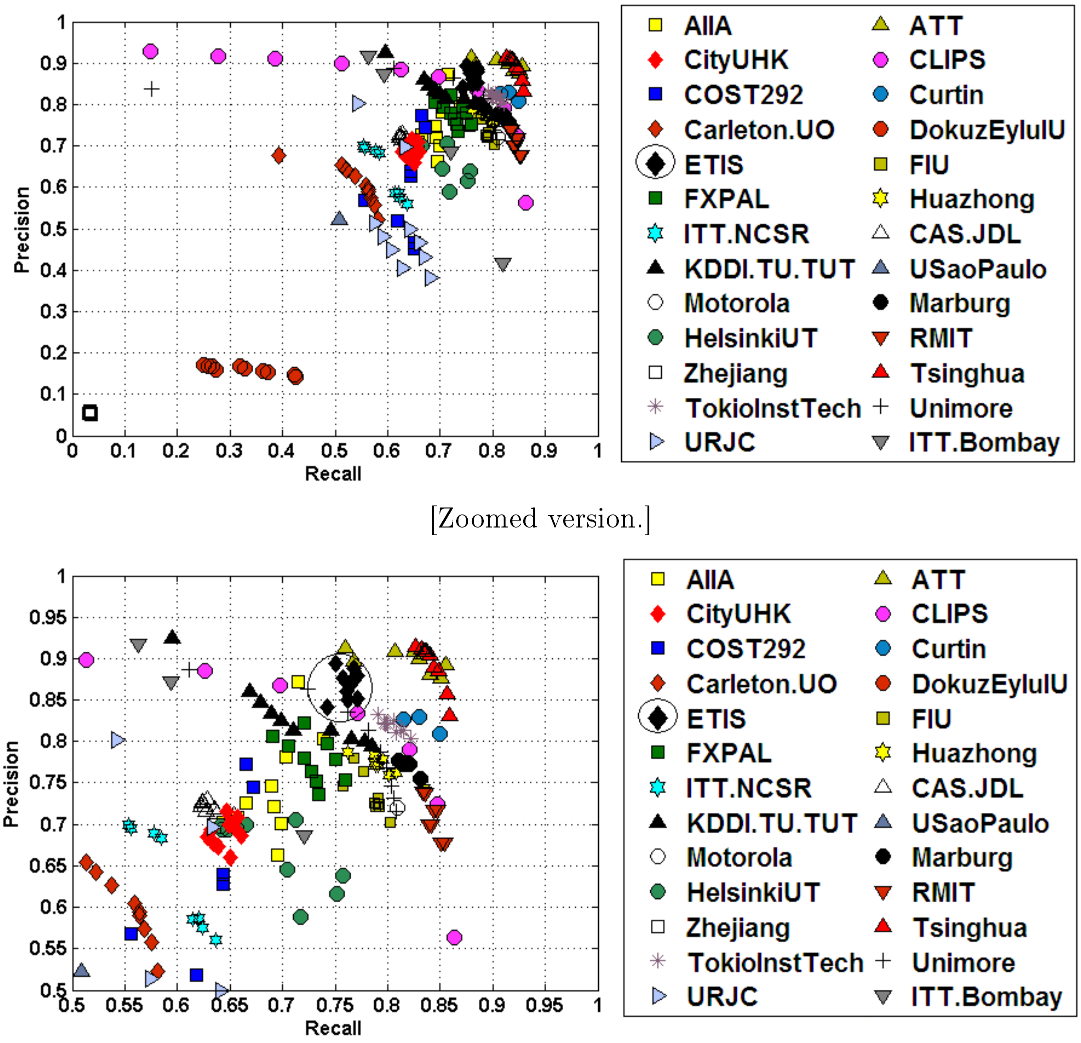
Table 5.6: Results for all runs for various settings

histograms), preserving its accuracy. Our experiments shows that the performance of the original method (DCD) is almost the same with our improved method (SD).

For fade out-in detection we use the modified method developed for dissolve detection. We use the modified method because fade is a special case of dissolves and also for the good performance showed in dissolve detection. We characterize a fade out-in first detecting the “black frames” and then reproducing two downward parabola patterns, one for fade-out an the other for fade-in. We do not use a machine learning approach because the detector has a single parameter to be set. This parameter is used for detect “black frames” that separate a fade-out from a fade-in.

Although our system detects only two types of GTs, we are among the best results. We thus improve dissolve detection as our results show it. The good results are not only limited to number of transitions detected, but also in the accuracy of the interval detected, i.e., how well the interval of the gradual transition detected match with the real transition.

[All results.]



[Zoomed version.]

Figure 5.20: Precision/Recall measure of performance on the TRECVID 2006 for all type of transitions Cámara-Chávez et al. [2006b].

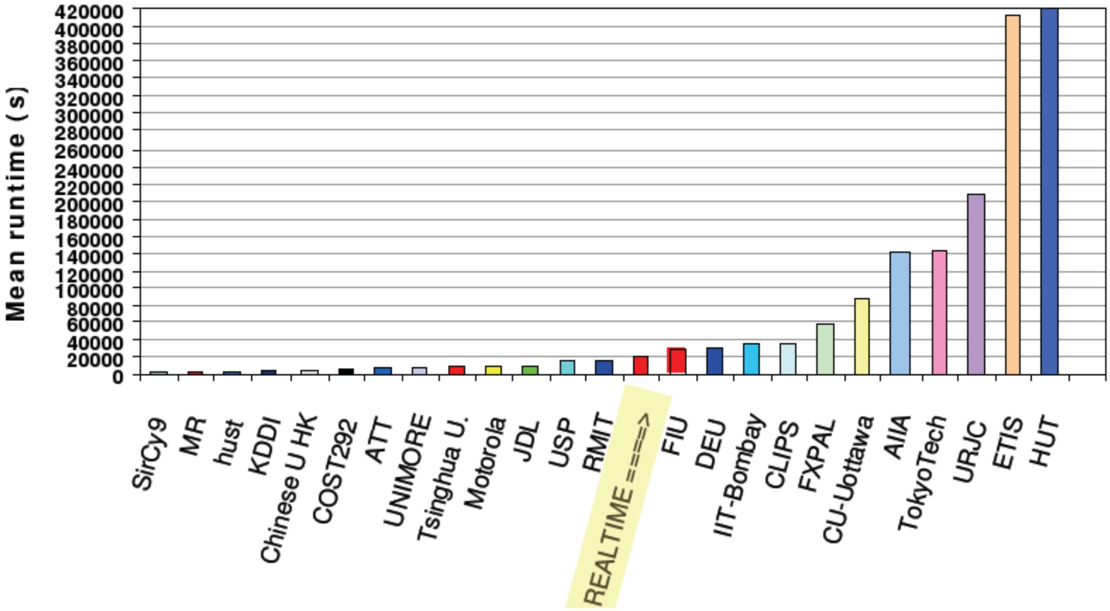


Figure 5.21: Mean runtime on the TRECVID 2006 Smeaton and Over [2006].

Chapter 6

Content-based video retrieval

With technology advances in multimedia, digital TV and information highways, a large amount of video data is now publicly available. However, without appropriate search technique all these data are nearly not usable. Traditionally, the main contents in many information retrieval systems are textual information. Text input is often the only mean for users to issue information requests. Systems accessible only through text input frequently frustrate users by providing abundant but irrelevant information. Users want to query the content instead of raw video data. For example, a user will ask for specific part of video, which contain some semantic information. Content-based search and retrieval of these data becomes a challenging and important problem. Therefore, the need for tools that can manipulate the video content in the same way as traditional databases manage numeric and textual data is significant.

6.1 Introduction

With the recent developments in technology, large quantities of multimedia data has become available in both public and proprietary archives. News videos, consisting of visual, textual and audio data, are important multimedia sources because of their rich content and high social impact. Most commercial video search engines such as Google, Blinkx, and YouTube provide access to their repositories based on text, as this is still the easiest way for a user to describe an information needed. The indices of these search engines are based on the filename, surrounding text, social tagging, or a transcript. This results in disappointing performance when the visual content is not reflected in the associated text because natural language is highly ambiguous. For example, describing an object such as an airplane in terms of its shapes and colors would be a demanding task, providing an example can give all the information that is required.

Numerous attempts have been made to represent and describe the visual world (a world without language) with inherent meaning, far more complex than words. The success of retrieval depends on the completeness and effectiveness of the indexes. Indexing techniques are determined by the extractable information through automatic or semi-automatic content extraction. The content-based image retrieval research community has emphasized a visual only approach. It has resulted in a wide variety of image and video search systems Flickner et al. [1995]; Gupta and Jain [1997]; Pentland et al. [1996]. Since video contains rich and multidimensional information, it needs to be modeled and summarized to get the most compact and effective representation of video data. A common denominator in these prototypes is that they first partition videos into a set of access units such as shots, objects or regions Deng and Manjunath [1998], and then follow the paradigm of representing video via a set of features (low-level visual information), such as color, texture, shape, layout and spatiotemporal features Al-Omari and Al-Jarrah [2005]; Shahraray and Gibbon [1997].

As shown in Figure 6.1, there are three processes that capture different levels of content information: The first is temporal segmentation to identify shot boundaries. At the second level each segment is abstracted into key-frames. Finally, visual features, such as color and texture, are used to represent the content of key-frames and in measuring shot similarity. Indexing is then supported by a learning process that classifies key-frames into different visual categories; this categorization may also support manual user annotation. These results composite the data set of video, which facilitate retrieval and browsing in a variety of ways.

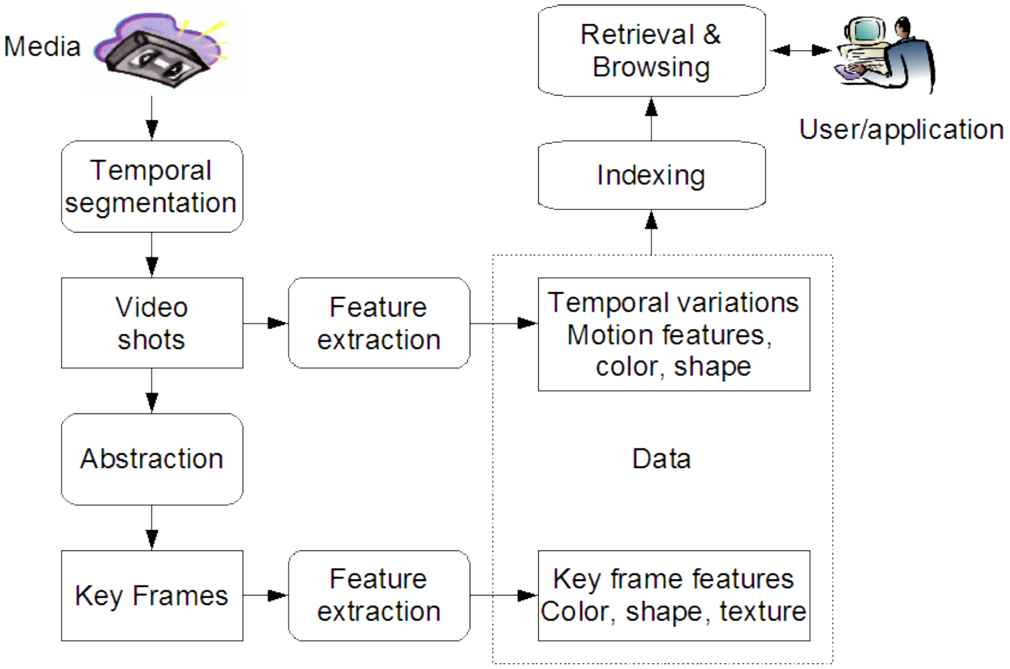


Figure 6.1: A diagram of an automated video indexing system.

While video browsing using key frames has been achieved for some applications, video retrieval, on the other hand, is still in its preliminary state and considered a hard problem. Besides lack of effective tools to represent and model spatial-temporal information, video retrieval has the same difficulties as traditional image retrieval. That is the so-called "semantic gap", utilizing low-level features for retrieval does not match human perception well in the general domain.

This means that low level features are easily measured and computed, but a high level query from a human is typically the starting point of the retrieval process. However, the semantic gap is not merely translating high level features from low level features. The essence of the semantic query consists in understanding the meaning that is behind the query. Therefore, this can involve understanding not only the intellectual side of human, but also the emotional side. For example, suppose we have two sets of pictures, one of “dogs” and the other of “birds”. If a search task looking for images that belong to “animal” category is executed, then images in these two sets should be considered similar. However, if the task consists in searching images of “dogs”, then the pictures with “birds” are not relevant. This means that the user is the only one who knows exactly what he is searching for and the system needs to learn the dissimilarity based on the user’s feedback.

This interactive stage (human-machine) contains two main steps: visualization and relevance feedback, which are iterated Smeulders et al. [2000]. The visualization step displays a selected set of images to the user. Based on his needs, the user judges how relevant those images are with respect to what the user is looking for. The perceptual similarity relies on the application, the person, and the context of usage. Therefore, the machine not only needs to learn the associations, but also has to learn them on-line with the user’s interaction in the loop.

However, the old problems of labor-intensive manual annotation and subjectivity of human perception still persist. The conventional relevance feedback algorithms converge slowly because users are led to label only the most relevant documents, which is usually not informative enough for systems to improve the learned query concept model.

Using learning is well-known in interactive content-based retrieval. Some comprehensive overviews of techniques are presented in Smeulders et al. [2000]; Zhou and Huang [2003]. Recently the use of support vector machines in learning has gained interest. It has proved to give the highest boost to the performance Chen et al. [2005, 2001]; Gosselin and Cord [2004a].

The video retrieval system described here simplifies the labeling task to identifying relevant key frames. The easiest way to reduce the labeling effort is to request the user to label some selected data, and automatically propagate the labels to the entire collection using a supervised learning algorithm. It greatly reduces the need for labeled data by taking advantage of active learning.

In this work, we show how the automatic video analysis techniques, such as shot boundary detection and key frame selection can be used in the content based video retrieval process. Therefore, our framework consists of:

1. *Shot boundary detection*: In the case of video retrieval, a video index is much smaller and thus easier to construct and use if it references video shots instead of every video frame. Shot transitions provide convenient jump points for video browsing. The detection of a shot change between two adjacent frames simply requires the computation of an appropriate continuity or similarity metric. Therefore, scene cut detection often performed by detecting transitions between shots, is required in the early stages of video indexing. In Chapters 4 and 5, we presented a broadly study of shot boundary detection.

2. *Key frame selection*: The predominant approach to automate the video indexing process is to create a video abstract. A video abstract is defined as a sequence of images ex-

tracted from a video, much shorter than the original yet preserving its essential message Lienhart et al. [1997b]. This abstraction process is similar to extraction of keywords or summaries in text document processing. That is, we need to extract a subset of video data from the original video such as key frames as entries for shots, scenes, or stories. As well as being less time consuming to produce than a textual annotation, a visual summary to be interpreted by a human user is semantically much richer than a text. Abstraction is especially important given the vast amount of data even for a video of a few minutes duration. The result forms the basis not only for video content representation but also for content-based video browsing. Using the key frames extracted in video abstraction, we can build a visual table of contents for a video or they can be used to index video.

3. *Retrieval process*: A video retrieval system generally consists of 3 components:

   a) *Feature extraction*: Most of the current video retrieval techniques are extended directly from image retrieval techniques. A typical example is the key frame based video indexing and retrieval systems. Image features such as color and texture are extracted from these key frames. Those features are used for indexing and retrieval.

   b) *Similarity measures*: A set of similarity measures, each of which captures some perceptively meaningful definition of similarity, and which should be efficiently computable when matching an example with the whole database. Compared with feature-based image retrieval, it is more difficult to combine multiple features to define the content similarity between two video sequences of shots for retrieval since more features (often with different levels of importance) are involved. Besides, content similarity comparison can be performed based on key-frame-based features, shot-based temporal and motion features, object-based features, or a combination of the three. There are several sophisticated ways to calculate the similarity measure: dynamic programming Dagtas et al. [2000], spatio-temporal matching Sahouria and Zakhor [1999]; Zhao et al. [2000], tree structure Yi et al. [2006], machine learning Adams et al. [2003]; Fan et al. [2004]; Naphade and Huang [2001]; Snoek et al. [2006a,b, 2005].

   c) *User interface*: A user interface for the choice of which definition(s) of similarity is necessary for retrieval, and for the ordered and visually efficient presentation of retrieved videos and for supporting user interaction.

Figure 6.2 depicts the structure followed by our system, RETINVID. This deals with video browsing based on shot detection, key frame extraction, indexing and content-based retrieval. The video browsing and retrieval can also be seen as a classification problem. From one or several frames brought by a user, the aim is to retrieve the shots illustrating the same concept. Key frame extraction is based on a clustering of each segmented shot. The closest frame to the cluster center is considered as a key frame. RETINVID is a complete system of video retrieval from the visual content. We have opted for an active learning scheme, which has proved its efficiency in content-based image retrieval Gosselin and Cord [2006], notably through the use of kernel functions.

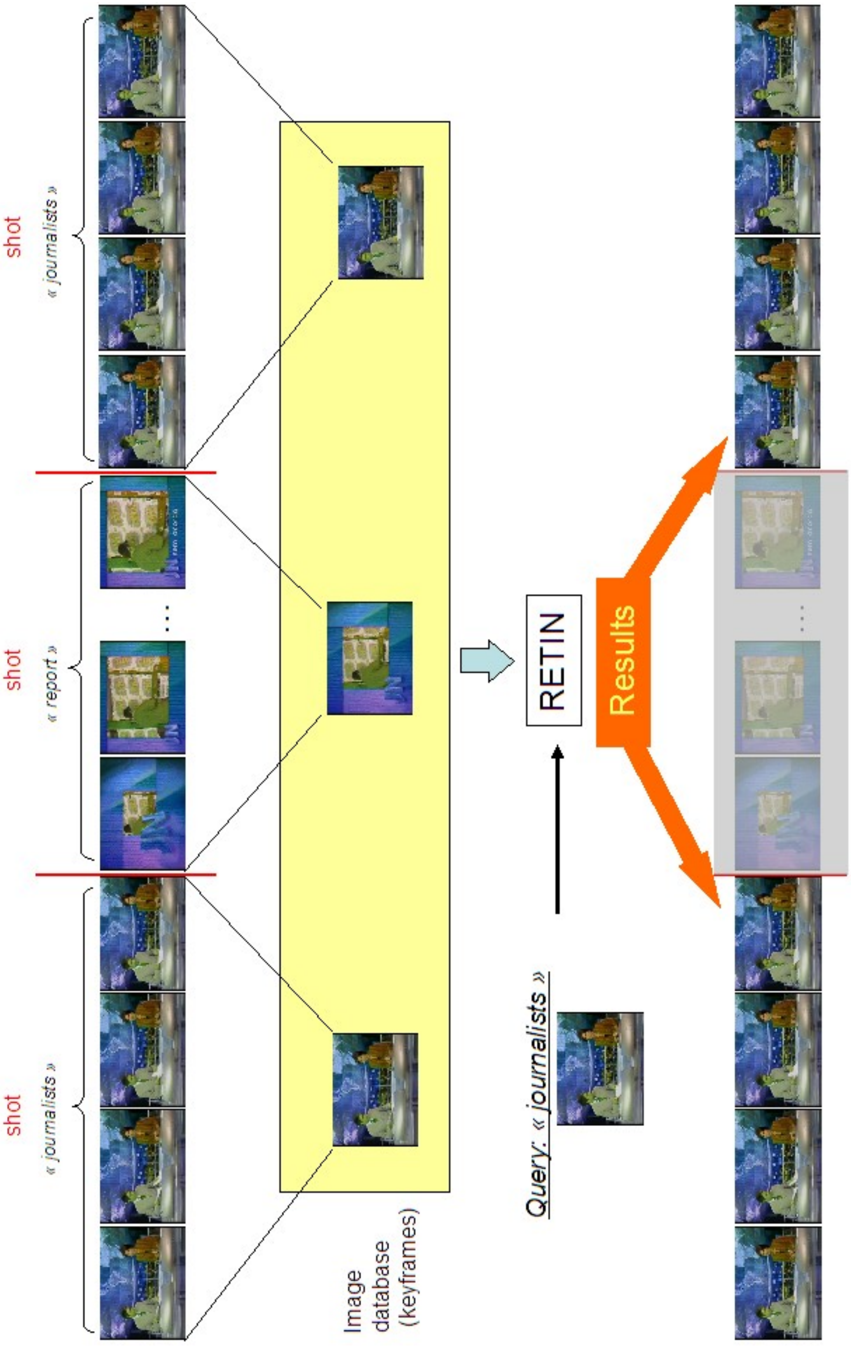


Figure 6.2: Content-based video retrieval schema.

The rest of this chapter is organized as follows. In Section 6.2, key frame extraction is presented, which consists in summarizing the shot content, this could be represented by one or more key frames, it would depend in the content complexity. Video indexing is presented in Section 6.3, the success of retrieval depends on a good indexation. In Section 6.4, we introduce the approach to active learning with support vector machines. A machine learning technique is used to improve performance in retrieval systems. In Section 6.5, we present the results of our proposed method and in Section 6.6, we discuss our conclusions.

6.2 Key frame extraction

Key frames provide a suitable abstraction and framework for video indexing, browsing and retrieval Zhuang et al. [1998]. One of the most common ways of representing video segments is by representing each video segment such as shot by a sequence of *key frame(s)* hoping that a "meaningful" frame can capture the main contents of the shot. This method is particularly helpful for browsing video contents because users are provided with visual information about each video segment indexed. During query or search, an image can be compared with the key frames using similarity distance measurement. Thus, the selection of key frames is very important and there are many ways to automate the process. There exist different techniques for key frame extraction Zhuang et al. [1998].

6.2.1 Key frame extraction techniques

In this section, we review some principal approaches for key frame extraction:

Shot boundary based approach

After video is segmented into shots, an easy way of key extraction is to use the first frame as the key frame Nagasaka and Tanaka [1992]. Although it is a simple method, the number of key frames is limited to one, regardless of the shot's visual content. A drawback of this approach if that the first frame normally is not stable and does not capture the major visual content.

Visual content based approach

This approach uses multiple visual criteria to extract key frames Zhuang et al. [1998].

- *Shot based criteria:* Selects a key-frame from a fixed position in the scene or several frames separated by a fixed distance Lu [1999]. Although this method considers only length of shots, the performance should be effective enough to save all the processing complexities and time needed to divide a shot into sub-shots and assign a key frame to them based on changes in contents Divakaran et al. [2002].

- *Color feature based criteria:* The current frame of the shot will be compared with the last key-frame. If significant content change occurs, the current frame will be selected as the new key-frame Zhang et al. [1997].

- *Motion based criteria*: The third criteria selects key-frames at local minima of motion Narasimnha et al. [2003]; Wing-San et al. [2004]. For a *zooming-like* shot, at least two frames will be selected: the first and last frame, since one will represent a global view, while the other will represent a more focused view. For a *panning-like* shot, frames having less than 30% overlap are selected as key-frames Zhang et al. [1997].

Motion analysis based approach

Wolf [1996] proposed a motion based approach to key frame extraction. First, the optical flow for each frame is calculated Horn and Schunck [1981], then a simple motion metric based on the optical flow is computed. Finally, the metric is used as a function of time in order to select key frames at the local minima of motion. The justification of this approach is that in many shots, the key frames are identified by stillness Wolf [1996].

Shot activity based approach

Gresle and Huang [1997] propose a shot activity based approach motivated by the same observation of Wolf [1996]. They first compute the intra and reference histograms and then compute an activity indicator. The local minima are selected based on the activity curve as the key frames Diklic et al. [1998]; Gresle and Huang [1997].

Clustering based approach

Clustering is a powerful technique used in various disciplines, such as pattern recognition, speech analysis, and information retrieval. In Ferman et al. [1998] and Zhang et al. [1997], key-frame selection is based on the number and sizes of the unsupervised clusters. Progress has been made in this area, however, the existing approaches either are computationally expensive or cannot capture adequately the major visual content Zhang et al. [1997]. A novel clustering approach based on statistical model is introduced by Yang and Lin [2005]. This method is based on the similarity of the current frame with their neighbors. A frame is important, if it contains more temporally consecutive frames that are spatially similar to this frame. The principal advantage of this method is that the clustering threshold is set by a statistical model. This technique is based on the method of Zhang et al. [1997] with the difference that the parameters are set by a statistical classifier.

Depending on the complexity of the content of the shot, one or more key frames can be extracted. For example, in the case of camera operations more than one key frame is needed, as it was explained in the motion based criteria for key frame extraction. Clustering is thus a good way to determinate both the most representative key frames, as well as their number. We based our unsupervised key frame detector in the method proposed by Yang and Lin [2005].

6.2.2 Features

Given a video shot $s = \{f_1, f_2, \ldots, f_N\}$ obtained after a shot boundary detection. Then, we cluster the N frames into G clusters, say $c_1, c_2, \ldots, c_G$. The similarity of two frames is defined as the similarity of their visual content, the color histogram of a frame is our visual content. The color histogram we used is the same computed with our ATs detector (see Chapter 4), i.e., we used a RGB color histogram (2 bits per channel). The similarity between frames i and j is defined by L1 norm.

Any clustering algorithm has a threshold parameter ρ which controls the density of clustering, i.e., the higher the value of ρ, the more the number of clusters. The threshold parameter provides a control over the density of classification. Before a new frame is classified into a certain cluster, the similarity between this node and the centroid of the cluster is computed first. If this value is less than ρ, this node is not close enough to be added into the cluster.

Our unsupervised clustering algorithm is based on one of the algorithms of the Adaptive Resonance Theory (ART) neural network family, Fuzzy ART G. Carpenter [1991].

6.3 Video indexing

Video indexing approaches can be categorized based on the two main levels of video content: *low level (perceptual)* and *high level (semantic) annotation* Djeraba [2002]; Elgmagarmid et al. [1997]; Lu [1999]; Tusch et al. [2000]. The main benefits of low-level feature-based indexing techniques are Tjondronegoro [2005]:

- They can be fully automated using feature extraction techniques (visual features).
- User can use similarity search using certain features characteristics.

However, feature-based indexing tends to ignore the semantic contents, whereas users mostly want to search video based on the semantic rather than on the low-level characteristics. There are elements beyond perceptual level, which can make feature based-indexing very tedious and inaccurate. For example, users cannot always describe the characteristics of certain objects they want to retrieve for each query.

The main advantage of high-level semantic-based indexing is the possibility to achieve a query more natural, powerful and flexible. For example, users can browse a video based on the semantic hierarchy concepts and they can search a particular video according to the keywords. Unfortunately, this type of indexing is often achieved using manual intervention as the process of mapping low-level features to semantic concepts is not straight forward due to the *semantic gap*. Manual semantic annotation should be minimized because it can be very time-consuming, biased and incomplete Ahanger and Little [2001]; Leonardi [2002]; Snoek and Worring [2005].

There are three major indexing techniques Tjondronegoro [2005]: feature-based video indexing (including shot-based, object-based, and event-based indexing), annotation-based video indexing, and indexing approaches which aim to bridge semantic gap.

6.3.1 Feature-based video indexing

This type of indexing can be categorized based on the features and extracted segments.

Segment-based indexing techniques

During the process of indexing texts, a document is divided into smaller components such as sections, paragraphs, sentences, phrases, words, letters and numerals, and thereby indices can be built on these components Zhang [1999]. Using the same concept, video can also be decomposed into a hierarchy similar to the storyboards in filmmaking Zhang [1999]. For example, a *hierarchical video browser* consists of a multi-levels abstraction to help users in finding certain video segments. This type of browsing scheme is often called storyboard, contains a collection of frames that represent the main concepts in the video. An advantage of storing key-frames is that they require less storage space than the whole video.

Figure 6.3 shows a storyboard indexing for hierarchical video browsing. A video contains *stories*, for example, a birthday party, a vacation, a wedding, etc. Each of the stories contains a set of *scenes*, for example a vacation story contains the preparation of the travel and touristic places scenes. Each scene is then partitioned into *shots*, i.e., shots of the different places visited. Then, a *scene* is a sequence of shots that correspond to a semantic content, and a *story* is a sequence of scenes that reveals a single amusing semantic story. In Snoek and Worring [2005] we can find a review of this approach.

Object-based video indexing techniques

Object-based video indexing aims at distinguishing particular objects throughout video sequence to capture content changes. In particular, a video scene is defined with a complex collection of objects, the location and physical attributes of each object and the relationship between them.

Objects extraction process is more complex than extracting low-level features such as color, texture and volume. However, the process on video can be considered easier as compared to an image because an object region usually moves as a whole within a sequence of video frames.

Event-based video indexing techniques

Tracking activity of objects, events in video segments. Event-based video indexing aims at detecting interesting events from video track Zeinik-Manor and Irani [2005]. However, there is not yet a clear definition for “event” itself for video indexing. Event can be generally defined as the relations between appearing objects in a time interval that may occur before or after the other event Babaguchi et al. [2002]. Event can also be defined as long-term

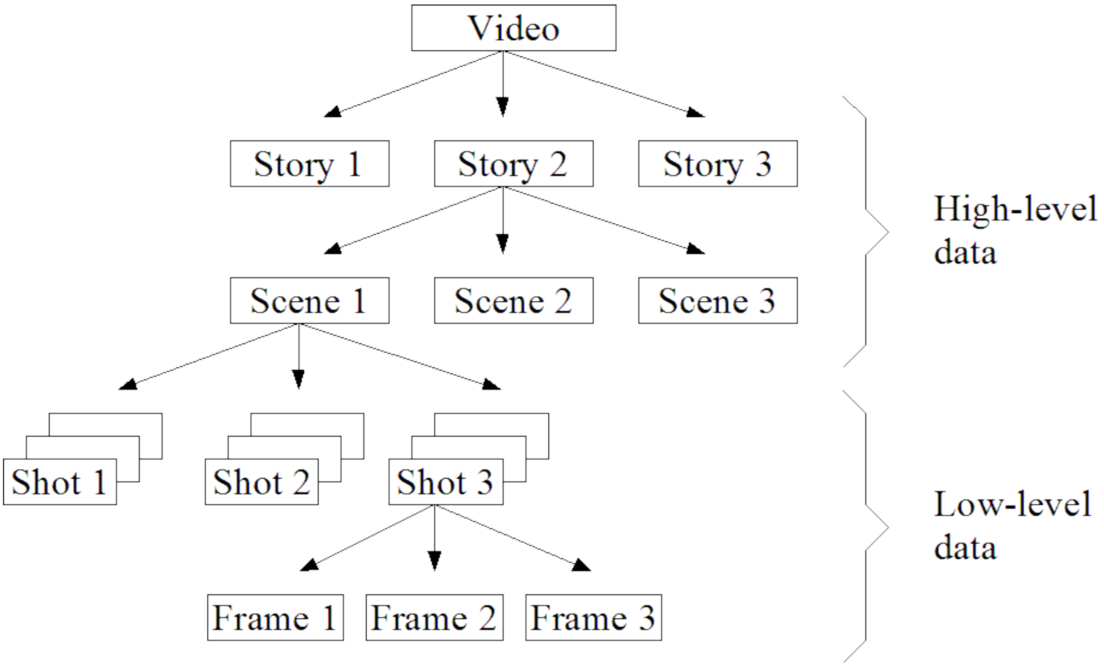


Figure 6.3: Segment-based indexing Tjondronegoro [2005].

temporal objects which are characterized by spatial-temporal features at multiple temporal scales, usually over tens or hundreds of frames. An event includes a) *temporal textures* such as flowing water: indefinite spatial and temporal type, b) *activities* such as person walking: temporally periodic but spatially restricted and c) *isolated motion events* such as smiling.

6.3.2 Annotation-based video indexing

Another alternative for managing video is to annotate the semantics of video segments using keywords or free texts. Thus, user queries can be managed using standard query language, such as SQL and browsing can be based on hierarchical topic (or subject) classification [10, 64]. However, the major limitation of this approach is the fact that it would be extremely tedious and ineffective to manually annotate every segment of video. On the other hand, the process of mapping low-level video features into high-level semantic concepts is not straight forward.

There are also some major drawbacks which can already be expected from annotation-based indexing:

- Keywords/free text selection is subjective and often depends on application and domain requirements.
- Words are often not able to fully describe a single frame therefore it is expected that words will be extremely insufficient to describe a video segment.
- When users do not know how to explain what they want using words, it is often the case that they would like to query based on a similar image or sound. Similarly in browsing

a video document, users often find that visual key frames representation is more helpful and interesting compared to pure texts.

6.3.3 Indexing by bridging semantic gap

The objective is to bridge the semantic gap between high-level concepts and low-level features. Audio-visual feature extraction is easier than semantic understanding, and thus generally possible to be fully automated. Content-based video retrieval can be benefited from *query-by-example* (QBE). For example, given a sample video shot, the system should find the indexed segments which have the closest characteristics such as similar speaker pitch and similar face. The usage of QBE has been demonstrated in news applications Satoh et al. [1999] by associating faces and names in news videos. To accomplish this task, their system uses face sequence extraction and similarity evaluation from videos, name extraction from transcripts, and video-caption recognition.

QBE assumes that when video frames are represented by key frames, retrieval can be performed by users selecting the visual features, and the specified weights on each feature when more than one feature is used. The retrieval system then finds images similar to the query. Such systems are not always satisfactory due to the fact that best feature representation and manually assigned weights are sometimes not sufficient to describe the high-level concepts in queries. In the QBE paradigm, two tasks are dominant. The first is to produce a compact signature representation of video segments (normally a segment is one camera shot). The second is to provide algorithms to compare different signatures from different segments. For example, most users think with high-level concepts such as "a vase", rather than the shape and textures. After its success in text-based retrieval, relevance feedback has been tested for image retrieval systems Lu et al. [2000]; Rui et al. [1998].

Even though relevance feedback does not map low-level features with high-level semantic, it aims to adjust an existing query automatically. This is achieved by using the information feedback provided by the users about the relevance of previously retrieved objects so that the adjusted query is a better approximation of user's need. Thus, relevance feedback technique tries to establish the link between these features based on users' feedback. The burden of specifying the weights is removed from the user as they need to mark images that are relevant to the query. The weights are dynamically embedded in the query to represent the high-level concepts and perception subjectivity.

The conventional relevance feedback algorithms converge slowly because users are led to label only the most relevant documents, which is usually not informative enough for systems to improve the learned query concept model. Recently, active learning algorithms have been proposed to speed up the convergence of the learning procedure Schohn and Cohn [2000]; Tong [2001]. In active learning, the system has access to a pool of unlabeled data and can request the user's label for a certain number of instances in the pool. However, the cost of this improvement is that users must label documents when the relevance is unclear or uncertain for the system. These "uncertain documents" are also proven to be very informative for the system to improve the learned query concept model quickly Xu et al. [2003]. Recently, active learning is being used on image retrieval systems Chang et al. [2005]; Cord et al. [2007]; Gosselin and Cord [2006, 2004b] and video analysis Qi et al. [2006]; Song et al. [2006]; Yang and Hauptmann [2006].

6.4 Active learning

The idea is to improve the classifier by asking users to label informative shots and adding the labeled shots into the training set of the classifier. The major difference between conventional relevance feedback and active learning is that the former only selects top-ranked examples for user labeling, while the latter adopts more intelligent sampling strategies to choose informative examples from which the classifier can learn the most. A general assumption on the informativeness of examples is that an example is more useful if the classifier's prediction of it is more uncertain. Based on this assumption, active learning methods typically sample examples close to the classification hyperplane. Another general belief is that a relevant example is more useful than an irrelevant one especially when the number of relevant examples is small compared to that of the irrelevant ones.

Optimized training algorithms are able to cope with large-scale learning problems involving tens of thousands of training examples. However, do not solve the inherent problem which consists in the fact that conventional supervised machine learning relies on a set of patterns which have to be assigned to the correct target objects. In many applications, the task of assigning target objects cannot be accomplished in an automatic manner, but depends on time-consuming and expensive resources, such as complex experiments or human decisions. Hence, the assumption that a set of labeled examples is always available, does not take into account the labeling effort which is necessary in many cases.

Let us consider the *pool-based active learning* model (see Figure 6.4), which was originally introduced by Lewis and Catlett [1994a] in the context of text classification learning. We refer to the pool-based active learning model as active learning herein after to simplify our presentation. The essential idea behind active learning is to select promising patterns from a given finite set U (also referred as the pool of unlabeled examples) in a sequential process in the sense that the corresponding target objects contribute to a more accurate prediction function. The active learning algorithm sequentially selects patterns from set U and requests the corresponding target objects from a teacher component (also referred to as oracle). In contrast to standard supervised learning, pool-based active learning considers an extended learning model in which the learning algorithm is granted to access to a set of unlabeled examples. The learning algorithm is provide with the ability to determine the order of assigning target objects with the objective of attaining a high level of accuracy without requesting the complete set of corresponding target objects. Moreover, the stopping criterion can be of dynamic nature and depends on a measure of the learning progress or be of static nature such as a fixed number of requested target objects.

The problem of labeling effort in supervised machine learning arises naturally in many fields of application. The crucial point in active learning is that by ordering the sequential

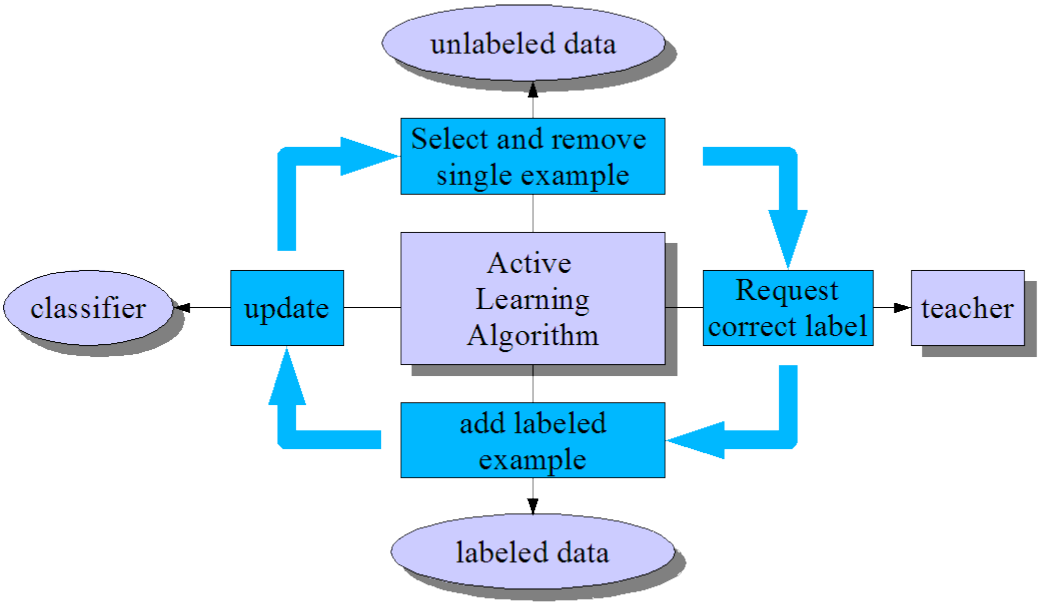


Figure 6.4: Pool-based active learning: an extended learning model in which the learning algorithm is granted access to the set of unlabeled examples and provided with the ability to determine the order of assigning target objects Brinker [2004].

process of requesting target objects with respect to an appropriate measure of the information content, it is possible to reduce the labeling effort. In many applications, active learning achieves the same level of accuracy as standard supervised learning, which is based on the entire set of labeled examples, while only requesting a fraction of all the target objects.

The goals of active learning can be summarized as follows:

- improve the utility of the training set, i.e., make better use of the information that is available from the current training data with the aim to use less training data than passive learning to achieve the same generalization ability.
- improve the cost efficiency of data acquisition by labeling only those data that are expected to be informative with respect to the improvement of the classifier's performance.
- facilitate training by removing redundancy from the training set.

6.4.1 Basic main algorithms

The typical active learning settings consist of the following components Tong [2001]: an unlabeled pool U, an *active learner* l composed of three components, (f, q, X). The first component is a classifier, $f : X \rightarrow [-1, 1]$, trained on the current set of labeled data X (typically few). The second component $q(X)$ is the querying function that, given a current labeled set X, decides which example in U to query next. The active learner can return a classifier f after each query or after some fixed number of queries. Figure 6.5 illustrates the framework of active learning. The query function q selects informative data from the unlabeled pool, then users annotate the selected data and feed them into the labeled data set. Given the labeled data X, the classifier f is trained based on X.

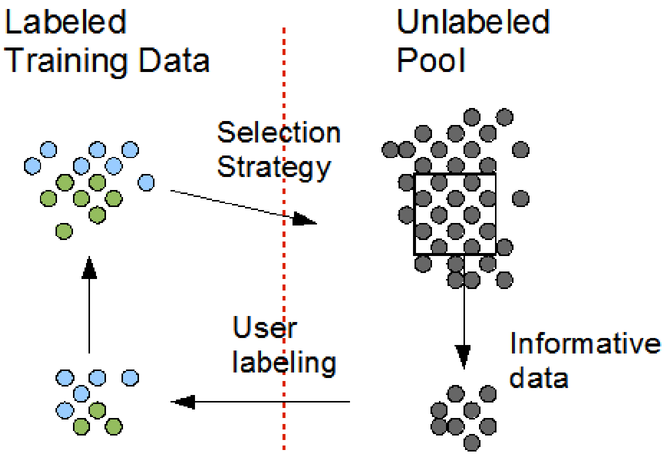


Figure 6.5: Illustration of basic learning Hauptmann et al. [2006].

In Algorithm 6.1, we show the pool-based active learning algorithm, where the basic three operations are: sampling (query function), user labeling and training.

Algorithm 6.1: Algorithm of pool-based active learning.

```
while a teacher can label examples do
    Apply the current classifier to each unlabeled example;
    Find the $m$ examples which are the most informative for the classifier ;
    Let the teacher label the $m$ examples ;
    Train a new classifier on all labeled examples;
end
```

In 2000, two groups proposed an algorithm for SVMs active learning Schohn and Cohn [2000]; Tong and Koller [2000]. Algorithm 6.2 describes the selection process proposed by them. This corresponds to step 4 in Algorithm 6.1.

Algorithm 6.2: Selection Algorithm.

```
while a teacher can label examples do
    Compute $f(x_i)$ over all $x_i$ in a pool;
    Sort $x_i$ with $|f(x_i)|$ in decreasing order;
    Select top $m$ examples ;
end
```

The query function is the central part of active learning process and active learning methods differ in their respective query functions. There exist two broad approaches for query function design Li and Sethi [2006]:

1. statistical learning approach: query function is designed to minimize future errors Cohn et al. [1996]. They take a probabilistic approach by picking examples that minimize the generalization error probability. The statistical learning approach is also used by Fukumizu [2000] for training multilayer-perceptron networks to perform regression;

2. pragmatic approach: some sort of minimization is performed without directly considering future performance. An early example of this approach is query by committee Freund et al. [1997], the unlabeled example to be picked is the one whose predicted label is the most ambiguous. Their choice of the query function is related to reducing the size of the version space. Tong and Koller [2000] suggest a querying approach based on version space splitting and apply it for text classification. They query examples closest to the decision boundary, this method is known as "simple margin" scheme. The objective is to reduce the version space under the assumption that it is symmetric. Similar schemes that query samples close to boundary are proposed by Schohn and Cohn [2000] and Campbell et al. [2000]. Another example is the uncertainty sampling scheme of Lewis and Catlett [1994b] where the example picked is the one with the lowest certainty.

This research proposes an approach to *active learning* for content-based video retrieval. The goal of active learning when applied to content-based video retrieval is to significantly reduce the number of key frames annotated by the user. We use active learning to aid in the semantic labeling of video databases. The learning approach proposes sample video segments to the user for annotation and updates the database with the new annotations. It then uses its accumulative knowledge to propagate the labels to the rest of the database, after which it proposes new samples for the user to annotate.

6.4.2 Active learning for video retrieval

When comparing results of fully automated video retrieval to interactive video retrieval Hauptmann and Christel [2004] in TRECVID evaluation, there is a big difference in performance. The fully automated search (no user in the loop) succeeds with good recall for many topics, but low precision because relevant shots tend to be distributed throughout the top thousands in the ordered shot list, causing the standard metric of mean average precision (MAP, which is the area under the Precision/Recall curve) for automated search to fall behind almost any interactive system. One explanation is that query finds the relevant stories, but finding the individual relevant clips is very difficult. Interactive system performance Smeaton et al. [2006] appears strongly correlated with the system's ability to allow the user to efficiently survey many candidate video clips (or key frames) to find the relevant ones. Interactive systems allow the user to annotate video shots, look at the results, improve the query by choosing relevant shots and iterate in this by reformulating or modifying the query Hauptmann and Christel [2004]; Smeaton et al. [2006]; Snoek et al. [2006a].

Vasconcelos and Kunt [2001] divide retrieval techniques in two categories: statistical and geometrical. Geometrical methods are based on the calculation of similarity between a query, usually represented by an image, and the images of the database Rui and Huang [2000a].

Statistical methods are based on the update of relevance function or a binary classification of images using the user labels. The relevance function estimation approach aims to associate a score to each image, expressing by this way the relevance of the image to the query Cox et al. [2000]. The binary classification approach uses relevant and irrelevant images as input training data Chapelle et al. [1999]. This approach has been successfully used in the context-based image retrieval Tong [2001].

We focus on statistical learning technique for image retrieval, specifically a binary classification method adapted to image retrieval. The classification in content-based image retrieval context has some specifies Gosselin and Cord [2005]: the input dimension is usually very high, the training set is small compared with the test set (the whole database), the training data set grows step by step due to user annotations, unlabeled data are available, and limited computation time. We also deal with these characteristics in the context of content-based video retrieval. Therefore, we use the *RETIN* system, a content-based search engine image retrieval Gosselin and Cord [2006], for content-based video retrieval: *RETINVID*. This system belongs to binary classification approach, which is based on SVM classifier and on an active learning strategy Cohn et al. [1996].

6.4.3 RETIN system

This system is based on SVM_{active} method Tong and Chang [2001] which query examples closest to the decision boundary. In content-based image retrieval, the training set remains very small, even after interaction where new labeled examples are added, in comparison to the whole database size. In that context get a reliable estimation of the boundary constitutes a major problem. In this particular context, statistical techniques are not always the best ones. Cord et al. [2007] propose a heuristic-based correction to the estimation of f close to the boundary.

Let $(x_i)_{i\in\{1,\dots,n\}}$, $x_i \in \mathbb{R}$ be the feature vectors representing images from the database, and $x_{(i)}$ the permuted vector after a sort according to the function f (Equation 4.19). At the feedback step j, SVM_{active} proposes to label m images from rank s_j to s_{j+m-1}:

$$\underbrace{x_{(1),j}}_{\text{most relevant}}, x_{(2),j}, \dots, \underbrace{x_{(s_j),j}, \dots, x_{(s_{j+m-1}),j}}_{\text{images to label}}, \dots, \underbrace{x_{(n),j}}_{\text{less relevant}}$$

While the strategy of SVM_{active} consists in selecting s_j from the images that are closer to the SVM boundary, Cord et al. [2007] propose to use the ranking operation. The drawback of the former is that the boundary changes a lot during the first iterations, while the ranking operation persists almost stable, this characteristic is exploited by the latter. In fact, they suppose that the best s allows to present as many relevant images as irrelevant ones. In their method, the selected images are restricted to be well balanced between relevant and irrelevant images, then s_j is considered good. Therefore, they exploit this property to adapt s during the feedback step.

In order to maintain the training set balanced, they adopt the following upgrade rule for $s_{j+1} : s_{j+1} = s_j + h(r_{rel}(j), r_{irr}(j))$, where r_{rel} and r_{irr} are the number of relevant and irrelevant labels, respectively. $h(.,.)$ is a function which characterizes the system dynamics where $h(x, y) = k(x - y)$. Through this rule, they ensure to maintain the training set s balanced, increasing the set when $r_{rel} > r_{irr}$ and decreasing in the other case.

With the objective to optimize the training set, they increase the sparseness of the training data. In fact, nothing prevents to select an image that is closer to another (already labeled or selected). To overcome this problem, m cluster of images from $x_{(s_j),j}$ to $x_{(s_j+M-1),j}$ (where $M = 10m$ for instance) can be computed using an enhanced version of Linde-Buzo-Gray (LBG) algorithm Patanè and Russo [2001]. Next, the system selects for labeling the most relevant image in each cluster. Thus, images close to each other in the feature space will not be selected together.

6.4.4 RETINVID system

Our content-based video retrieval system consists of 3 basic steps: video segmentation (cf. Chapters 4 and 5), key frame extraction (cf. Section 6.2) and video indexing (cf. Section 6.3). Figure 6.6 illustrates our framework. First, the video is segmented into shot detecting the ATs and GTs. From each shot, a key frame extraction is executed. One or more key frames could represent the content of the shot, it depends on the complexity of the shot content. Then, we extract color and texture features from the key-frames. We perform the feature extraction implemented in RETIN system. We used Color L^*a^*b and *Gabor* texture features Philipp-Foliguet et al. [2006] for still images and the Fourier-Mellin and Zernike moments extracted for shot detection. For the active classification process, a SVM binary classifier with specific kernel function is used. The interactive process starts with a coarse query (one or a few frames), and allows the user to refine his request as much as necessary. The most popular way to interact to the system is to let the user annotate examples as relevant or irrelevant to his search. The positive and negative labels are then used as examples or counterexamples of the searched category. The user decides whether to stop or continue with the learning process. If the user decides to continue, new examples are added to the training set and the classification process is iterated. Finally, if the user decides to stop, the final top similarity ranking is presented to him.

6.5 Experiments

A potentially important asset to help video retrieval and browsing is the ability to automatically identify the occurrence of various semantics features such as “Indoor/Outdoor”, “People”, etc., which occur in video information. In this section, we present the features and parameters set used for our content-based video retrieval system.

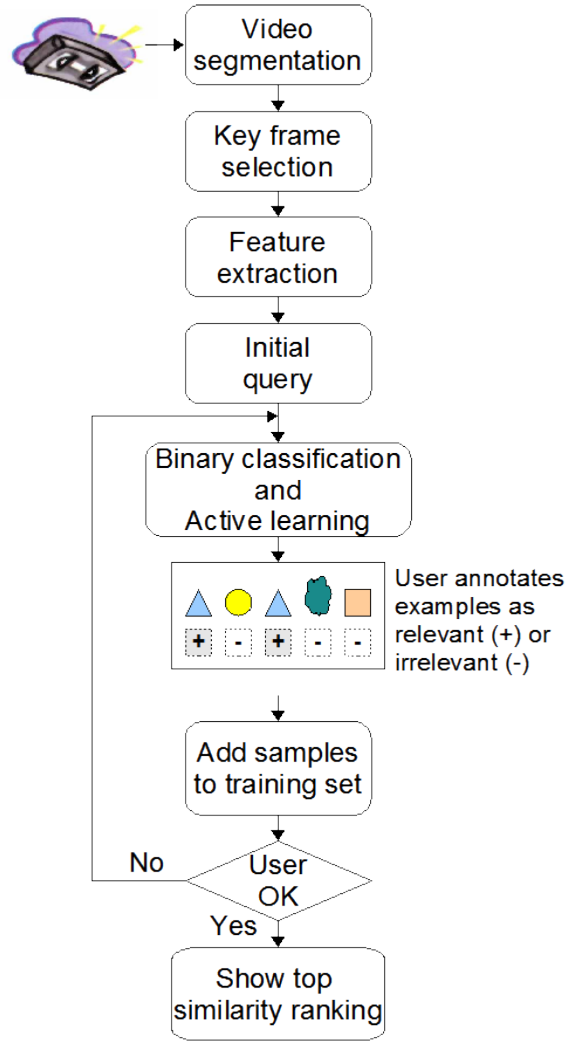

Video segmentation
Key frame selection
Feature extraction
Initial query
Binary classification and Active learning
+
-
+
-
-
User annotates examples as relevant (+) or irrelevant (-)
Add samples to training set
No
User OK
Yes
Show top similarity ranking

Figure 6.6: RETINVID System.

6.5.1 Data set

We use the TRECVID-2005 data set for high level feature task. Given a standard set of shot boundaries for the feature extraction test collection and a list of features definitions, participants are asked to return for each chosen feature, the top ranked video shots (ranked according to the system's confidence). The presence of each feature is assumed to be binary, i.e., it is either present or absent in the given standard video shot.

The features to be detected are defined (briefly) as follows and are numbered 38-47: (38) People walking/running, (39) Explosion or fire, (40) Map, (41) US Flag, (42) Building exterior, (43) Waterscape/ waterfront, (44) Mountain, (45) Prisoner, (46) Sports, (47) Car.

The feature test collection for TRECVID-2005 high level task contains 140 videos and 45,765 reference shots. The features were annotated using a tool developed by Carnegie Mellon University.

6.5.2 Features and parameters

Color, texture and shape information are used to perform the high level task. We used color L^*a^*b, Gabor texture (features provided by RETIN system) and the Fourier-Mellin and Zernike moments extracted for shot detection.

Features provided by RETIN system are statistical distributions of color and textures resulting from a dynamic quantization of the feature spaces. That is, the color and texture space clusterings are used to compute the image histograms. The clustering process is performed using the enhanced version of LBG algorithm. The main problem is to determinate the number of bins, i.e., the number of clusters.

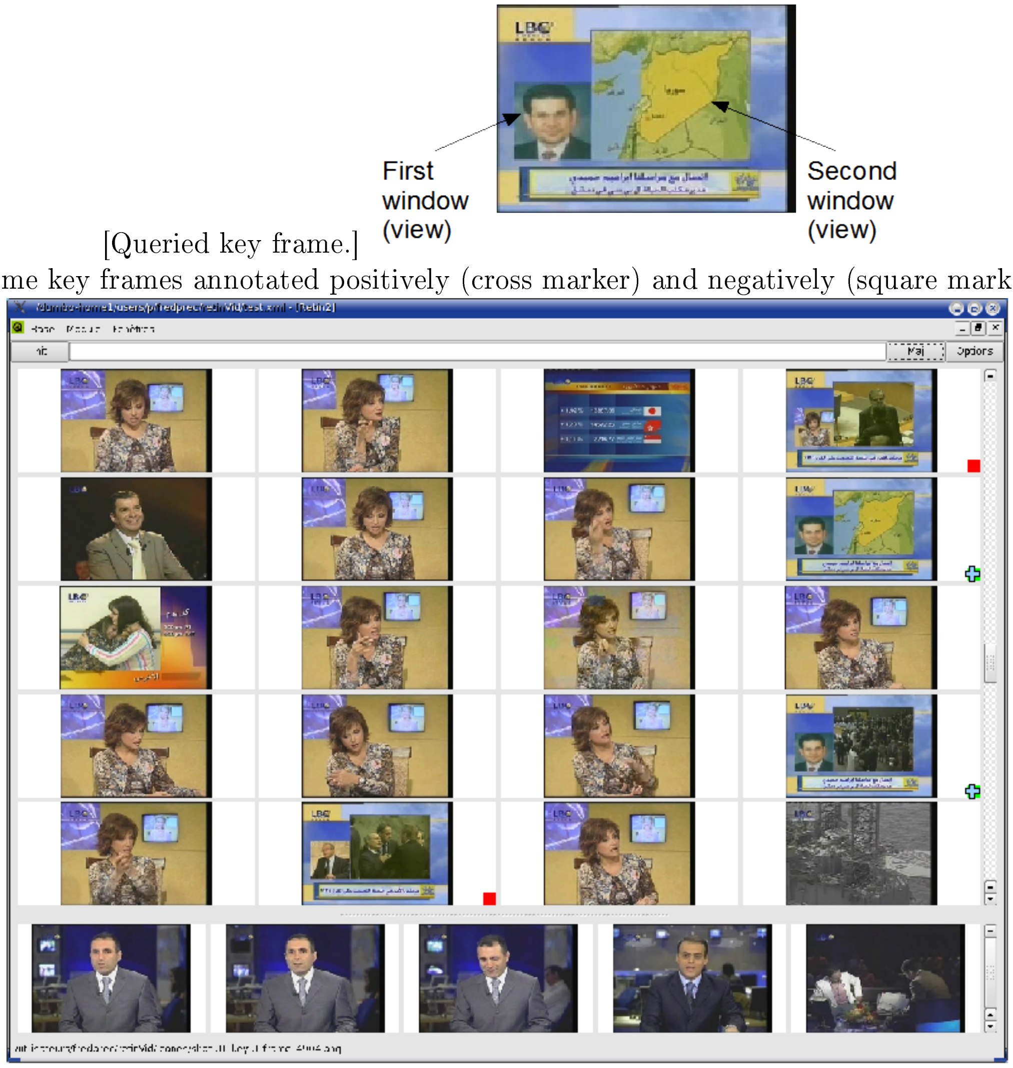
Different studies were performed in order to determine the number of histogram bins. Brunelli and Mich [2001] have evaluated many feature histograms and concluded that histograms with small number of bins are reliable. For color histograms, Tran and Lenz [2001] suggest to use around of 30 bins. Fournier et al. [2001] performed many experiments, using different numbers of clusters for dynamic quantization of feature space, and confirm all these prepositions. An interesting characteristic and also the major advantage of dynamic approach is that it is possible to reduce the size of the feature without performance degradation. Therefore, we have adopted the dynamic quantization with 32 classes, i.e., 32 for color and 32 for texture. In the case of shape descriptors, as we use the features extracted for shot boundary detection, the number of bins for Zernike moments are 11 bins and for Fourier Mellin are 24 bins.

When distributions are used as feature vectors, a Gaussian kernel gives excellent results in comparison to distance-based techniques Gosselin and Cord [2004b]. That is also confirmed in the excellent performance of Gaussian-χ^2 kernel for shot boundary detection Cámara-Chávez et al. [2007]. Thus, we use this kernel associated to SVM to compare key frames and compute classification. The number m of key frames labeled at each interactive feedback is set to $m = 10$. The number of feedbacks is set to 25.

6.5.3 Evaluation

The active strategy is implemented through an "active" window, which proposes the most useful key frames for annotations (Figure 6.5.3). The interface is composed on one hand of the key frames ranked by relevance result and on the other hand of a few key frames, which are at the very brink of the category. The lower window displays the key frames to be labeled during the learning process. The upper one (the bigger one) is the final window, where the key frames are displayed according to their relevance. These key-frames are the most likely to make the category boundary rapidly evolve towards the solution.

Figures 6.7 and 6.8 show the performance of our system. In Figure 6.5.3, the queried key frame is shown. The key frame has the following characteristics: two windows (views), the first window at the left of the key frame presents a young reporter and, the second window (the bigger one) situated at the the right of the key frame may contains different scenes. The only constrain of the query is that the key frame must contain a young reporter in the first



[Queried key frame.]

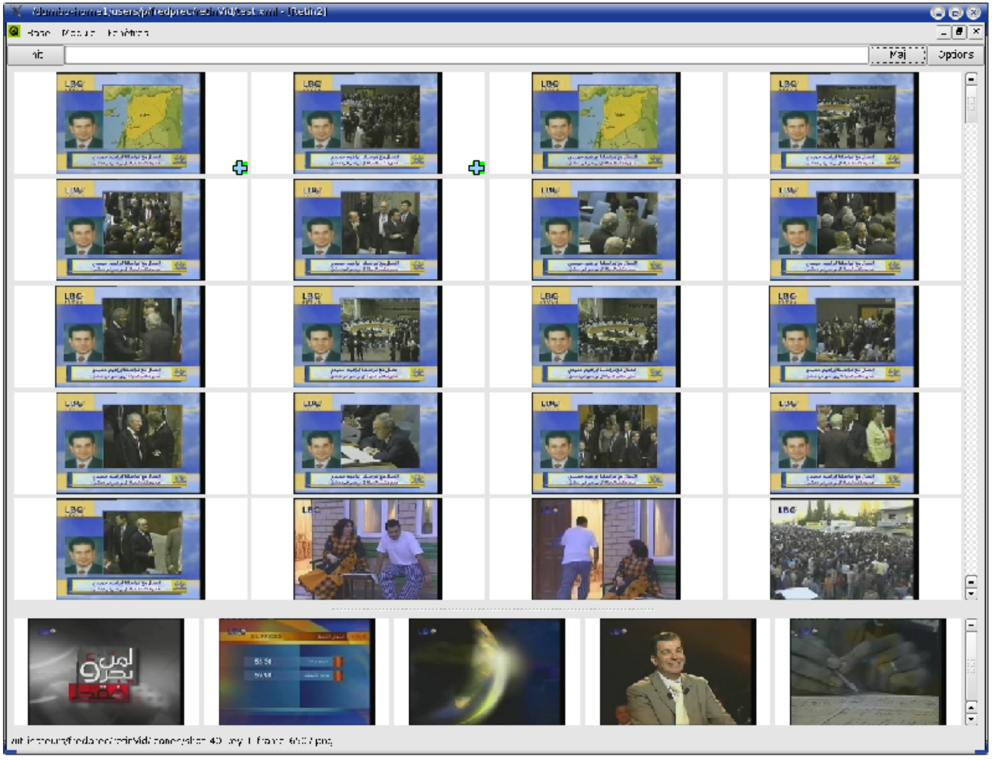
[Some key frames annotated positively (cross marker) and negatively (square marker).]

Figure 6.7: RETINVID Interface.

window. In Fig. 6.5.3, the user initializes the query and annotates key frames (the markers are at the right of the keyframe). The user annotates positively (cross marker) two key frames where the first window shows a young reporter and negatively (square marker) other two key frames where the small window shows a lady and an older reporter, respectively. Figure 6.8 shows the key frames retrieved according to their relevance. Figure 6.5.3 displays the most relevant key frames. At the beginning of these top ranked key frames are the two positive labeled key frames. The most relevant key frames have the same characteristics of the queried key frame. That is, key frames with two windows where the first window presents a young reporter and the second window may show any content as in Figure 6.5.3. In Figure 6.5.3 the less relevant key frames are shown. The last key frames are the ones that were labeled as negative. This example shows the power of our retrieval system. That is, it is capable to retrieve the desired query even though the positively and negatively labeled key frame are very similar.

Now we show the results of the experiments where we retrieve the key frames from

[RETINVID Results: top ranked relevant key frames.]



[RETINVID Results: less relevant key frames.]

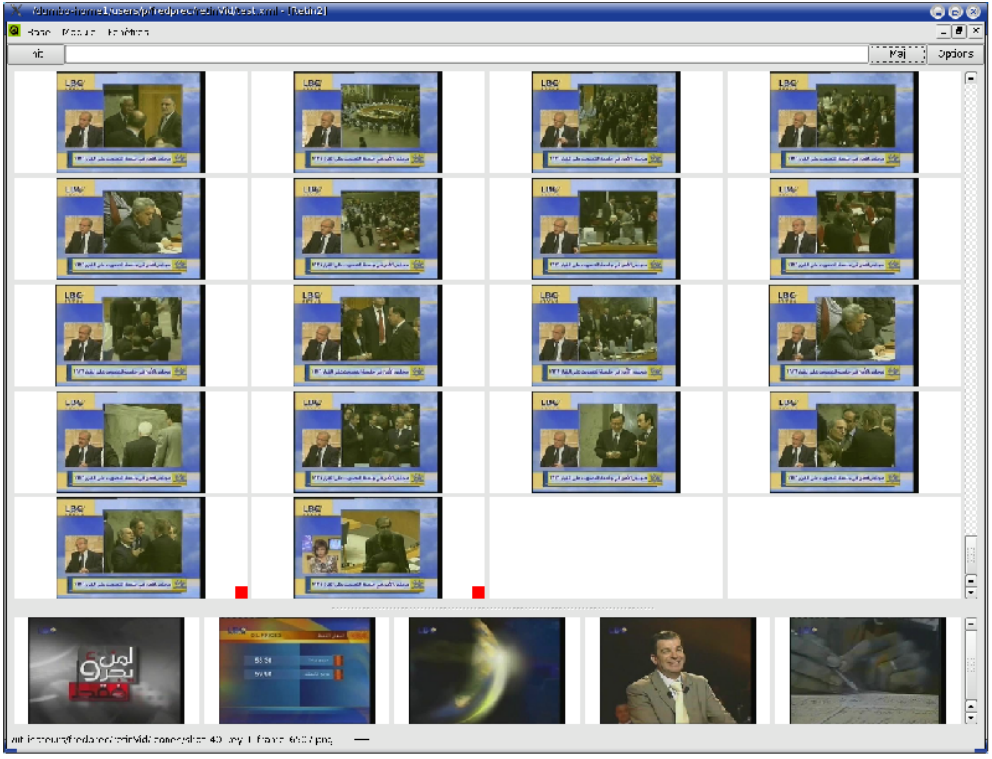


Figure 6.8: RETINVID Interface.

TRECVID-2005 data containing the 10 concepts chosen during high level feature task of the TRECVID-2005 evaluation. Results are compared through the Mean Average Precision

(MAP). We compare the MAP for our system with the average MAP of all the participants of TRECVID-2005 high level feature task in Table 6.1.

Categories	**our MAP**	**mean MAP 05**
38. People-Marching	**0.836**	0.106
39. Explosion-Fire	**0.159**	0.031
40. Maps	0.167	**0.171**
41. Flag-US	**0.168**	0.061
42. Building	0.177	**0.225**
43. Waterscape-Waterfront	**0.242**	0.165
44. Mountain	**0.151**	0.128
45. Prisoner	**0.832**	0.001
46. Sports	0.163	**0.206**
47. Car	**0.163**	0.158

Table 6.1: Comparison of the MAP for our system with average MAP of TRECVID-2005 participants for 10 official concepts chosen during 2005 evaluation.

These results are very encouraging in the context of high-level feature task and search task for our RETINVID system. We have quite comparable results with the average MAPs of TRECVID-2005 participants for 5 of the 10 features tested, better, or even far better, results for the 5 left.

6.6 Conclusion

In this Chapter, we addressed the problem of retrieving parts of videos illustrating a semantic concept, such as “Car”, “Prisioner”, etc., using only visual information. We can basically find three main steps for content-based video retrieval: temporal video segmentation, key frame extraction and video indexing.

For temporal video segmentation, we use our kernel-based SVM detector (cf. Chapters 4 and 5). Depending on the complexity of the content of the shot, one or more key frames can be extracted. For example, in the case of camera operations more than one key frame is needed, as it was explained in the motion based criteria for key frame extraction. Clustering is thus a good way to determinate both the most representative key frames, as well as their number. Thus, for key frame extraction, we explore a clustering approach.

For video indexing and retrieval, we present an interactive strategy. We have already pointed out some specific characteristics in context-based image retrieval like: high dimensionality, few training data and interactive learning. It is possible to reduce this problem through the theory of kernel functions Smola and Scholkopf [2002], specially in the case when kernel functions can be adapted to a specific application Cord et al. [2007]. We explore the characteristics of RETIN system over content-based image retrieval specificities and extend to our RETINVID system.

The Gaussian kernel gives excellent results in comparison to distance-based techniques Gosselin and Cord [2004b]. We confirm that in our content-based video retrieval system and also in our shot boundary detector. Thus, the use of this kernel associated to SVM compares key frames and computes the classification. Regarding the second characteristic, unlabeled key frames are available. Through interaction with the user it is possible for the system to acquire knowledge, i.e., the user decides whether to stop or continue with the learning process. If the user decides to continue new examples are added to the training set, improving the accuracy of the classifier. And finally, concerning the third characteristic, active learning could deal with the lack of training data. The training data is dynamic since samples take place gradually thanks to user interaction. The active learning strategy which selects for labeling new key frames close to the boundary between relevant and irrelevant key frames (RETIN's strategy) allows us to get good performance of classification with a small training set. Another advantage of active learning has to concern with the limited computation time, because user would not like to wait a long time between each feedback iteration.

Chapter 7

Conclusion and future work

Advances in multimedia technology accelerate the amount of digital information like data stored as image and video content. Both of these types of data require application-dependent processing strategies, easy-to-handle storage and indexing methods as well as sophisticated querying mechanisms. Finding methodologies to handle the temporal segmentation, storage, retrieval, searching, and browsing of digitized video data has been an active area of recent research. There are two important aspects, among many others, surrounding the development of video indexing and retrieval systems: temporal segmentation and content classification.

We present some general concluding remarks that come from the contributions described in this thesis. This thesis presented work in the areas of video segmentation, key frame selection and the use of active learning for the purpose of indexing and retrieval of video sequences.

7.1 Summary

In Chapters 1 and 2, we argued the importance of developing an automatic technique for video segmentation and content-based retrieval. Temporal video segmentation, often performed by detecting transitions between shots, is required in the early stages of video indexing and retrieval. Shots, considered as the smallest indexing unit, are not only useful for indexing, but also for summarizing the video content through key frames and to allow video browsing.

Following a review of some recent works on temporal video segmentation in Chapter 3, Chapter 4 focuses on improving existing algorithms and detecting automatically ATs instead of investigating new features in which the effect of shot is used and detected. The drawback of many well-known methods resides on the problem of fine tuning of thresholds and parameters. Some methods consider few visual features and as a consequence of this lack, these methods need pre-processing and post-processing steps. We consider AT detection from a supervised classification perspective in order to overcome threshold and parameter settings, and pre-processing and post-processing steps. Our approach is able to use multiple features simultaneously and just requires a small training. We tested different dissimilarity measures and different kernel functions in our classifier. Our system was evaluated in TRECVID-2006 on shot boundary task. Even though the performance of our AT transition detector is affected by some type of GTs, we can claim that we are among the best teams in AT detection.

In Chapter 5, we present a hierarchical system for GT detection. The first step is dedicated to detect the boundaries of ATs. Once the video sequence is segmented into cut-free segments, we seek for fade out-in transitions based on our improved method and, finally, we look for dissolves inside the shots delimited by the sharp cuts and fade out-in bounders resulting from the AT detection and fade out-in detection. The hierarchical structure of our system allows us to reduce to two modalities of identification of GTs: fast motion or dissolve. We improved an existing method that characterizes dissolves, reducing the dimension of the feature from 2D to 1D and preserving its accuracy. We also tested the performance of our system in TrecVid-2006 evaluation. Although our system detects only two types of GTs, we are among the best results. The good results are not only limited to number of transitions detected, but also in the accuracy of the interval detected, i.e., how well the interval of the gradual transition detected match with the real transition.

We can basically follow three main steps for content-based video retrieval: temporal video segmentation, key frame extraction and video indexing which were introduced in Chapter 6. A video index is much smaller and thus easier to construct and use if it references video shots instead of every video frame. One of the most common ways of representing video segments is to represent each video segment by a sequence of key frame(s). One or more key frames could be extracted, this depends on the complexity of shot's content. Camera operations and object motions are the factors that influence in the complexity of the shot content. We adopt a clustering approach for key frame extraction, since this approach is capable to extract the most representative key frames and also determine automatically their number.

Human interactive systems have attracted a lot of research interest in recent years, especially for content-based image retrieval systems. We have chosen an active learning approach because of its capacity to retrieve complex categories, specifically through the use of kernel functions. Our system is based on a content-based image retrieval machine which allows optimization of the image samples that are annotated by the user.

In this work we dealt with the following characteristics:

- high dimensionality, it is possible to reduce this problem through the theory or kernel functions;
- small training data set, unlabeled key frames are available; and
- interactive learning, through interaction with the user it is possible for the system to acquire knowledge (the user decides whether to stop or continue with the learning process).

Another advantage of active learning concerns with the limited computation time, the user would not like to wait long time between each feedback iteration.

7.2 Principal contributions

Our main contributions in temporal segmentation and video retrieval are:

Shot boundary detection

1. We proposed and implemented a hierarchical supervised approach which views temporal video segmentation as a 2-class clustering problem ("transition" and "no transition"). Our method first detects ATs using a machine learning approach. Once the video sequence is segmented into cut-free segments then they are split into GTs and normal frames. Since our objective was to develop an automatic shot boundary detector, we tried to avoid as much as possible to define thresholds and other parameters, such as sliding windows (it is necessary to define the size) as it was suggested by other author that also adopts a hierarchical approach. Our system is totally parameter free for ATs and dissolve detection. We only set one parameter for fade out-in detection.

2. Our system does not need pre-processing and post-processing steps like motion compensation and dramatic illuminance changes filtering. We decided to use the well known kernel-based SVM classifier which can deal with large feature vectors and combine a large number of visual features (color and shape) in order to avoid additional processes.

3. We used entropy as the *goodness-of-fit* measure in block-based correlation coefficients to measure the visual content similarity between frame pairs. We executed tests in AT detection and our method (entropy-based) showed better performance than maximum correlation (a method proposed by other author). The advantage of our method is that it considers the global information of the block instead of a single element of the block.

4. Our dissolve detector uses a three step process: pattern detection based on curve matching, refinement based on a modified feature for modeling error and learning step for classifying dissolve from non dissolves. We reduced the dimension of a well-known feature used for dissolve detection from 2D to 1D, preserving its capacity for dissolve detection. Indeed, we use projection histograms (1D) instead of the frame itself (2D).

5. We proposed and implemented a new method for fade detection based on the modified version of the feature developed for dissolve detection. Our method is more robust to motion changes which causes false detection due to the effects produced by motion that are similar to fade effects.

Video retrieval

We proposed and implemented an interactive video retrieval system which is based on a content-based image retrieval engine (Retin). Our system aids in the semantic labeling of video scenes. We use an active learning strategy to select new key frames for labeling that are closer to the boundary of relevant and irrelevant key frames, strategy provided by Retin system. In few iterations, the system supplies a semantic query composed by key frames ranked by relevance result.

7.3 Future work

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

Shot boundary detection

In order to improve the computation complexity, we can consider the approximation of these features using DC-Images[1] extracted from an MPEG sequence, i.e., using the compressed domain. Furthermore, it is useful to compare the current detection performance of proposed algorithms against the case when features are approximately computed from DC-Images.

A drawback of our system is on the computational complexity, since shape descriptors like moments require more time to be computed. We can use the fast computation of pseudo-Zernike moments instead of Zernike moments. Pseudo-Zernike moments have also better feature representation capabilities and are more robust to image noise than the conventional Zernike moments.

Our fade detection module is the only detector that requires to set an unique parameter. A machine learning approach for fade detection will be also very useful, keeping our primal objective to develop a system totally parameter free. We have ignored the problem of wipe detection in this thesis. We can extend the number of event detections: wipe, fade-in, fade-out and fast dissolves. We also want to improve the performance of our detectors by the interaction with the user, i.e., using active learning in all modules.

Video retrieval

Initial work on content-based retrieval focused on extracting color and texture features globally from an entire image. More recent work extended content extraction to region-based analysis where feature vectors are computed from segmented regions and similarity is evaluated between individual regions, thus we can extend our system to also compute region features.

Other characteristics that must be explored are the temporal and motion information. The temporal consistency of video data has not been well studied in the context of semantic concept detection and retrieval despite its potential value to such tasks. Temporal consistency refers to the observation that temporally adjacent video shots have similar visual and semantic content. This implies that the relevant shots matching a specific semantic concept or a query topic tend to gather in temporal neighborhoods or even appear next to each other consecutively.

[1] Reduced images formed from the collection of scaled Discrete Cosine (DC) coefficients in intra-coded discrete cosine transformation compressed video retain “global” feature.

Temporal consistency provides valuable contextual clues to video analysis and retrieval tasks. In our approach, the relevance of a given shot with respect to a semantic concept or query topic is determined based on its own content and independently from its neighboring shots. With temporal consistency, one can make more informed prediction as to the relevance of the shot by considering the relevance of its neighboring shots, thus enhancing the overall performance of the predictions.

Appendix A

Support Vector Machines

Support Vector Machines (SVMs) were introduced as a machine learning method by Cortes and Vapnik Cortes and Vapnik [1995]. The objective is that given a two-class training set they project datapoints in a higher dimensional space and attempt to specify a maximum-margin separating hyperplane between the datapoints of the two classes.

We consider SVMs in the binary classification setting. Given training data $x_1, x_2, \ldots, x_n$ that are vectors in some space $\mathcal{X} \subseteq R^d$. Also given their labels $y_1, y_2, \ldots, y_n$ where $y_i \in \{-1, 1\}$. SVM hyperplanes the training data by a maximal margin. All vectors lying on one side of the hyperplane are labeled as -1, and all vectors lying in the other side are labeled as 1. *Support vectors* are the training instances that lie the closest to the hyperplane. There exist different cases of SVM, we will review briefly some cases.

A.1 Linear separable case

Suppose we have some hyperplane which separates the positive from the negative examples. The points x which lie on the hyperplane satisfy $\mathbf{w}.x + b = 0$, where $\mathbf{w}$ is normal to the hyperplane, $\frac{|b|}{||\mathbf{w}||}$ is the perpendicular distance from the hyperplane to the origin, and $||\mathbf{w}||$ is the Euclidean norm of $\mathbf{w}$. Suppose that all the training data satisfy the following constraints:

$$x_i \cdot \mathbf{w} + b \geq +1 \text{ for } y_i = +1 \tag{A.1}$$

$$x_i \cdot \mathbf{w} + b \leq -1 \text{ for } y_i = -1 \tag{A.2}$$

which can be combined as :

$$y_i(x_i \cdot \mathbf{w} + b) \geq 0 \ \forall i \text{ (combined constraints)} \tag{A.3}$$

Consider the points for which Equation (A.1) holds. These points lie on the hyperplane $H_1 : x_i \cdot \mathbf{w} + b = 1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $\frac{|1-b|}{||w||}$. Similarly, the points for Equation (A.2) holds lie on the hyperplane $H_2 : x_i \cdot \mathbf{w} + b = -1$, with normal again $\mathbf{w}$, and perpendicular distance from the origin $\frac{|-1-b|}{||w||}$. Hence $d_+ = d_- = \frac{1}{||w||}$ and the margin is simply $\frac{2}{||w||}$. H_1 and H_2 are parallel (they have the same normal) and that no training points fall between them. Thus we can find the pair of hyperplanes which gives the maximum margin (in Figure A.1, corresponds to maximizing the distance $d_+ + d_-$) by minimizing $||w||^2$, subject to constraints (A.3).

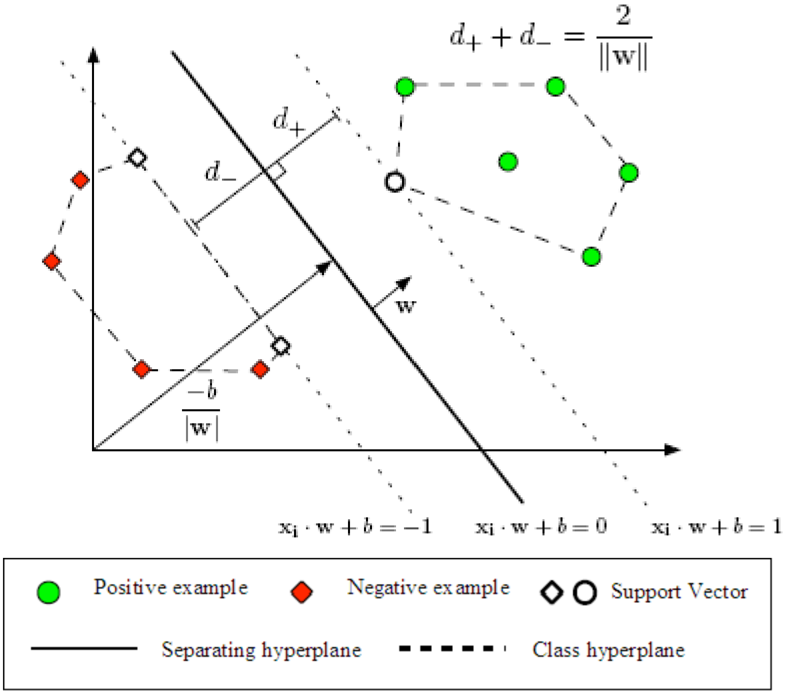

$d_+ + d_- = \frac{2}{||\mathbf{w}||}$
d_+
d_-
$\mathbf{w}$
$\frac{-b}{|\mathbf{w}|}$
$\mathbf{x_i} \cdot \mathbf{w} + b = -1$
$\mathbf{x_i} \cdot \mathbf{w} + b = 0$
$\mathbf{x_i} \cdot \mathbf{w} + b = 1$
Positive example
Negative example
Support Vector
Separating hyperplane
Class hyperplane

Figure A.1: Linearly separable classes Mueller [2004].

There are two reasons for switching to a Langrangian formulation of the problem. The first is that the constraints in Equation A.3 will be replaced by constraints on the Lagrange multipliers, which will be much easier to handle. The second is that in this formulation the training data will only appear in the form of dot products between vectors. This is a crucial property which allows generalize the procedure to the nonlinear case Burges [1998].

By applying Lagrange multipliers α_i, $i = 1, \ldots, l$ and taking the resulting dual function, we get:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{A.4}$$

subject to:

$$\alpha_i > 0. \tag{A.5}$$

$$\sum_i \alpha_i y_i = 0. \tag{A.6}$$

with solution given by :

$$\mathbf{w} = \sum_i \alpha_i y_i x_i. \tag{A.7}$$

Support vector training (for the separable, linear case) therefore amounts to maximizing the L_D with respect to the α_i. There is a Lagrange multiplier α_i for every training point. In the solution, those points for which $\alpha_i > 0$ are called "support vectors", and lie on hyperplanes H_1 and H_2.

A.2 Soft margin

Obviously, not all datasets are linearly separable, and so we need to change the formalism to account for that. Clearly, the problem lies in the constraints, which cannot always be satisfied. So, let's relax those constraints by introducing "slack variables", ζ_i. In this case, positive slack variables ζ_i, $i = 1, \ldots, l$ are added Cortes and Vapnik [1995]. For most $\mathbf{x}_i$, $\zeta_i = 0$. However, for some it effectively moves the point to the hyperplane at the edge of its class, see Figure A.2.

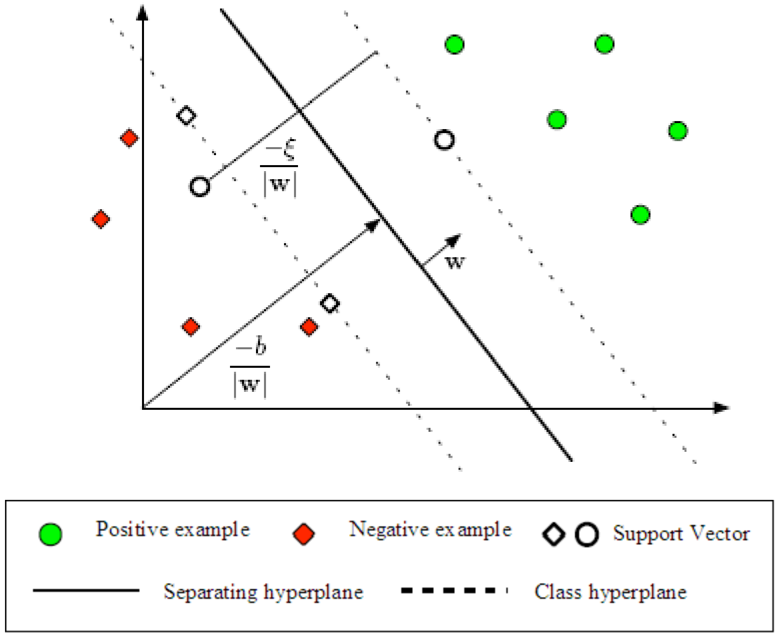


Figure A.2: Non linearly separable classes Mueller [2004].

The constraint equations are modified as follows:

$$\begin{aligned} x_i \cdot \mathbf{w} + b &\geq +1 - \zeta_i \text{ for } y_i = +1 && \text{(A.8)} \\ x_i \cdot \mathbf{w} + b &\leq -1 - \zeta_i \text{ for } y_i = -1 && \text{(A.9)} \\ \zeta_i &\geq 0 \ \forall i && \text{(A.10)} \end{aligned}$$

The purpose of the variables ζ_i is to allow misclassified points, which have their corresponding $\zeta_i > 1$. Therefore $\sum \zeta_i$ is an upper bound on the number of training errors. Hence a natural way to assign an extra cost for errors is to change the objective function to be minimized from $\frac{||w||^2}{2}$ to $\frac{|||w||}{2} + C(\sum_i \zeta_i)^k$, The term $C \sum_i \zeta_i$ leads to a more robust solution, in the statistical sense, i.e., this term makes the optimal separating hyperplane less sensitive to the presence of outliers in the training set. C is a parameter to be chosen by the user, a larger C corresponding to assigning a higher penalty to errors. This is a convex programming problem for any positive integer k; for $k = 2$ and $k = 1$ it is also a quadratic programming problem, and the choice $k = 1$ has the further advantage that neither the ζ_i, nor their Lagrange multipliers, appear in the dual function, which becomes:

Maximize:

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{A.11}$$

subject to:

$$0 \leq \alpha_i \leq C, \tag{A.12}$$

$$\sum_i \alpha_i y_i = 0. \tag{A.13}$$

The solution is again given by

$$\mathbf{w} = \sum_i^{N_s} \alpha_i y_i x_i. \tag{A.14}$$

where N_s is the number of support vectors. Thus the only difference from the optimal hyperplane case is that the α_i now have an upper bound C.

A.3 Nonlinear SVMs

In most cases, linear separation in input spaces is a too restrictive hypothesis to be of practical use. Fortunately, the theory can be extended to nonlinear separating surfaces by mapping the input points into features points and looking for optimal hyperplane in the corresponding feature space Cortes and Vapnik [1995].

In order to use higher-level functions to classify data using SVMs, the data is first mapped to a higher-order feature space, possibly of infinite dimension (see Figure A.3):

$$\Phi : R^d \mapsto \mathcal{H} \tag{A.15}$$

Because the operations on the x_i are always dot products, a *kernel function* K can be used to perform the mapping

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j). \tag{A.16}$$

It will only be necessary to use K in the training algorithm, and would never need to explicitly even know what Φ is. Thus, the SVM equation becomes:

$$f(x) \quad = \quad \sum_{i=1}^{N_s} \alpha_i y_i \Phi(s_i) \cdot \Phi(x) + b \tag{A.17}$$

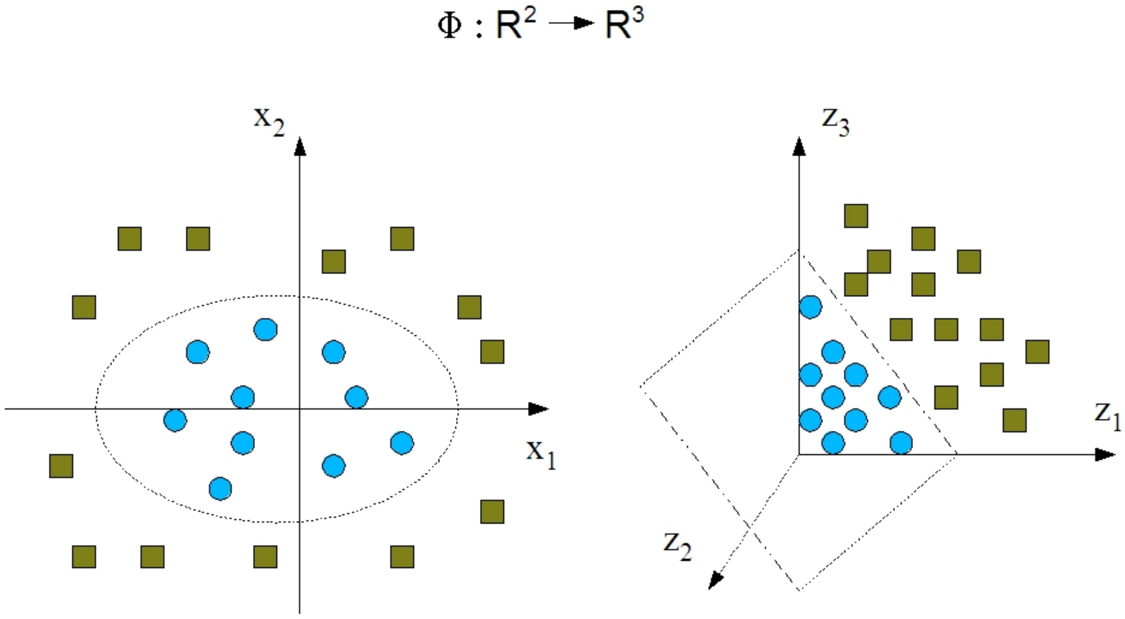


Figure A.3: Input data mapped to a higher-order feature space Mueller [2004].

$$= \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b \tag{A.18}$$

where s_i are the support vectors, subject to:

$$0 \leq \alpha_i \leq C, \tag{A.19}$$

$$\sum_{i}^{N_s} \alpha_i y_i = 0. \tag{A.20}$$

The solution is again given by

$$\mathbf{w} = \sum_{i}^{N_s} \alpha_i y_i \Phi(x_i). \tag{A.21}$$

where N_s is the number of support vectors.

Several common kernel functions are used to map data into higher dimension feature space:

Linear

$$K(x_i, x_j) = x_i \cdot x_j \tag{A.22}$$

Polynomial kernel:

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{A.23}$$

Gaussian radial basis kernel :

$$K(x_i, x_j) = e^{-||x_i - x_j||^2/2\sigma^2} \tag{A.24}$$

Gaussian kernel with χ^2 distance (Gauss-χ^2):

$$K(x_i, x_j) = e^{-\chi^2(x_i,x_j)/2\sigma^2} \tag{A.25}$$

Triangular kernelFleuret and Sahbi [2003]:

$$K(d_t, d_s) = -||d_t - d_s|| \tag{A.26}$$

Each kernel function results in a different type of decision boundary.

Figure A.4 shows classes that are separable by a polynomial shaped surface in the input space, rather than a hyperplane.

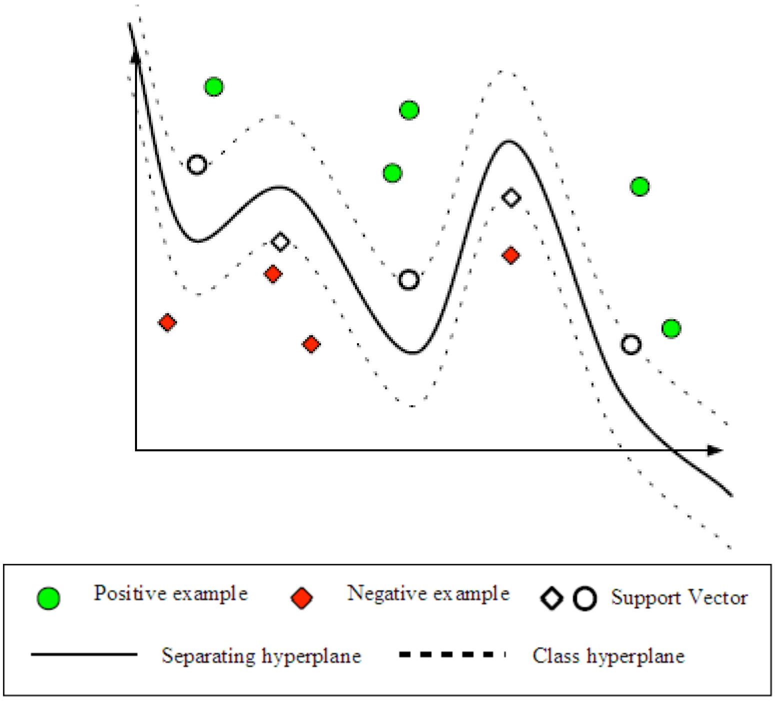


Figure A.4: Nonlinear classes Mueller [2004].

There are many possible kernels, and the most popular ones are given above. All of them should fulfill the so-called Mercer's conditions. The Mercer's kernels belong to a set of reproducing kernels.

A.3.1 Mercer condition

There exists a mapping Φ and an expansion

$$K(x, y) = \sum_i \Phi(x)_i \Phi(y)_i \tag{A.27}$$

if and only if, for any $g(x)$ such that

$$\int g(x)^2 dx \text{ is finite} \tag{A.28}$$

then

$$\int K(x,y)g(x)g(y)dxdy \geq 0. \tag{A.29}$$

Mercer's condition tell us whether or not a prospective kernel is actually a dot product in some space. The theory of Mercer Kernels allows data which may be embedded in a vector space, such as spectral lines, physical measurements, stock market indices, or may not arise from a vector space, such as sequences, graphs, and trees to be treated using similar mathematics.

Bibliography

W. H. Adams, G. Iyengar, C.-Y. Lin, M. Naphade, C. Neti, H. Nock, and J. Smith. Semantic indexing of multimedia content using visual, audio, and text cues. *EURASIP J. Appl. Signal Process.*, 2:170–185, 2003.

G. Ahanger and T.D.C. Little. Data semantics for improving retrieval performance of digital news video systems. *IEEE Trans. on Knowledge and Data Engineering*, 13:352–360, 2001.

A. Akutsa, Y. Tonomura, H. Hashimoto, and Y. Ohba. Video indexing using motion vectors. In *SPIE Visual Communication and Image Processing*, volume 1818, pages 1522–1530, 1992.

Faruq A. Al-Omari and Mohammad A. Al-Jarrah. Query by image and video content: a colored-based stochastic model approach. *Data Knowl. Eng.*, 52(3):313–332, 2005.

A.M. Alattar. Detecting and compressing dissolve regions in video sequences with a dvi multimedia image compression algorithm. *IEEE International Symposium on Circuits and Systems (ISCAS)*, 1:13–16, 1993.

Sameer Antani, Rangachar Kasturi, and Ramesh Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35:945–965, 2002.

E. Ardizzone, G. Gioiello, M. LaCascia, and D. Molinelli. A real-time neural approach to scene cut detection. In *Proc. of IS & T/SPIE - Storage & Retrieval for Image and Video Databases IV*, 1996.

N. Babaguchi, Y. Kawai, and T. Kitahashi. Event based indexing of broacasted sports video by intermodal collaboration. *IEEE Trans. on Multimedia*, 4:68–75, 2002.

F. N. Bezerra and E. Lima. Low cost soccer video summaries based on visual rhythm. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 71–78, New York, NY, USA, 2006. ACM Press.

Francisco Nivando Bezerra. A longest common subsequence approach to detect cut and wipe video transitions. In *SIBGRAPI '04: Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium on (SIBGRAPI'04)*, pages 154–160, Washington, DC, USA, 2004. IEEE Computer Society.

Francisco Nivando Bezerra and Neucimar Jerônimo Leite. Using string matching to detect video transitions. *Pattern Analysis & Application*, 10(1):45–54, Feb. 2007.

J. Boreczky and L. Rowe. Comparison of video shot boundary detection techniques. In *Conf. on Storage and Retrieval for Image and Video Databases (SPIE)*, pages 170–179, San Jose, Febrary 1996. Avaliable on `citeseer.ist.psu.edu/boreczky96comparison.html`.

B.E. Boser, I. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc of the 5th Annual Workshop on Computational Learning Theory*, volume 5, pages 144–152, 1992.

S. Boughorbel, J.-P. Tarel, and F. Fleuret. Non-mercer kernels for svm object recognition. In *Proceedings of British Machine Vision Conference (BMVC'04)*, pages 137 – 146, London, England, 2004.

P. Bouthemy, M. Gelgon, and F. Ganansia. A unified approach to shot change detection and camera motion characterization. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1030–1044, 1999.

L. Breiman. Bagging predictor. *Machine Learning*, 24(2):123–140, 1996.

Klaus Brinker. Active learning with kernel machines. Master's thesis, Faculty of Electrical Engineering, Computer Science and Mathematics. University of Paderbron, 2004.

R. Brunelli and O. Mich. Histograms analysis for image retrieval. *Pattern Recognition*, 34(8): 1625–1637, 2001.

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

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

X. Cabedo and S. Bhattacharjee. Shot detection tools in digital video. In *Proc. of Non-linear Model Based Image Analysis 1998*, pages 121–126, Glasgow, July 1998. Springer Verlag.

C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proc. of the Seventeenth International Conference on Machine Learning*, pages 111–118, 2000.

P. Campisi, A. Neri, and L. Sorgi. Automatic dissolve and fade detection for video sequences. In *14th International Conference on Digital Signal Processing, 2002. DSP 2002*, volume 2, pages 567–570, 2003.

J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

Jie Cao, Yanxiang Lan, Jianmin Li, Qiang Li, Xirong Li, Fuzong Lin, Xiaobing Liu, Linjie Luo, Wanli Peng, Dong Wang, Huiyi Wang, Zhikun Wang, Zhen Xiang, Jinhui Yuan, Wujie Zheng, Bo Zhang, Jun Zhang, Leigang Zhang, and Xiao Zhang. Intelligent multimedia group of tsinghua university at trecvid 2006. In *TREC Video Retrieval Evaluation Online Proceedings*, 2006.

Z. Cernekova, I. Pitas, and C. Nikou. Information theory-based shot cut/fade detection and video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(1):82–91, Jan. 2006.

E.Y. Chang, S. Tong, K.-S. Goh, and C.-W. Chang. Support vector machine concept-dependent active learning for image retrieval. *IEEE Transactions on Multimedia*, 2005. accepted.

M.G. Chang, H. Kim, and S.M.-H. Song. A scene change boundary detection method. In *Proc. Int. Conf. Image Processing*, volume 3, pages 933–936, 2000.

O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram based image classification. *IEEE Trans. on Neural Networks*, 10(5):1055–1064, 1999.

M. Chen, Michael Christel, Alexander Hauptmann, and Howard Wactlar. Putting active learning into multimedia applications: dynamic definition and refinement of concept classifiers. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 902–911, New York, NY, USA, 2005. ACM Press.

Y. Chen, X. Zhou, and T. Huang. One-class svm for learning in image retrieval. In *International Conference on Image Processing*, volume 1, pages 34–37, 2001.

Tat-Seng Chua, HuaMin Feng, and A. Chandrashekhara. An unified framework for shot boundary detection via active learning. In *Proc IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2003.*, volume 2, pages 845–848, 2003.

G. Cámara-Chávez, M. Cord, F. Precioso, S. Philipp-Foliguet, and Arnaldo de A. Araújo. Robust scene cut detection by supervised learning. In *European Signal Processing Conference (EUSIPCO'06)*, 2006a.

G. Cámara-Chávez, F. Precioso, M. Cord, S. Philipp-Foliguet, and Arnaldo de A. Araújo. Shot boundary detection at trecvid 2006. In *TREC Video Retrieval Evaluation Online Proceedings*, 2006b.

G. Cámara-Chávez, F. Precioso, M. Cord, S. Philipp-Foliguet, and A. de A. Araújo. Shot boundary detection by a hierarchical supervised approach. In *14th Int. Conf. on Systems, Signals and Image Processing (IWSSIP'07)*, Jun. 2007. accepted for publication.

D.A. Cohn, Z. Ghahramani, and M.I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

M. Cooper. Video segmentation combining similarity analisys and classification. In *Proc. of the 12th annual ACM international conference on Multimedia (MULTIMEDIA '04)*, pages 252–255, 2004.

M. Cooper and J. Foote. Scene boundary detection via video self-similarity analysis. In *Proc. IEEE Int. Conf. on Image Processing (ICIP '01)*, 2001.

Matthew Cooper, John Adcock, Robert Chen, and Hanning Zhou. Fxpal experiments for trecvid 2005. In *TREC Video Retrieval Evaluation Online Proceedings*, volume 3, pages 378–381, 2005.

Matthew Cooper, John Adcock, and Francine Chen. Fxpal at trecvid 2006. In *TREC Video Retrieval Evaluation Online Proceedings*, 2006.

M. Cord, P.-H. Gosselin, and S. Philipp-Foliguet. Stochastic exploration and active learning for image retrieval. *Image and Vision Computing*, 25:14–23, 2007.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Costas Cotsaces, Nikos Nikolaidis, and Loannis Pitas. Video shot detection and condensed representation: A review. *IEEE Signal Processing Magazine*, 23(2):28–37, 2006.

I. Cox, M. Miller, T. Minka, T Papathomas, and Y. Yianilos. The bayesian image retrieval system. pichunter: Theory, implementation and psychophysical experiments. *IEEE Trans. on Image Processing*, 9(1):20–37, 2000.

S. Dagtas, W. A. khatib, A. Ghafoor, and R. L. Kashyap. Models for motion-based video indexing and retreival. *IEEE Trans. on Image Processing*, 9(1):88–101, Jan 2000.

A. Dailianas, R.B. Allen, and P. England. Comparison of automatic video segmentation algorithms. In *SPIE Photonics West*, volume 2615, pages 2–16, Philadelphia, October 1995. Avaliable on `citeseer.ist.psu.edu/dailianas95comparison.html`.

A. del Bimbo. *Visual Information Retrieval.* Morgan Kaufmann, San Francisco, California, 1999.

Claire Demarty and Serge Beucher. Morphological tools for indexing video documents. *IEEE International Conference on Multimedia Computing and Systems (ICMCS'02)*, 2:991–992, 1999.

Y. Deng and B. S. Manjunath. Netra-v: Toward and object-based video representation. *IEEE Trans. Circuits Syst. Video Technol.*, 8:616–627, 1998.

D. Diklic, D. Petkovic, and R. Danielson. Automatic extraction of representative key-frames based on scene content. In *Conference Record of the Asilomar Conference on Signals, Systems and Computers.*, volume 1, pages 877–881, 1998.

A. Divakaran, R. Radhakrishnan, and K.A. Peker. Motion activity-based extraction of key-frames from video shots. In *International Conference on Image Processing*, volume 1, pages I:932–I:935, 2002.

C. Djeraba. Content-based multimedia indexing and retrieval. *Multimedia IEEE*, 9(2):18–22, 2002.

J.P. Eakins. Toward intelligent image retrieval. *Pattern Recognition*, 35(1):3–14, 2002.

A.K. Elgmagarmid, H. Jiang, A.A. Helal, A. Joshi, and M. Admed. *Video Satabase Systems: Issues, Products, and Applications.* Kluwer Academic Publishers, Boston, 1997.

Ralph Ewerth and Bernd Freisleben. Video cut detection without thresholds. In *Proc. of 11th Workshop on Signals, Systems and Image Processing*, pages 227–230, Poznan, Poland, 2004. PTETiS.

Ralph Ewerth, Markus Mühling, Thilo Stadelmann, Ermir Qeli, Björn Agel, Dominik Seiler, and Bernd Freisleben. University of marburg at trecvid 2006: Shot boundary detection and rushes task results. In *TREC Video Retrieval Evaluation Online Proceedings*, 2006.

J. Fan, A. Elmagarmid, X. Zhu, W. Aref, and L. Wu. Classview: Hierarchical video shot classification, indexing, and accessing. *IEEE Trans. Multimedia*, 6(1):70–86, 2004.

D. Feng, W.C. Siu, and H. Zhang. *Multimedia Information Retrieval and Management.* Springer-Verlag, Berlin Heidelberg, 2003.

Huamin Feng, Wei Fang, Sen Liu, and Yong Fang. A new general framework for shot boundary detection and key-frame extraction. In *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 121–126, New York, NY, USA, 2005.

A.M. Ferman and A.M. Tekalp. Efficient filtering and clustering methods for temporal segmentation and visual summarization. *Journal of Visual Communication and Image Representation*, 9(5):336–351, 1998.

A.M. Ferman, A.M. Takalp, and R. Mehrotra. Effective content representation for video. In *International Conference on Image Processing (ICIP' 98)*, volume 3, pages 521–525, 1998.

W.A.C Fernando, C.N. Canagarajah, and D. R. Bull. Video segmentation and classification for content based storage and retrieval using motion vectors. In *Proceeding of the SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 687–698, 1999.

W.A.C. Fernando, C.N. Canagarajah, and D.R. Bull. A unified approach to scene change detection in uncompressed and compressed video. *IEEE Trans. on Consumer Electronics*, 46(3):769–779, 2000.

F. Fleuret and H. Sahbi. Scale-invariance of support vector machines based on the triangular kernel. In *3th International Workshop on Statistical and Computational Theories of Vision (part of ICCV'03)*, Nice, France, 2003.

M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, M. Gorkani B. Dom, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *IEEE Comput.*, 28(9):23–32, 1995.

J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer graphics: principles and practice.* Addison Wesley, 2nd edition, 1990.

R.M. Ford, C. Robson, D. Temple, and M. Gerlach. Metrics for scene change detection in digital video sequences. In *IEEE International Conference on Multimedia Computing and Systems '97*, pages 610–611, 3-6 June 1997.

J. Fournier, M. Cord, and S. Philipp-Foliguet. Retin: A content-based image indexing and retrieval system. *Pattern Analysis and Applications Journal, Special issue on image indexation*, 4(2/3):153–173, 2001.

Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

K. Fukumizu. Statistical active learning in multilayer perceptrons. *IEEE Trans. Neural Networks*, 11(1):17–26, Jan. 2000.

B. Furht, S.W. Smoliar, and H.J. Zhang. *Video and image processing in multimedia systems.* Kluwer Academic Publishers, 1995.

D. B. Rosen G. Carpenter, S. Grossberg. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Network*, 4(6):759–771, 1991.

S. Grossberg G. Carpenter. Art2: Self-organizing of stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930, 1987.

X. Gao and X. Tang. Unsupervised video-shot segmentation and model-free achorperson detection for news video story parsing. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(9):765–776, 2002.

U. Gargi, S. Oswald, D. Kosiba, S. Devadiga, and R. Kasturi. Evaluation of video sequence indexing and hierarchical video indexing. In *Proc. of SPIE Conf. on Storage and Retrieval in Image and Video Databases*, pages 1522–1530, 1995.

U. Gargi, R. Kasturi, and S.H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(1):1–13, 2000.

P.-H. Gosselin and M. Cord. Precision-oriented active selection for interactive image retrieval. In *International Conference on Image Processing (ICIP'06)*, pages 3127–3200, October 2006.

P.-H. Gosselin and M. Cord. Retin al: an active learning strategy for image category retrieval. In *International Conference on Image Processing (ICIP'04)*, volume 4, pages 2219–2222, Oct. 2004a.

P.H. Gosselin and M. Cord. A comparison of active classification methods for content-based image retrieval. In *CVDB '04: Proceedings of the 1st international workshop on Computer vision meets databases*, pages 51–58, Paris, France, June 2004b.

P.H. Gosselin and M. Cord. Active learning techniques for user interactive systems: application to image retrieval. In *Int. Workshop on Machine Learning techniques for processing MultiMedia content*, Bonn, Germany, Aug. 2005.

P.O. Gresle and T.S. Huang. Gisting of video documents: A key frame selection algorithm using relative activity measure. In *The 2nd Int. Conf. on Visual Information Systems*, pages 279–286, 1997.

Silvio Jamil Ferzoli Guimarães, Michel Couprie, Arnaldo de Albuquerque Araújo, and Neucimar Jerónimo Leite. Video segmentation based on 2d image analysis. *Pattern Recogn. Lett.*, 24(7):947–957, 2003.

S.J.F. Guimarães, N.J. Leite, M. Couprie, and A. de A. Araújo. Flat zone analysis and a sharpening operation for gradual transition detection on video images. *EURASIP Journal on Applied Signal Processing*, 2004(12):1943–1953, 2004.

B. Gunsel, A. Fernan, and A. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, pages 592–604, 1998.

A. Gupta and R. Jain. Visual information retrieval. *Commun. ACM*, 40(5):70–79, 1997.

N. Haering, N. da Vitoria Lobo, R. Qian, and I. Sezan. A framework for designing event detectors. In *Fourth Asian Conference on Computer Vision*, Taipe, Taiwan, 2000.

A. Hampapur, R. Jain, and T. Weymouth. Digital video segmentation. In *ACM Multimedia 94 Proceedings*, pages 357–364, November 21-24 1994.

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

Seung-Hoon Han and In So Kweon. Detecting cuts and dissolves through linear regression analysis. *Electronics Letters*, 39(22):1579–1581, 2003.

Alan Hanjalic. Shot boundary detection: Unraveled and resolved? *IEEE Trans. on Circuits and System for Video Technology*, 12(2):90–105, 2002.

Alexander G. Hauptmann and Michael G. Christel. Successful approaches in the trec video retrieval evaluations. In *Proc. of ACM Multimedia*, pages 668–675, New York, Oct 10-16 2004. ACM Press.

Alexander G. Hauptmann, Wei-Hao Lin, Rong Yan, Jun Yang, and Ming-Yu Chen. Extreme video retrieval: joint maximization of human and computer performance. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394, New York, NY, USA, 2006. ACM Press.

W. J. Heng and K. N. Ngan. High accuracy flashlight scene determination for shot boundary detection. *Signal Processing: Image Communication*, 18(3):203–219, Mar. 2003.

Wei Jyh Heng and King Ngi Ngan. Integrated shot boundary detection using object-based technique. In *Proc. IEEE Int. Conference on Image Processing*, volume 3, pages 289–293, 1999.

B.K.P. Horn and B.G. Schunck. Determinating optical flow. *Artificial Intelligence*, 17:185–203, 1981.

A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

Omar Javed, Sohaib Khan, Zeeshan Rasheed, and Mubarak Shah. A framework for segmentation of interview videos. In *IASTED: International Conference on Internet and Multimedia Systems and Applications*, November 2000. Available on citeseer.ist.psu.edu/372719.html.

F. Jing, M. Li, H.-J. Zhang, and B. Zhang. An efficient and effective region-based image retrieval framework. *IEEE Trans. on Image Processing*, 13(4):699–709, May 2004.

R.A. Joyce and B. Liu. Temporal segmentation of video using frame and histogram-space. *IEEE Trans. on Multimedia*, 8(1):130–140, Feb. 2006.

S.-C Jun and S.-H. Park. An automatic cut detection algorithm using median filter and neural network. In *Proc. Int. Technical Conference on Circuits/Systems, Computers and Communications*, pages 1049–1052, Jul. 2000.

C. Kan and M.D. Srinath. Combined features of cubic b-spline wavelet moments and zernike moments for invariant pattern recognition. In *International Conference on Information Technology: Coding and Computing.*, pages 511–515, 2001.

C. Kan and M.D. Srinath. Invariant character recognition with zernike and orthogonal fourier-mellin moments. *Pattern Recogntion*, 35(1):143–154, 2002.

S. H. Kim and R.-H. Park. Robust video indexing for video sequences with complex brightness variations. In *Proc. IASTED Int. Conf. Signal Image Process*, pages 410–414, 2002.

Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, 2001. Elsevier Science.

F.W. Lancaster. *Vocabulary Control for Information Retrieval.* Information Resources Press, Arlington, Virginia, USA, 1986.

Man-Hee Lee, Hun-Woo Yoo, and Dong-Sik Jang. Video scene change detection using neural network: Improved art2. *Expert Systems with Applications*, 31(1):13–25, 2006.

R.M. Leonardi. Semantic indexing of multimedia documents. *IEEE Multimedia*, 9:44–51, 2002.

D. Lewis and J. Catlett. A sequencial algorithm for training text classifiers. In Springer-Verlag, editor, *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994a.

D.D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proc 11th Int'l Conf. Machine Learning*, pages 148–156, 1994b.

Mingkun Li and Ishwar K. Sethi. Confidence-based active learning. *IEEE Trans. on Pattern Analysys and Machine Intelligence*, 28(8):1251–1261, 2006.

R. Lienhart. Reliable dissolve detection. In *Proc SPIE Storage Retrieval for Media Database*, volume 4315, pages 219–230, 2001a.

R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *IJIG*, 1(3):469 – 486, 2001b.

R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *Proc. SPIE. Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 290–301, December 1999.

R. Lienhart, C. Kuhmunch, and W. Effelsberg. On the detection and recognition of television commercials. *IEEE Int. Conf. on Multimedia Computing and Systems (ICMC '97)*, pages 509–516, 1997a.

R. Lienhart, S. Pfeiffer, and W. Effelsberg. Video abstracting. *Communications of the ACM*, 40(12):54–62, Dec 1997b.

Y Lin, M. S. Kankanhalli, and T.-S. Chua. Temporal multiresolution analysis for video segmentation. In *Proc. SPIE Storage Retrieval Media Database VIII*, volume 3972, pages 494–505, 2000.

Jian Ling, Yi-Qun Lian, and Yue-Ting Zhuang. A new method for shot gradual transiton detection using support vector machine. In *Proc. International Conference on Machine Learning and Cybernetics, 2005*, volume 9, pages 5599–5604, 1998.

T.-Y. Liu, J. Feng, X.-D. Zhang, and K.-T. Lo. Inertia-based video cut detection and its integration with video coder. In *IEE Proceedings - Vision, Image, and Signal Processing*, volume 150, pages 186–192, 2003.

G.J. Lu. *Multimedia database management systems.* Artech House Publishers, London, 1999.

H.B. Lu, Y.J. Zhang, and Y.R. Yao. Robust gradual scene change detection. In *International Conference on Image Processing (ICIP' 99)*, volume 3, pages 304–308, 1999.

Ye Lu, Chunhui Hu, Xingquan Zhu, HongJiang Zhang, and Qiang Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *The eighth ACM international conference on Multimedia*, pages 31–37, 2000.

Z. Lui, Y. Wang, and T. Chen. Audio feature extraction and analysis for scene segmentation and classification. *J. VLSI Signal Processing Syst. Signal Image, Video Tech.*, 20:61–79, Oct. 1998.

G. Lupatini, C. Saraceno, and R. Leonardi. Scene break detection: a comparison. In *8th Int. Workshop on Research Issues in Data Engineering*, pages 34–41, 1998.

Petros Maragos. Work package 6: Cross-modal integration for performance improving in multimedia. report onthe state-of-the-art. Technical report, MUSCLE Network of Excellence, Greece, 2004.

Jordi Mas and Gabriel Fernandez. Video shot boundary detection based on color histograms. In *TREC Video Retrieval Evaluation Online Proceedings*, 2003.

Kazunori Matsumoto, Masaki Naito, Keiichito Hoashi, and Fumiaki Sugaya. Svm-based shot boundary detection with a novel feature. In *Proc. IEEE Internatinal Conference on Multimedia and Expo (ICME'06)*, pages 1837–1840, 2006.

M. Miyahara and Y. Yoshida. Mathematical transform of (rgb) color data to munsell (hvc) color data. In *Proc of SPIE Visual Communications and Image Processing*, volume 1001, pages 650–657, 1988.

Chris Mueller. Support vector machines, 2004. Avaliable on `http://www.osl.iu.edu/~chemuell/classes/b659/svm.pdf`.

Umut Naci and Alan Hanjalic. Tu delft at trecvid 2005: Shot boundary detection. In *TREC Video Retrieval Evaluation Online Proceedings*, 2005.

A. Nagasaka and Y. Tanaka. *Automatic video indexing and full video search for object appearances.* E. Knuth and L.M. Wegner (eds), Elsevier, 1992.

J. Nam and A.H. Tewfik. Detection of gradual transitions in video sequences using b-spline interpolation. *IEEE Transactions on Multimedia*, 7:667–679, 2005.

J. Nam and A.H. Tewfik. Combined audio and visual streams analysis for video sequence segmentation. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2665–2668, 1997.

M. Naphade and T. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Trans. Multimedia*, 3(1):141–151, 2001.

R. Narasimnha, A. Savakis, R.M. Rao, and R. de Queiroz. Key frame extraction using mpef-7 motion descriptors. In *Conference on Signals, Systems and Computers, 2003*, volume 2, pages 1575–1579, 2003.

G. Navarro. A guided tour to approximate string matching. *ACM Comp Surveys*, 33(1): 31–88, 2001.

C. W. Ngo, T. C. Pong, and R. T. Chin. Detection of gradual transitions through temporal slice analysis. In *IEEE Proc. of Computer Vision and Pattern Recognition (CVPR '99)*, pages 36–41, 1999.

Chong-Wah Ngo. A robust dissolve detector by support vector machine. In *Proc of the eleventh ACM international conference on Multimedia*, pages 283–286, 2003.

Chong-Wah Ngo, Zailiang Pan, Xiaoyong Wei, Xiao Wu, and Hung-Khoon Tan. Motion driven approaches to shot boundary detection, low-level feature extraction and bbc rush characterization. In *TREC Video Retrieval Evaluation Online Proceedings*, 2005.

C.W. Ngo, T.-C. Pong, and R.T. Chin. Video parsing by temporal slice coherency. *IEEE Trans. Circuits Syst. Video Technol.*, 11(8):941–953, 2001.

J. Ortega-Garcia, J. Gonzalez-Rodriguez, D. Simon-Zorita, and S. Cruz-Llanas. *Biometrics Solutions for Authentication in an E-World*, chapter From Biometrics Technology to Applications regarding Face, Voice, Signature and Fingerprint Recognition Systems, pages 289–337. Kluwer Academic Publisher, 2002. ed. D. Zhang.

Mihai Osian and Luc Van Gool. Video shot characterization. *Mach. Vision Appl.*, 15(3): 172–177, 2004.

C. O'Toole. An mpeg-1 shot boundary detector using xil colour histograms. Technical Report 98-04, Centre for Digital Video Processing, Dublin City University, 1998.

K. Otsuji and Y. Tonomura. Projection detecting filter for video cut detection. In *ACM Multimedia '93 Proceedings*, pages 271–257, 1993.

T.N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Trans. on Signal Processing*, pages 901–914, April 1992.

Greg Pass and Ramin Zabih. Comparing images using joint histograms. *Multimedia Systems*, 7(3):234–240, 1999.

G. Patanè and M. Russo. The enhancement lbg algorithm. *IEEE Trans. on Neural Networks*, 14(9):1219–1237, 2001.

A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *Int. J. Comput. Vis.*, 18(3):233–254, 1996.

S. Philipp-Foliguet, G. Logerot, P. Constant, PH. Gosselin, and C. Lahanier. Multimedia indexing and fast retrieval based on a vote system. In *International Conference on Multimedia and Expo*, pages 1782–1784, Toronto, Canada, July 2006.

S. V. Porter, M. Mirmehdi, and B. T. Thomas. Temporal video segmentation and classification of edit effects. *Image and Vision Computing*, 21(13-14):1097–1106, December 2003.

W.K. Pratt. *Digital Image Processing*. John Wiley & Sons, 1991.

W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988-1992.

Guo-Jun Qi, Yan Song, Xian-Sheng Hua, Hong-Jiang Zhang, and Li-Rong Dai. Video annotation by active learning and cluster tuning. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '06)*, page 114, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2646-2.

Y. Qi, T. Liu, and A. Hauptmann. Supervised classification of video shot segmentation. In *Proc. Int. Conf. on Multimedia and Expo (ICME '03)*, volume 2, pages 689–692, Baltimore, MD, July 6-9 2003.

Xueming Qian, Guizhong Liu, and Rui Su. Effective fades and flashlight detection based on accumulating histogram difference. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(10):1245–1258, 2006.

W. Ren, M. Singh, and S. Singh. Automated video segmentation. In *Proc. 3rd International Conference on Information, Communications & Signal Processing (ICICS '01)*, Singapore, Oct. 2001.

Oscar Robles, Pablo Toharia, Angel Rodriguez, and Luis Pastor. Using adaptive thresholds for automatic video cut detection. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004.

Y. Rui and T Huang. Optimizing learning in image retrieval. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 236–243, June 2000a.

Y. Rui and T. Huang. A unified framework for video browsing and retrieval. In A. Bovik, editor, *Image and Video Processing Handbook*, pages 705–715, New York, 2000b.

Y. Rui, S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems fr Video Technology*, 8:644–655, 1998.

E. Sahouria and A. Zakhor. Content analysis of video using principal components. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(8):1290–1298, 1999.

S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Trans. on Knowledge and Data Engineering*, 13(3):337–351, 2001.

Thiago Teixeira Santos. Shot-boundary detection on video. Master's thesis, Institute of Mathematics and Statistics (IME), University of São Paulo, 2004.

S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in news videos. *IEEE Multimedia*, 6:22–35, 1999.

G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*, pages 839–846, 2000.

B. Shahraray. Scene change detection and content-based sampling of video sequences. In *SPIE Conference on Digital Video Compression*, volume 2419, pages 2–13, 1995.

B. Shahraray and D.C. Gibbon. Pictorial transcripts: Multimedia processing applied to digital library creation. In *IEEE 1st Multimedia Signal Processing Workshop*, pages 581–586, June 1997.

Y. Sheng and L. Shen. Orthogonal fourier-mellin moments for invariant pattern recognition. *J. Opt. Soc. Am.*, 11:1748–1757, 1994.

A.F. Smeaton and P. Over. The trec-2002 video track report. In *The Eleventh Text Retrieval Conference (TREC 2002)*, 2002. `http://trec.nist.gov//pubs/trec11/papers/VIDEO.OVER.pdf`.

A.F. Smeaton and P. Over. Trecvid 2006: Shot boundary detection task overview. In *TREC Video Retrieval Evaluation Online Proceedings*, 2006. `http://www-nlpir.nist.gov/projects/tvpubs/tv6.papers/tv6.sb.slides-final.pdf`.

A.F. Smeaton, C. Foley, C. Gurrin, Hyowon Lee, and S. McGivney. Collaborative searching for video using the fischlar system and a diamondtouch table. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'06)*, 2006.

A. Smeulders, M.Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

A. Smola and B. Scholkopf. *Learning with kernels*. MIT Press, Cambridge, MA., 2002.

C. Snoek, J. v. Gemert, Th. Gevers, B. Huurnink, D. Koelma, M van Liempt, O. d. Rooij, K.E.A van de Sande, F.J Seinstra, A. Smeulders, A.H.C. Thean, C.J. Veenman, and M. Worring. The mediamill trecvid 2006 semantic video search engine. In *TREC Video Retrieval Evaluation Online Proceedings*, Gaithersburg, MD, 2006a.

C. Snoek, M. Worring, J. Geusebroek, D. Koelma, F. Seinstra, and A. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1678–1689, Oct. 2006b.

Cees G. M. Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25:5–35, 2005.

Cees G.M. Snoek, Marcel Worring, Jan van Gemert, Jan-Mark Geusebroek, Dennis Koelma, Giang P. Nguyen, Ork de Rooij, and Frank Seinstra. Mediamill: Exploring news video archives based on learned semantics. In *Proceedings of ACM Multimedia*, November 2005.

Yan Song, Guo-Jun Qi, Xian-Sheng Hua, Li-Rong Dai, and Ren-Hua Wang. Video annotation by active learning and semi-supervised ensembling. In *IEEE International Conference on Multimedia and Expo (ICME '06)*, pages 933–936, 2006.

M.J. Swain. Interactive indexing into image databases. In *Proc of SPIE Conference on Storage and Retrieval in Image and Video Databases*, pages 173–187, 1993.

M. R. Teague. Image analysis via the general theory of moments. *J. Opt. Soc. Amer.*, 70: 920–930, 1980.

Charles W. Therrier. *Decision estimation and classification: An introduction to pattern recognition and related topics.* John Wiley & Sons, 1989.

Dian W. Tjondronegoro. *Content-based Video Indexing for Sports Applications.* PhD thesis, Deakin University, 2005.

Pablo Toharia, Oscar D. Robles, Ángel Rodríguez, and Luis Pastor. Combining shape and color for automatic video cut detection. In *TREC Video Retrieval Evaluation Online Proceedings*, 2005.

Simon Tong. *Active Learning: Theory and Applications.* PhD thesis, Stanford University, 2001.

Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. In Pat Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 999–1006, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

L. Tran and R. Lenz. Pca-based representation of color distributions for color-based image retrieval. In *International Conference in Image Processing (ICIP'01)*, volume 2, pages 697–700, Thessloniki, Greece, Oct. 2001.

O. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition. *Pattern Recognition*, 29(4):641–662, 1996.

Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 219–227, 2000a.

Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh. Improved fade and dissolve detection for reliable videosegmentation. In *Proc. International Conference on Image Processing*, volume 3, pages 961–964, 2000b.

Roland Tusch, Harald Kosch, and Laszlo Böszörmenyi. Videx: An integrated generic video indexing approach. In *Proceedings of the ACM Multimedia Conference*, pages 448–451, 2000. `http://citeseer.ist.psu.edu/tusch00videx.html`.

H. Ueda, T. Miyatake, and S. Yoshizawa. Impact: An interactive natural-motion-picture dedicated multimedia authoring system. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '91)*, pages 343–350, 1991.

Oguzhan Urhan, M. Kemal Gullu, and Sarp Erturk. Shot-cut detection for b&w archive films using best-fitting kernel. *International Journal of Electronics and Communications (AEU)*, 2006. Available on `http://dx.doi.org/10.1016/j.aeue.2006.08.002`.

A. Vailaya and A.K. Jain. Detecting sky and vegetation in outdoor images. In *Proceedings of SPIE: Storage and Retrieval for Image and Video Databases VIII*, volume 3972, San Jose, USA, 2000.

A. Vailaya, A.K. Jain, and H.-J. Zhang. On image classification; city images vs. landscapes. *Pattern Recognition*, 31(12):1921–1936, 1998.

N. Vasconcelos and M. Kunt. Content-based retrieval from image databases: current solutions and future directions. In *International Conference in Image Processing (ICIP'01)*, volume 3, pages 6–9, Thessaloniki, Greece, Oct. 2001.

T. Vlachos. Cut detection in video sequences using phase correlation. *IEEE Signal Processing Letters*, 7(7):173–175, 2000.

James Ze Wang. Methodological review - wavelets and imaging informatics : A review of the literature. *Journal of Biomedical Informatics*, pages 129–141, July 2001. Avaliable on `http://www.idealibrary.com`.

Chau Wing-San, O.C. Au, and Chong Tak-Song. Key frame selection by macroblock type and motion vector analysis. In *IEEE International Conference on Multimedia and Expo (ICME '04)*, volume 1, pages 575–578, 2004.

W. Wolf. Key frame selection by motion analysis. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal*, volume 2, pages 1228–1231, 1996.

Jing-Un Won, Yun-Su Chung, In-Soo Kim, Jae-Gark Choi, and Kil-Houm Park. Correlation based video-dissolve detection. In *International Conference on Information Technology: Research and Education*, pages 104 – 107, 2003.

Zhao Xu, Xiaowei Xu, Kai Yu, and Volker Tresp. A hybrid relevance-feedback approach to text retrieval. In *Proc. of the 25th European Conference on Information Retrieval Research (ECIR'03)*, pages 281–293, April 14-16 2003.

S. Abe Y. Tonomura. Content oriented visual interface using video icons for visual database systems. *Journal of Visual Languages and Computing*, 1(2):183–198, 1990.

Jun Yang and Alexander G. Hauptmann. Exploring temporal consistency for video analysis and retrieval. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 33–42, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-495-2.

Shuping Yang and Xinggang Lin. Key frame extraction using unsupervised clustering based on a statistical model. *Tshinghua Science and Technology*, 10(2):169–173, 2005.

Haoran Yi, Deepu Rajan, and Liang-Tien Chia. A motion-based scene tree for browsing and retrieval of compressed videos. *Information Systems*, 31:638–658, 2006.

H. Yu, G. Bozdagi, and S. Harrington. Feature-based hierarchical video segmentation. In *ICIPInternational Conference on Image Processing (ICIP' 97)*, volume 2, pages 498–501, 1997.

Ho Yu-Hsuan, Lin Chia-Wen, Chen Jing-Fung, and Mark Hong-Yuan. Fast coarse-to-fine video retrieval using shot-level spatio-temporal statistics. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(5):642–648, 2006.

J. Yuan, W. Zheng, Z. Tong, L. Chen, D. Wang, D. Ding, J. Wu, J. Li, F. Lin, and B. Zhang. Tsinghua university at trecvid 2004: Shot boundary detection and high-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004. Tsinghua National Laboratory for Information and Technology.

Jinhui Yuan, Jianmin Li, Fuzong Lin, and Bo Zhang. A unified shot boundary detection framework based on graph partition model. In *Proc. ACM Multimedia 2005*, pages 539–542, Nov 2005.

Yusseri Yusoff, William J. Christmas, and Josef Kittler. A study on automatic shot change detection. In *Proceedings of the Third European Conference on Multimedia Applications, Services and Techniques (ECMAST '98)*, pages 177–189, London, UK, 1998. Springer-Verlag.

R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, 7(2):119–128, 1999.

R. Zarih, J. Miller, and M. Kai. Feature-based algorithms for detecting and classifying scene breaks. In *ACM Int. Conf. on Multimedia*, pages 97–103, San Francisco, November 1996.

L. Zeinik-Manor and M. Irani. Event-based analisys of video. In *Proc of the 2001 IEEE Computer Society Conference on, The Weizmann Institute of Science, 2001*, pages 11–17, 2005.

H.-J. Zhang, C.Y. Low, and S.W. Smoliar. Video parsing and browsing using compressed data. *Multimedia Tools Appl.*, 1(1):89–111, 1995.

H.J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

Hong Jiang Zhang, Jianhua Wu, Di Zhong, and Stephen W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30(4):643–658, 1997.

HongJiang Zhang. Content-based video browsing and retrieval. pages 83–108, 1999.

L. Zhao, W. Qi, S. Z. Li, S. Q. Yang, and H. J. Zhang. Key frame extraction and shot retrieval using nearest feature line. In *Proc. of ACM International Workshop on Multimedia Information Retrieval*, pages 217–220, 2000.

Jie Zheng, Fengmei Zou, and Mandel Shi. An efficient algorithm for video shot boundary detection. In *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004*, pages 266–269, 2004.

W. Zheng, J. Yuan, H. Wang, F. Lin, and B. Zhang. A novel shot boundary detection framework. In *Proc. SPIE Vis. Commun. Image Process*, volume 5960, pages 410–420, 2005.

X. Zhou and T. Huang. Relevance feedback in image retrieval: a comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.

Y. Zhuang, T.S. Huang, and S. Mchrotra. Adaptive key frame extraction using unsupervised clustering. In *Int. Conference on Image Processing (ICIP 98)*, volume 1, pages 866–870, 1998.