Semantic Macro-Segmentation of Multimedia Sequences

Viachaslau Parshyn

A Thesis under Supervision of
Prof. Liming Chen, Lab. LIRIS, Ecole Centrale de Lyon

LYON 2006

Acknowledgments

I would like to express my thanks to my research supervisor Liming Chen for his confidence to me, support and guidance throughout my thesis. I'm grateful to him and his colleague Alexandre Saidi for their help in my accommodation and adaptation on my arrival in France. I would also like to express my acknowledgments to my colleagues Walid Mahdi, Hadi Harb and Mohsen Ardebilian who were my predecessors in the field of video structuring and provided theoretical background for my thesis which is in part the continuation of their work.

I would like to thank all my colleagues at the Department with whom I had have pleasure to work in a friendly atmosphere. I express my warmest thanks to my fellows Dzmitry Tsishkou and Aliaksandre Paradzinets, with whom I was working over years at the same laboratory, for their practical help and moral support, fruitful discussions and interchanges of ideas, the aid in everyday life.

Contents

Introduction

1.1 Research Topic

The progress in information technologies and appearance of more and more powerful front-end devices lead to a constantly growing amount of digitized video used in various fields of application, including video archives, distance learning, communication, entertainment etc. A content-based access could greatly facilitate navigation in huge video storages, providing, for instance, hierarchical tables of content and allowing a user to locate the segment of interest by browsing at first longer high level semantic units and moving then to shorter low level ones. Organizing video according to its semantic structure could also benefit the task of automatic video retrieval, restricting the search by the scope of meaningful semantic segments. Another potential area of application is an automatic generation of video summaries or skims preserving the semantic organization of the original video.

Manual indexing the content of video often is not appealing as it is tedious and requires much time and human resources. This is aggravated by the fact that a content structure is not always unique and various definitions can be proposed depending on the needs of a particular user such as a desirable level of detailing. The aim of this work is to develop approaches to automatic generation of video content table representing temporal decomposition into meaningful semantic units. This generation of high-level content table is based on lower-level indexes and is called a macro-segmentation. We adopt and test the developed approaches for the scope of sports video (tennis) and feature films hoping that they will be general enough to be applied to other types of video as well.

1.2 Problems and Objectives

As the basic building blocks of professional video are the shots – sequences of contiguous frames recorded from a single camera, it is natural to divide a video into these units. Many effective and quite reliable shot segmentation techniques have been proposed [ARD 00, BOR 96, LIE 99, LIE 01]. Unfortunately, the semantic meaning provided by shots is of a too low level. Common video of about one or two hours, e.g. a full length movie film, contains usually hundreds or thousands of shots – too many to be efficient representation of the content. In this work we focus on the task of automatic segmentation of video into more meaningful high-level time units which share a common semantic event. These units are usually considered as aggregates of shots and can have various semantic meaning depending on the type of video and

desirable content structure definition. For narrative films they are defined as narrative segments representing a common dramatic event or locale. For sports video semantic segments can distinguish logical parts of a match, e.g. points, games or sets in tennis video which can be collected into a hierarchical content structure where a higher-level segment can include several nested lower-level ones.

Many existing works aim to elaborate specific signal-based features destined to distinguish only some particular segments or short-time events. To unfold the whole content structure of a given video there is often the need to combine multiple features that sometimes are dispersed through time and to take into consideration grammar constraints imposed into possible content structures. The aim of this work is to propose quite a general approach which allows one to express his knowledge about a specific content structure of a specific type of video in terms of combinations of detectable features (that can have quite a general semantic meaning) characterizing semantic segments and to impose grammar constraints so as to enable automatic content parsing.

In real-world applications features of one or just several types often cannot provide reliable segmentation accuracy due to erroneous detections of the features and ambiguities in their relation to semantic segments. It is usually possible to find several types of features, extracted sometimes from different modalities, providing the evidence about the same entity. For instance, in the task of narrative film segmentations into scenes visual similarity between two adjacent groups of shots can be used to separate scenes together with audio dissimilarity characterizing a change of sound sources. So, there is a need to properly combine these multiple features so as to compensate for their inaccuracies. The common approach uses a set of rules according to which one source of information is usually chosen as the main one, used to generate initial segment boundaries, while the others serve for their verification or further decomposition into segments. Rules based techniques, however, are convenient for a small number of features, generally do not take into account fine interaction between them and are hardly extensible. Another frequent drawback inherent to such methods is binarization of real-valued features that often leads to losses of information. In this work we attempt to include excessive features into the segmentation approach in a systematic and flexible manner, without the need to make intermediate restricting decisions as it is done in many rule-based techniques.

1.3 Our Approach and Contributions

We propose a deterministic approach for the task of automatic video segmentation which is a sort of a finite automaton whose states relate to content units [PAR 05a]. The approach allows for multilevel hierarchical content structures which are generated recursively, beginning at the

highest semantic level. At each semantic level the parsing automaton is governed by specific templates which cause state transitions according with grammar restrictions. These templates are defined as combinations of intermediate semantic features or short-term events connected by certain relationships in time. They allow one to express prior knowledge about particular characteristics of semantic segments, usually relying on specific production rules that are typically employed by video producers to convey semantic information to a viewer.

An advantage of the proposed segmentation approach is in its expressiveness and low computational complexity. As it is based solely on prior knowledge, it does not require preliminarily learning and can be employed at once, without the need of tedious manual annotation of learning data. We apply and experimentally evaluate this approach on the task of tennis video segmentation where output content is naturally represented in a hierarchical manner so as an input tennis match at first is divided into sets and pauses between sets, or breaks, then each set is further decomposed into games and breaks etc. Automatically recognized score boards and tennis court views are used as intermediate events in this task.

In practical applications the detectable features often cannot be related to the semantic segments unambiguously. To reduce this ambiguity and, hence, to enhance the segmentation performance, multiple features should be fused into the final decision. So, there is a need to resolve properly the conflicts between these features. The number of possible combinations growth exponentially with the number of features, and it becomes too difficult to enumerate all these combinations in the framework of the deterministic approach, especially when the features are real-valued. To enable inferring the fusion rules automatically based on a set of manually labeled learning data, when available, we also propose a stochastic segmentation approach [PAR 05b], where the feature uncertainty is modeled explicitly. Moreover, the approach deals properly with probabilistic time constraints imposed on semantic segments durations. Its stochastic nature allows for fusion of multi-modal audio-visual evidences in a symmetrical, consistent and scalable manner. Instead of definitive rules of the deterministic approach, the posterior probability of semantic segment transitions is estimated first. Segment boundaries are then positioned so as to maximize the total posterior probability. It is shown that such a decision rule yields the maximal recall and precision which are commonly used segmentation performance measures. A computationally tractable algorithm for the corresponding task of constrained maximization is proposed. The posterior probability of segment boundaries is estimated using, in particular, a variable duration hidden Markov model which has been proved to be a powerful mean in modeling of time sequences. In contrast to the Viterbi segmentation procedure, which is commonly used with hidden Markov models to find the most probable path, we, however, select optimal segment boundaries so as to maximize the segmentation performance directly.

As an alternative to the posterior probability maximization total for the whole input video, a one-pass version of segmentation approach is proposed which selects each subsequent segment boundary as the most probable one assuming that the previous boundary is known definitively [PAR 06]. This modification is particularly useful for real-time applications where segmentation is performed already before the end of video is attained.

The proposed stochastic approach has been applied and experimentally evaluated on the task of narrative film segmentation into scenes. The test results showed enhancement of segmentation performance when multiple audio-visual segment evidences of segment boundaries are fused and time constraints are taken into account. The resulting performance was higher as compared to deterministic rule-based fusion techniques. Higher segmentation performance was also observed in the case where our segmentation criterion, that maximizes the total posterior probability of segment boundaries, was applied using a hidden Markov model instead of the commonly used Viterbi segmentation algorithm.

In this work we are also concerned with the problem of video summarization – compact representation of the original video. A video summary can have an independent meaning aimed to quickly get acquainted a viewer with the content of video or it can be generated for each semantic segment of a content table forming so called digest. Pictorial digests provide a convenient interface for navigation with content tables where each unit is visually represented with one or just several key frames. We propose a versatile approach which can be used to create summaries that are customizable to specific user's preferences to different type of video [PAR 04]. A high versatility of the approach is based on a unified importance score measure of video segments which fuses multiple features extracted from both the audio and video streams. This measure provides the possibility to highlight the specific moments in a video and at the same time to select the most representative video shots. Its coefficients can be interactively tuned due to a high computational speed of the approach.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. In the next chapter we give a review of the related work in the field of semantic video segmentation. In particular, we consider separately two video genres of interest in this work – sports video and narrative films, aiming to describe the state of the art, give semantic segments definition and provide some background information. Then we provide discussion concerning the related work and motivate our segmentation approach.

In chapter 3 we present our deterministic approach which infers a hierarchical content table based on mid-level events extracted from a raw video. We apply this approach to sports game video, namely to tennis one, as it has a well-defined temporal content structure whose

segments can be unambiguously related to mid-level events. As a result, a fully automatic content parsing system is built and tested on a ground-truth video.

In chapter 4 a stochastic approach to the video segmentation task is proposed. We first consider the general principles how to chose the optimal segments based on the corresponding probability estimates so as to maximize recall and precision metrics of system performance. Then we consider the video segmentation task based, more specifically, on a hidden Markov model and its extensions.

In chapter 5 we adopt and experimentally evaluate our stochastic segmentation approach to the task of narrative films segmentation into scene segments. We give a strict definition of a scene and describe our database of ground-truth video used for performance comparisons. Then we propose audio-visual features which provide evidence about scene boundaries. After this we derive and evaluate several particular segmentation techniques.

In chapter 6 we propose a video summarization approach using a shot-based approach that allows generating both a static storyboard and a video skim in the same manner. Video summary is generated based on our unified measure of video segment importance which fuses multiple features extracted from both the audio and video streams.

In chapter 7 we present final conclusions concerning this thesis and discuss some directions of our future work.

2 State of the Art

Semantic segmentation of video requires that several decisions be made. First, the underlying content structure and the meaning of the corresponding semantic segments must be defined. While the lower-level segments of professional video are traditionally chosen to be camera shots, the definition of the higher-level semantic segments is highly dependent on the type or genre of video and the specific of the practical needs. Second, if we consider the task of automatic indexing, relevant signal-level features or mid-level events should be properly chosen. Third, robust content indexing usually requires the use of multiple features extracted from multiple modalities, so there is a need to choose the method of their integration to obtain the final segmentation. As videos of different types or genres convey different semantic meaning and are produced using different production models, the genre of video has a strong impact on an automatic content parsing system to be developed, especially as it concerns the first two decisions mentioned above. In this chapter we consider these choices separately for two video genres of interest in this work – sports video and narrative video, aiming to describe the state of the art, give semantic segments definition and some background information and discuss the motivations of our segmentation approach.

2.1 Sports Video Segmentation

Works concerned with sports video content indexing problem usually aim at detection and classification of only one or several specific semantic segments or events of interest. These segments, referenced hereafter as events, are often distinguished based on signal-level features using domain-specific models or pattern recognition techniques and can be used as independent indexes or as intermediate-level semantic keys or syntactic elements for further analysis. In this section we first consider several approaches to event detection and then describe works concerned with complete content structure analysis of sports video where event detection is often used as a necessary preliminary step of video processing.

2.1.1 Event Detection

Sports video is often taken by a number of cameras mounted in fixed positions, each providing a certain unique view. The cameras are switched in a manner that certain views correspond to specific events, such as serves or rallies in tennis and pitching in baseball. Detection of these

events, hence, can be performed as shot segmentation followed by recognition of the correspondent shots.

In [DIZ 01] views covering tennis court field with players (referenced hereafter as global court views, see Figure 2-1) are detected to distinguish serve or rally scenes. Corresponding camera shots are represented by a color histogram feature computed for shot key frames. A supervised k-means learning algorithm is used to cluster manually labeled court view shots so that each cluster represents a specific tennis match model. To make their approach applicable to various types of match models, the authors propose to include learning samples extracted from as many different tennis broadcasts as possible. A shot then is considered to be a court view if it is close enough to one of the clusters. To remove false alarms the authors also use an additional procedure verifying that players on the tennis court have consistent locations based on an automatic player segmentation technique.

Figure 2-1. Global court views in tennis match

In [ROZ 98] court views are detected using multimodal data extracted from the image sequence and the audio track of an input video. The second moment of the Hough transform [HOU 59] of the edges averaged over the frames of a shot is used as the feature extracted from the image sequence, the idea being to catch the geometry of tennis court lines. In the auditory domain the corresponding feature reflects the possibility of racket hits to be present in the shot sound track and is calculated as follows. First, a learning set of acoustic vectors (power spectrum coefficients) corresponding to rocket hits is collected. A principal component analysis is then performed over these training data and J eigenvectors corresponding to the J highest eigenvalues are retained to span the eigenspace. The audio feature is finally calculated as a distance between the closest acoustic vector of an input shot and the eigenspace. The decision rule is based on the likelihood value of both the audio and visual features which are modeled with the Gaussian distributions and considered to be independent given the shot class. An input shot is claimed to

be a general view if the joined likelihood exceeds a threshold value. Experimental evaluations on a ground-truth video has shown that the fusion of multi-modal data sources enhance the precision of view classification.

After the event segments have been localized with one of quite general view classification technique, more specific detectors can be applied to further classify tennis events based on analyses of players' positions and their movements. Miyamori and Iisaku [MIY 00] automatically annotate different tennis actions considering three representation methods: based on player position only, based on player and ball position and based on both positions plus player's behavior. Player and ball position is considered with respect to tennis court geometry extracted using static color filters corresponding to several standard court types. Player's behavior is modeled with a hidden Markov model (HMM) [RAB 89] which allows the authors to categorize player's swings into three classes: backside, foreside and over-the-shoulder swing.

Zivkovic et al. [ZIV 01] recognize different classes of tennis strokes, such as service, smash, backhand slice etc. by modeling player action in the visual domain with a HMM. First, the player in the lower rectangle of global court view frames is segmented from the background. A robust player segmentation algorithm is proposed which separates player region pixels from the tennis field and court lines based on estimated statistics of tennis field dominant color and a 3D model of court geometry. Then, the authors extract a number of different features from the player binary representation: orientation and eccentricity of the whole shape, the position of the upper half of the mask with respect to the mass center, sticking-out parts of the shape etc. Finally, player activity represented by an input frame sequence is classified into different tennis stroke types using discrete left-to-right HMMs pre-trained on a set of manually labeled data.

The progress of certain events in sports broadcasts are captured by several cameras switching according to specific production rules. Thus, these events normally correspond to sequences of views or scene shots of certain types which can be recognized and tracked automatically using rule-based or stochastic techniques. In [CHA 02] seven types of scene shots are recognized for the purpose of baseball game highlights detection: pitch view, catch overview, catch close-up, running overview, running close-up, audience view and touch base close-up (see Figure 2-2). The authors distinguish four baseball highlights (nice hits, nice catches, home runs and plays within the diamond) by modeling the corresponding shot scene sequences with HMMs. Hidden states of these models represent the mentioned above types of scene shots. The state probability values are estimated using the following features extracted from the image sequence: a field shape descriptor (positions of field grass or sand blocks), an edge descriptor, camera motion, the amount of grass and sand, player height.

Figure 2-2. Seven types of scene shots of a baseball game [CHA 02].

Ekin, Tekalp and Mehrotra [EKI 03] propose a technique for the task of automatic soccer goal detection which is based on a set of rules used to combine information about specific shot scenes types and their duration. A goal event leads to a break in the game which is used by the producers to convey the emotions on the field to the TV audience and show one or more replays for a better visual experience. As a result, occurrence of a goal is generally followed by a special pattern of cinematic features. To detect this pattern, the authors define a cinematic template which is a set of constraints imposed on the appearance of certain shot scene types (player close-up, out of field and slow-motion replay) and their relative positions. The required shot classification is performed automatically based on the ratio of grass color pixels for player close-ups and out-of-field shots and on analysis of frame-to-frame change dynamics for slow-motion replays.

2.1.2 Semantic Structure Analysis

Sports videos, especially sports games, usually have a quite well-defined content structure. A number of techniques have been proposed by now aiming at automatic content parsing of these videos based on intermediate-level semantic events or low-level features. In the simplest case the content structure is represented by a simple one-level chain of semantic segments. In a more general case this structure is represented hierarchically so that semantic levels of a higher level can include several nested lower level segments. For example, a tennis match is divided first into sets, each set is decomposed into games which in their turn are further divided into points.

A rule-based technique of one-level decomposition of a soccer video into a chain of semantic segments of two types – play and break [SOC] is proposed in [XUP 01]. Each frame of an input video is first classified into three kinds of view (global, zoom-in and close-up, see Figure 2-3) using a unique domain-specific feature, grass-area-ratio. To handle possible variations of lighting and filed conditions, this feature is calculated based on grass color estimated adaptively

for each video clip. Then heuristic rules are used in processing the view label sequence to obtain play/break segmentation. These rules take into consideration the time duration of views and their relative positions in time.

Figure 2-3. Three kinds of view in soccer video [XUP 01].

In [XIE 02] a stochastic approach based on HMMs is proposed for the same task of soccer video segmentation into plays and breaks. Instead of discrete labels identifying view type of each video frame, real-valued observations are used: dominant color ratio and motion intensity. The color ratio indicates mainly the scale of view in the current shot, taking high values for wide shots and low values for close-ups. Motion intensity roughly estimates the gross motion in the whole frame, including object and camera motion. HMMs are applied to classify a smoothed feature vector sequence in a fixed-length sliding window. The resulting probabilities of play/break classes are then smoothed using a dynamic programming technique to obtain the final segmentation. The experimental evaluations has shown the performance improvements of the HMM-based approach with respect to the discrete rule-based one described above.

Relevant information about semantic segments of sports video can be provided, if available, by the close-caption text which is the speech transcript of the announcers. Nitta and Babaguchi [NIT 02] propose a generic scheme for segmentation of TV sports game programs into “Live”, “Replay”, “Others” and “Commercial Message” scenes. The closed-caption text is used as the only data source. Segmentation is performed through the labeling of each of the close-captioned segments into one of the four target scene categories listed above. Six features are first extracted for each closed-caption segment: the name of the announcers, the number of sentences, the length of the sentences, the number of players’ names, the situational phrases and the numbers. Then a Bayesian network is used to estimate the probability that each segment belongs to one of the category x as:

$$P(x) = \sum_{all\ b} P(x \mid b) P(b) \prod_{j=1}^{|F|} P(f_j \mid x), \qquad (2\text{-}1)$$

where b stands for the category of the previous segment, $F \equiv \{f_1, f_2, ...\}$ is the feature space. Thus, the category of a closed-caption segment depends on the category of the previous segment as well as its own features. Finally the category of the each segment is set to the one which has the maximal probability. Unfortunately the authors do not fuse the textual information with the audio and visual modalities to possibly achieve better performance.

Hierarchical semantic content analysis for sports game video is considered in [CHE 04]. The authors define recurrent important semantic parts during the game as "Basic Semantic Unit" (BSU). An example of such a BSU is a serve in tennis or pitching in baseball. Accordingly, the residual less-important parts are non-BSUs, e.g. commercial breaks or changes of players. Thus, sports video is modeled as a sequence of BSUs interleaved with non-BSUs, where each BSU can be further decomposed into the same sequence of the lower semantic level. An automatic technique for segmentation of soccer video according to two-level semantic structure is realized. The first-level segmentation is performed through advertisement detection based on the fact that advertisement shots are short duration and are accompanied with speech and music sound. At the second semantic level non-advertisement parts are further decomposed into plays and breaks based on view classification and using heuristic rules similar to [XUP 01] considered above.

In [KOL 04] the authors propose to parse the evolution of a tennis match through tracking elementary events, such as the tennis ball being hit by the players, the ball bouncing on the court, the players' positions etc. Guided by the rules of the game of tennis they build a graphical model which allows them for awarding of a tennis point. Two-level hierarchical representation of this model is given in Figure 2-4. The evolution of a tennis point can be inferred using statistical reasoning tools (such as HMMs) or rule based tools – such as grammars. The authors have applied deterministic rules in their experimental evaluations carried out on a ground-truth tennis video comprising about 100 points. As the elementary events in these experiments were extracted manually, without errors, and the deterministic rules were not broken, the perfect accuracy of 100% was achieved. However they propose to use statistical reasoning tools if these events are detected automatically to deal properly with detection errors. A little extension is required to the proposed point awarding model to move on to the award of games and sets in the match. For instance, games of a tennis match are awarded out of points won by both sides as follows: if a player has scored 4 or more points in the current game and his/her opponent has at least 2 points less, then this player has won the game – otherwise the game goes on.

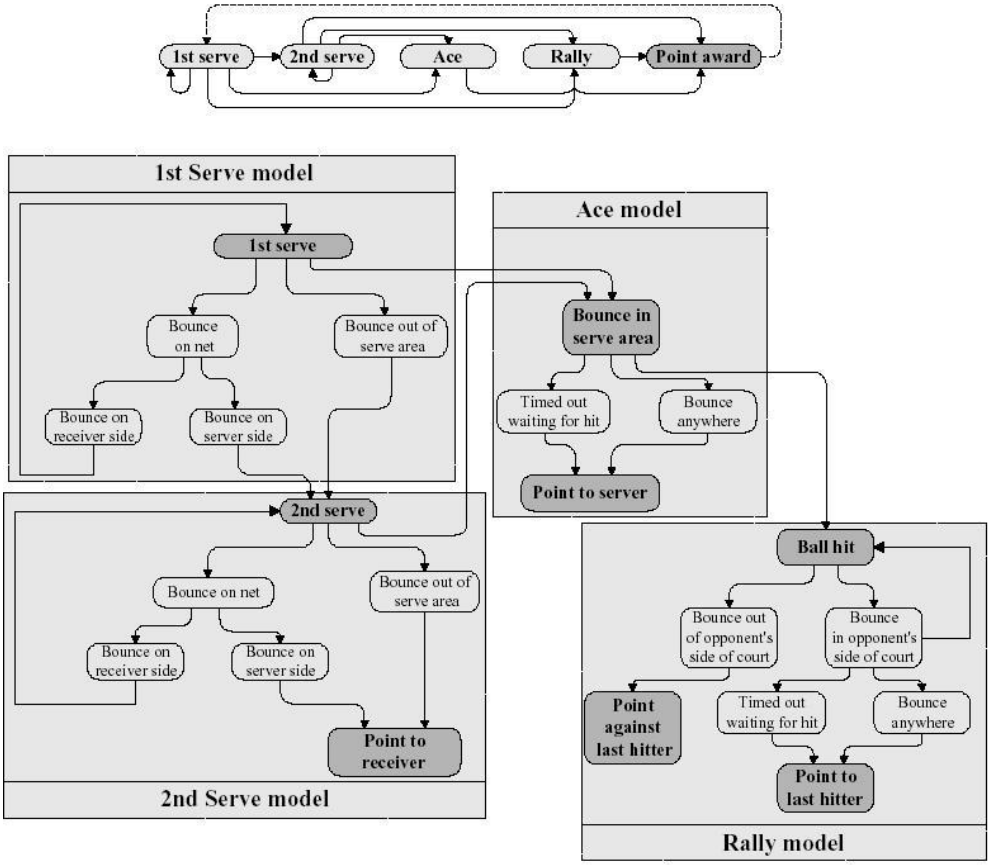


Figure 2-4. Two-level graphical model for awarding a point in a tennis match [KOL 04].

A statistical approach based on a hierarchical hidden Markov model (HHMM) [SHA 98] is used in [KIJ 03a, KIJ 03b] for full content structure parsing for tennis video. According to such a structure a tennis match is first divided into tennis sets and breaks (pauses between sets), each set is decomposed into tennis games and breaks and each game is finally divided into a chain of tennis points. HMMs are used to classify tennis points into several types: missed first serve and rally, rally, replay and break. These HMMs are included as the lowest level of into a HHMM used to represent the syntactical constraints stemming from tennis game rules (see Figure 2-5). It allows the authors to take into account the long-term structure of a tennis match. The content decomposition of an input tennis video is performed as follows. At first the video is segmented into shots and per-shot feature vectors are computed. The feature vectors combine the data extracted from both the image sequence and the audio modality and include shot duration, dominant colors and their respective spatial coherencies, a measure of camera motion and audio classes (speech, applause, ball hits, noise and music). The sequence of feature vectors

constitutes, then, the input chain to the HHMM. Segmentation and classification of the observed sequence into the different structural elements are finally performed simultaneously using a Viterbi algorithm [VIT 67] which finds the most likely sequence of the HHMM states.

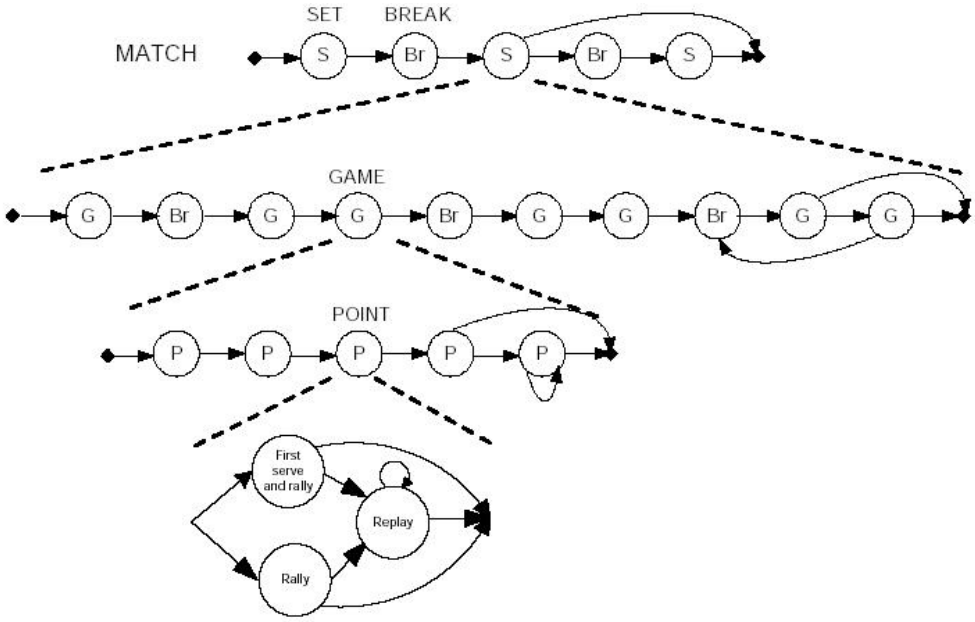


Figure 2-5. Content hierarchy of broadcast tennis video [KIJ 03a].

2.1.3 Discussion

The large diversity of sports videos and semantic segments/events leads to a variety of proposed video segmentation techniques, which especially concerns event detection or feature extraction. To attain the reliable detection of events or their detailed classification, various domain-specific fine-tuned techniques are often elaborated. Instead of elaboration of specific detectors, in the task of sports video segmentation we rather rely on quite common characteristics of video stemming from production rules, such as specific views or score boards. These events appear to be enough to achieve the final goal – the full video decomposition into content elements. Moreover, being quite common properties, such events provide us with quite a general basis to deal with the diversity of video sub-genres in a unified fashion. Using common elements allows us to avoid the difficult task of semantic understanding of the video content and to be based solely on the video organization syntax provided by producers.

Currently we do not try to elaborate complicated event detectors, applicable to the possibly largest number of videos representing the same sport, since this usually requires the use of

domain-specific models. Also we do not adopt statistical techniques, such as hidden Markov models, as they usually require that quite a large set of manually marked-up learning data be provided. We aim to propose quite a simple segmentation technique which needs the minimum preparations, such as learning. To attain the reliable event detection, we apply quite common signal processing techniques that do require some simple learning to be adapted to a particular type of video, but this learning does not consume a lot of time. Different views, for example, can be identified using dominant colors matching or visual similarity with learning samples; score boards are distinguishable due to their fixed form and position on the screen etc. Each of these events needs only one learning sample for a set of videos produced at the same setting using the same rules, e.g. all tennis matches of the same championship broadcasted over the same TV channel.

Instead of detection of semantic segments of just one or several types, that is often the case in the related work, in this thesis we aim at reconstructing the total content structure of video. It is a more general task as the elements of an output content table can contain specific segments of interest. Moreover, grammar restrictions and time duration constraints that are generally imposed into the content structure provide useful additional information restricting the choice of allowable semantic segments and their duration given the context.

Semantic segmentation of video usually integrates multiple low and mid-level features and is performed using generally two types of methods – deterministic rule-based and stochastic (usually based on HMMs). In spite of the powerful capacity of stochastic methods to deal properly with the uncertainty of observable data, they, however, generally require preliminary learning to fit data distributions. This takes additional computational resources and, which is often more problematic, gives rise to a need of manually labeled training data, thus making stochastic methods less appealing, especially in our case where the event detectors are adapted to specific production rules and, hence, a change of these rules requires the re-learning of the whole system. The advantage of rule-based techniques is that they allow us to express directly our understanding of relationships between detectable features and the semantic structure of a particular type of video. The applied rules, however, are not usually formulated on a regular basis, which makes us to rebuild the whole content generation system when these rules are changed. Moreover, such a set of rules can become too complicated if many constraints are taken into consideration simultaneously. In this thesis we develop a deterministic approach based on a finite state automaton which allows us to formulate video content parsing rules as grammar constraints and feature templates that control transitions between semantic segments. The approach can be realized as a generic engine adaptable to different types of video and content configurations. It is suitable for video having complex hierarchical content structure for which

reliable feature templates can be specified. In this thesis we adopt and test this approach for the task of tennis video segmentation.

2.2 Segmentation into Scenes

In this section we consider some basic ideas and related work concerned with the problem of automatic segmentation into logical story units or scenes. These units combine one or several shots and are basic meaningful elements of the organizational structure of narrative films such as feature films or sitcoms. Viewers often identify scenes intuitively as important events that have a complete meaning necessary for perceiving the whole story, e.g. scenes showing dialogue between two persons, pursuit scenes etc. However, as a notion of scene is based on human understanding of its meaning, it is difficult to give an objective and concise definition that covers all possible scenes judged by humans.

In cinematography a scene is defined as “a segment in a narrative film that takes place in one time and space or that uses crosscutting to show two or more simultaneous actions” [BOR 97] or “a series of shots that communicate a unified action with a common locale and time” [BOG 00]. Even these definitions, commonly accepted by researchers, are somewhat vague. It is not always clear, for instance, how to interpret the “common locale” property of a scene. Indeed, since the establishing shot often precedes a scene showing the outside of the building where the scene takes place, the “common locale” part of the scene definition allows for broad interpretations. Another difficulty is how to treat several actions showed simultaneously or parallel events.

To overcome this uncertainty, a more precise definition often is given. In [WAL 04] establishing shots are merged with the subsequent scenes; parallel actions are merged into one scene. A specific traveling scene is additionally defined which shows a traveling person passing through many locales very briefly. The shots of such a scene are unified by a common traveler rather then by a common place. In [VEN 02] a scene defined based on 4 important editorial techniques: elliptical editing, montage, establishing shot and parallel cutting. Due to the first two techniques viewers perceive scenes being continuous in space and time; an establishing shot is considered to be the part of the scene for which it determines the setting; the parts of parallel events are merged into one scene if they are composed from three or less shots. To avoid the problem of semantic understanding of scenes, Sundaram [SUN 02] focuses in his research on the detection of so-called computable scenes. These scenes are defined looking at the relationships between contiguous chunks of video and audio and structured segments (such as dialogues which are composed from interleaving shots). The only useful property of computable scenes which the author is interested in is that they are computable, i.e. they can be computed automatically using

low-level audio-visual features. Their semantic meaning is no of high importance since the goal of the proposed scene segmentation algorithm is to assist in the solution of another problem.

In spite of some diversity in scene definitions, the basic principles of segmentation often remain the same. Segmentation into “generic” scenes, without specifying their precise semantic meaning, is commonly based on the similarity of the constituent shots which stems from the production rules applied during the creation of narrative films and the common locale. Indeed, the underlying organizational structures depend on certain human expectations about the timing and placement of camera shots within a single scene. According to this most scenes are shot from several viewpoints with several cameras that are switched repeatedly. So, they can be detected from the image track as a group of interleaving visually similar shots. The similarity is established using the low level visual features such as color histograms or motion vectors [MAH 02, KEN 98, RAS 03, WAL 04, VEN 00]. On the other hand, a scene transition in movie video usually entails abrupt change of some audio features caused by a switch to other sound sources and by film editing effects [HAR 03c, CAO 03, SUN 00, CHE 02]. Hence, sound analysis provides useful information for scene segmentation as well.

Further in this section we first consider separately approaches based on shot similarity measured in visual and audio domains in the corresponding two subsections. Then the principles of fusing features from multi-modal data sources are described. After this we consider several segmentation systems that extract specific scenes or classify them into predetermined classes based on specific semantic keys. The final discussion then finishes up this section.

2.2.1 Visual Similarity-Based Segmentation

The common approach to video scene segmentation in the visual domain exploits the visual similarity between shots provided by specific editing rules applied during film montage [BOR 97]. According to these rules video scenes are usually shot by a small number of cameras that are switched repeatedly. The background and often the foreground objects shot by one camera are mostly static and, hence, the corresponding shots are visually similar to each other. In the classical graph-based approach [YEU 96] these shots are clustered into equivalence classes and are labeled accordingly. As a result, the shot sequence of a given video is transformed into a chain of labels identifying the cameras. Within a scene this sequence usually consists of the repetitive labels. When a transition to another scene occurs, the camera set changes. This moment is detected at a cut edge of a scene transition graph built for the video. For example, a transition from a scene shot by cameras A and B to a scene taken from cameras C and D could be represented by a chain ABABCDCD, where the scene boundary would be pronounced before the first C. Analogous approach was proposed in [RUI 99], where shots were first clustered into

groups which then were merged into scenes. Wallapak Tavanapong and Junyu Zhou [WAL 04] in their ShotWeave segmentation technique use additional rules to detect specific establishment and re-establishment shots which provide a wide view over the scene setting at the beginning and the end of a scene. They also suggest using only specific regions of video frames to determine more robustly the inter-shot similarity.

Two shots belonging to different scenes can be found visually similar because of their accidental resemblance or a reuse of the same locale, e.g. several scenes can take place at the same room. Grouping these shots into one cluster in the graph-based approach would lead to undesirable merging of the corresponding scenes. To reduce the probability of this merging, time constrained clustering is used where two shots which are far apart in time are never combined into one cluster as they are unlikely belonging to one scene. In [YEU 96] a fixed temporal threshold is used to delimitate distant shots. As this threshold should be dependent on the scene duration, an adaptive temporal delimitation is proposed in [MAH 00]. According to this work an input video is first divided into so-called sequences – narrative unities formed with one or with several scenes. The shot clustering is then performed within sequences. The resulting segmentation technique is based on temporal relationships between shots, such as *meets*, *during*, *overlaps* and *before*, defined according to Allen's algebra [ALL 83]. Instead of the scene transition graph of [YEU 96], a temporal-clusters graph is built in [MAH 00], where the aforementioned temporal relationships connect the nodes representing the shot clusters. *Meets* relationships in this graph separate sequences, as they correspond to gradual transitions between shots, while scenes boundaries are discerned using *before* relationships.

To overcome the difficulties resulting from a discrete nature of the segmentation techniques based on shot clustering, such as their rigidity and the need to choose a clustering threshold, continuous analogues have been proposed. Kender and Yeo [KEN 98] reduce video scene segmentation to searching of maxima or minima on a curve describing the behavior of a continuous-valued parameter called video coherence. This parameter is calculated at each shot change moment as an integral measure of similarity between two contiguous groups of shots based on a short-memory model which takes into consideration the limitation and preferences of the human visual and memory systems. Rasheed and Shah [RAS 03] propose to construct a weighted undirected shot similarity graph and detect scene boundaries by splitting this graph into subgraphs so as to maximize the intra-subgraph similarities and minimize the inter-subgraph similarities.

2.2.2 Audio Similarity-Based Segmentation

As the physical setting of a video scene remains usually fixed or change gradually (when, for instance, the cameras follow moving personages), the sources of the ambient sound rest stable or change their properties smoothly and slowly. A scene change results in a shift of the locale and, hence, the majority of the sound sources changes too. This change can be detected as the moment of a drastic change of audio parameters characterizing the sound sources.

Since short-term acoustic parameters often are not capable to represent properly the sound environment [CHE 02], these parameters are often combined within a long-term window. The resulting characteristics are evaluated within two contiguous time windows adjoining a point of potential scene boundary (usually shot breaks) or its immediate vicinity (as sound change sometimes shifted by a couple of seconds during montage to create an effect of inter-scene connectivity) and then compared. A scene boundary is claimed if their difference is large enough.

Sundaram and Chang [SUN 00] model the behavior of different short-term acoustic parameters, such as cepstral flux, zero crossing rate etc, with correlation functions characterizing a long-term properties of the sound environment. A scene change is detected when the decay rate of the correlation functions total for the all acoustic parameters reaches a local maximum, as it means low correlation between these parameters caused by the change of the sound sources. Cao et al. [CAO 03] approximate long-term statistical properties of short-term acoustic parameters using normal distribution. At a potential scene boundary these properties are compared by applying a weighted Kullback-Leibler divergence distance. The experimental evaluations are reported which suggest the better integral performance is attained when this distance is used as compared to the model proposed in [SUN 00]. Harb and Chen [HAR 06] segment the audio track of an input video into so-called audio scenes and chapters using an acoustic dissimilarity measure which combines two terms. The first one is the Kullback-Leibler distance between distributions of spectral parameters. The second term is a so-called semantic dissimilarity measure which is a difference between the results of sound classification into semantic classes: speech, music and noise.

2.2.3 Audio-Visual Data Fusion

While several ad hoc techniques have been proposed for narrative video segmentation into scenes in the visual or audio domains only, there is a lack of methods which fuse both the modalities in a systematic and symmetrical way so as to compensate for their inaccuracy and, hence, achieve better segmentation performance. The common approach to segmentation of narrative video into scenes is based only on visual keys extracted from the image stream. In

order to combine information extracted from the audio and image streams into one more reliable decision, a set of simple rules is usually applied. The audio stream can be used as an auxiliary data source to confirm or reject scene boundaries detected from the image sequence. For example, Cao et al. [CAO 03] first segment video into scenes in the visual domain and then apply sound analysis to remove a boundary of suspiciously short scenes, if it is not accompanied by a high value of audio dissimilarity. In [JIA 00] it is proposed first to segment the video in the audio domain and find potential scene boundaries at shot breaks accompanied by a change in the sound environment; these boundaries are then kept in the final decision if they are confirmed by low visual similarity between preceding and succeeding shots. Sundaram and Chang [SUN 00] first segment video into scenes independently in the video and audio domains and then align visual and audio scene boundaries as follows. For visual and audio scene boundaries lying within a time ambiguity window, only the visual scene boundary is claimed to be the actual scene boundary; the rest of the boundaries are treated as the actual scene boundaries.

2.2.4 Semantic Cues-Based Segmentation

While similarity-based segmentation techniques aim at grouping shots into scenes according to the common setting and disregarding the precise semantic meaning, some approaches have been proposed that additionally assign semantic labels to scenes or detect segments only of one or several specific types using the corresponding semantic keys. A HMM-based method is proposed in [ALA 01] to distinguish dialog scenes – groups of shots containing conversations of people. The states of the used HMMs correspond to camera shots. The per-shot observable data include classification into speech, music and silence for the soundtrack; face detection result (face or no-face label) and scene location change (obtained using a conventional shot visual similarity-based method) for the image sequence. Two types of HMM topology are proposed – circular and left-to-right, the states representing establishing scene, transitional scene or dialogue scene. The final segmentation result is obtained by finding the most probable state path using a Viterbi algorithm.

A technique for violent scenes detection in general TV drama and movies is presented in [NAM 98]. It integrates cues obtained from both the video and audio track. In the visual domain these cues are the spatio-temporal dynamic activity of a shot and specific events. The former is the measure that has a high value for short shots with much motion, which are typical for dynamic action scenes. The events are flame and gunfire/explosion segments which are detected using a predefined color table. In the auditory domain the corresponding cue is energy entropy allowing the authors to detect abrupt changes of the sound energy level which signify bursts of sound such as explosions. A knowledge-based combination of the audio-visual features is used to obtain the final scene classification.

Segmenting a video into four basic scene types (dialogues, stories, actions and generic) is considered in [SAR 98]. The following audio-visual features are used. An inter-shot visual similarity measure is computed within a time window and groups of similar shots are identified and labeled. The sound track is divided into silence, speech, music and miscellaneous segments. Scenes are detected using the following pre-defined rules. Dialogues are distinguished as alternating patterns of visually similar shots occurring together with speech. Stories are detected as repetitive patterns of visually similar shots. Progressive patterns of shots corresponding to non-speech audio segments are marked as stories. Scenes that do not fulfill the aforementioned criteria are recognized as being generic.

2.2.5 Discussion

In order to avoid the difficult problem of automatic understanding of video, scenes are sometimes defined as groups of shots having the similar visual content or/and the consistent audio properties. While this definition justifies the use of simple low-level similarity measures, it does not always correspond to the notion of scene admitted in cinematography. In this work we define scenes as important semantic units of a narrative video unified by a dramatic event or action. The continuity of the locale and time, being a typical property of scenes, together with the video production rules indeed allow us to distinguish scenes as segments of similar visual and audio content, but only with some degree of confidence. To reduce this ambiguity to the maximum extent, we have to choose the proper signal-based features which enable to establish the similarity in the visual and audio domains and to use both the modalities at the same time so as to compensate for their inaccuracy.

To establish the similarity of scene content in the visual domain, we derive a new measure, called video coherence, by considering a continuous generalization of the conventional discrete clustering-based technique which is analogous to the approach of [KEN 98] in the sense that it seeks for scene boundaries at local minima of a continuous measure of video coherence. In contrast to the binary output of the clustering-based segmentation, this measure provides a flexible confidence level of the presence or absence of a scene boundary at each point under examination; so that the lower is this measure, the more possible is the presence of a scene boundary. In contrast to the video coherence of Kender [KEN 98] which is a total sum of inter-shot similarities, our measure integrates only the similarity of the shot pairs that possibly taken from the same camera. In the audio domain we adopt Kullback-Leibler distance as it was been proved to be effective measure for the task of video segmentation into scenes. This distance serves as an audio dissimilarity feature providing the evidence of the presence or absence of a scene boundary in the audio domain. It represents the divergence between distributions of shot-

term spectral parameters estimated using the continuous wavelet transform. Currently we do not use specific semantic keys like speech detection, aiming to elaborate a generic segmentation approach which does not assign semantic labels to scenes.

In the common approach the audio-visual keys are fused into more reliable decision using a set of simple rules. While such rule-based fusion is convenient when being applied to binary features which can be combined using Boolean and time-ordering operators, it becomes too restrictive when these features are real-valued. The rule-based approach in this case suffers from rigidity of the logic governing the feature fusion. Generally, each feature provides evidence about the presence or absence of a scene boundary with a different level of confidence, depending on its value. Making intermediate decisions, rule-based techniques ignore this difference for one or several features, which leads to undesirable losses of information. Moreover, these techniques require the proper choice of thresholds which usually are the more numerous, the more rules are applied.

As the dependency between the values of audio-visual keys and the optimal segmentation can be hardly established a priori, automatic inferring based on a set of learning data seem to be more appropriate. For this purpose we derive a segmentation approach which fuses multiple evidences in a statistical manner, dealing properly with the variability of each feature. In this approach we consider the segmentation task as detection of segment boundaries by estimating their probabilities and applying time alignment so as to maximize the segmentation performance. The probability of scene boundaries is calculated in different ways depending on basic assumption about the input features, in particular with HMMs. Though we adopt and test our statistical approach to the task of narrative video segmentation into scenes, it is quite general to be applied for other genres of video, possibly having more complex content structure which can include multiple semantic segments at different levels of coarseness. As we could see it from the related work concerned with sports video segmentation, HMMs are quite widespread in this task. The novelty of our approach, when applying HMMs, is in using auto-regressive modeling to deal with feature interdependencies which are crucial in the task of narrative video segmentation into logical story units. Another peculiarity is in using the time alignment that maximizes the segmentation performance instead of the commonly used Viterbi alignment. The advantage of the proposed approach is that it is easily extensible to new features, in contrast to rule-based techniques that often become too complicated and cumbersome when many features are treated. The approach also takes into consideration a non-uniform distribution of scene durations by including it as prior information.

2.3 Conclusions

The related work in the field of semantic video segmentation reveals a large diversity of the processing techniques that stems from different specificity of different genres and sub-genres of video; this is especially the case for sports programs where domain-specific fine-tuned event detectors are often applied. In spite of this, we can notice that much of this work is based on the common idea of using production rules that are followed during creation of video. These rules determine specific syntactical organization which is related to semantic structure of video and, hence, can be exploited to perform automatic video content reconstruction. According to such organization of sports video semantic segments often are represented by the corresponding views or patterns of views. Also, in order to constantly keep the audience informed about the current game state, score or statistics boards are regularly inserted into the broadcast according to the rules of the game. In narrative films logical story units are composed from interleaving visually similar shots, while the corresponding soundtrack exhibits the consistency of the acoustic parameters caused by the common locale and specific editing used to convey the scene mood. In this thesis we rely on such quite common characteristics of video stemming from production rules and propose the corresponding feature detection techniques.

Instead of detection of semantic segments of just one or several types, that is often the case in the related work, in this thesis we aim at reconstructing the total content structure of video. Semantic segmentation of video usually integrates multiple low and mid-level features and is performed using generally two types of methods – deterministic rule-based and stochastic (usually based on HMMs). The advantage of rule-based techniques is that they allow us to express directly our understanding of relationships between detectable features and the semantic structure of a particular type of video, without the need to prepare a large set of manually marked-up learning data. In this thesis we develop a deterministic approach based on a finite state automaton which allows us to formulate video content parsing rules as grammar constraints and feature templates that control transitions between semantic segments. In contrast to the many existing rule-based techniques this approach is formulated on a regular basis and does not require rebuilding the whole content generation system when the underlying content parsing rules are changed. We adopt and test the deterministic approach for the task of tennis video segmentation. Deterministic methods, however, seem not to be appropriate in the case where multiple weak features should be fused into a single reliable decision, e.g. in the task of multi-modal segmentation of narrative video into scenes. Therefore we also derive a segmentation approach which fuses multiple evidences in a statistical manner, dealing properly with the variability of each feature. This approach is adopted and tested for the task of narrative video segmentation into scenes.

3 Deterministic Segmentation

In this chapter we present our deterministic approach which infers a hierarchical content table of video based on mid-level events extracted from raw video. We apply this approach to sports game video, namely to tennis video, as it has a well-defined temporal content structure whose segments can be unambiguously related to mid-level events. As a result, a fully automatic content parsing system is built and tested on a ground-truth video.

3.1 Introduction

Sports video is chosen as being one of the most popular types of the TV broadcasting that appeals large audience. Nowadays, however, we often cannot permit ourselves to spend hours on watching full-time long games such as tennis matches. Moreover, some people might find it boring to watch all the video and they are interested only in the most impressive scenes. This is especially the case if one just wants to refresh in memory some episodes of an already seen game record. As it is difficult to quickly localize an interesting scene in a long video using ordinary media playing tools which provide simple functions like a forward/backward rewind, there is an evident need to provide convenient means of effective navigation. Sports video has usually a well-defined temporal content structure which could be used to efficiently organize a content-based access that allows for such functions as browsing and searching, as well as filtering interesting segments to make compact summaries. As for a tennis match, it can be represented, for example, according to its logic structure as a sequence of sets that in their turn are decomposed into games etc.

To detect regular content units of video we rely on some particular characteristics and production rules that are typically employed to convey semantic information to a viewer. A tennis match, like a lot of other sports games, is usually shot by a number of fixed cameras that yield unique views during each segment. For example, a serve typically begins with switching of the camera into a global court view (see Figure 2-1). Since a tennis match occurs in a specific playground, this view can be detected based on its unique characteristics (we employ its color homogeneity property). In order to constantly keep the audience informed about the current game state, score or statistics boards are regularly inserted into the broadcast according to the rules of the game. In our content parsing technique these inserts are detected and used as indicators of transitions between semantic segments. We propose quite a general framework

which can be considered as a kind of a final state machine whose states relate to content units. It receives at the input a time-ordered sequence of instantaneous events like the beginning of a global view shot and processes it recursively according to pre-defined grammar rules. Some of these events, such as score board appearances are used as transition indicators while others allows for exact positioning of segment boundaries.

This chapter is organized as follows. In the next section we present a general scheme of our parsing system, define tennis content structure and give a detailed description of the parsing technique. After this we describe algorithms developed for automatic detection of the relevant events. In the next section the results of experimental evaluations are presented and discussed. In the section "Application: Tennis Analyzer" we describe our software realization of the proposed segmentation approach for the purpose of automatic content table generation and browsing of tennis video. Final conclusions then finish up the chapter.

3.2 Segmentation Framework

3.2.1 Semantic Structure of Video

We define a content table of video hierarchically as a sequence of nested temporal segments which are contiguous at each semantic level. Different content structures can be usually proposed depending on the needs of a user. An example of two configurations for tennis video is presented in Figure 3-1. It shows segment types allowed at each semantic level; segments of a higher level can comprise segments of several types in the lower level. The first configuration corresponds to the logical structure of a tennis match. According to this structure the match is decomposed into sets separated by breaks at the second semantic level; each set is divided into games and breaks at the third level etc. The second configuration just separates the scenes of tennis rallies ("play") from the rest parts of the video ("break"). Such more simple decomposition allows for building compact summaries consisting only of playing parts and can be used to reduce the duration of the video and the bandwidth for resource limited devices [CHA 01].

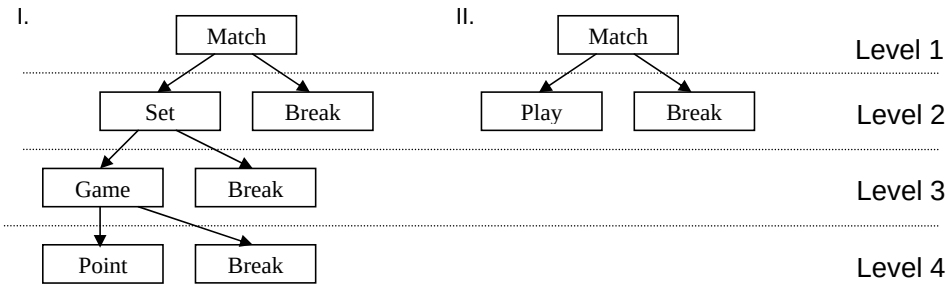


Figure 3-1. Two samples of a tennis video content structure.

3.2.2 Segmentation Principles

If we asked a person to segment video records of the same type, he would need to make up a decision concerning two interrelated problems. First, a desirable semantic structure has to be defined: how many levels of details and what segments can be included at each level. Two possible semantic structures for the tennis video are described above (see Figure 3-1). Second, a set of rules has to be clearly stated that are to be followed in segmenting. If the segmentation is performed only intuitively, without clear understanding of underlying principles, it will be subjective and unstable. The segmentation rules can be usually formulated as events or their combinations which signify transition between semantic segments. It is often the case when these events are suggested by the production principles, which is not surprising as these principles are based on the predefined semantic intention of the producer. For example, the beginning of a game in a tennis match could be recognized by a corresponding score board appearance or by switching to the court view after a pause and change of the serving player.

In order to segment video automatically we state the rules of transition between semantic segments explicitly at each semantic level as combinations or templates of primitive events that can be detected automatically. These templates are defined as sets of events satisfying some temporal constraints. As it was shown by Allen [ALL 83], thirteen relationships are sufficient to describe the relationship between any two intervals: *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, *equals* and their inverses. Additionally we determine relationship "*precedes*" between two point events $s1$ and $s2$ belonging to detectable classes of events $c1$ and $c2$, saying that $s1$ *precedes* $s2$ if $s1$ occurs before $s2$ and there is no other events of type $c1$ and $c2$ between them. Templates can be defined hierarchically so that templates of a higher level are composed from templates of the lower level or primitive events. The templates that determine the transition between semantic segments are referenced hereafter as transition templates. In the general case these templates depend on segment types. Hereafter we suppose that they are dependent only on the type of the segment to which the transition occurs; in the other words each transitional template determines the beginning of the corresponding semantic segment.

To decompose tennis video according to the semantic structure presented in Figure 3-1.I, we propose the following definitions of the semantic segments and the corresponding event templates. Let's suppose that the set of detectable primitive events consists of global court view (denoted as *GCV*) shots (see Figure 2-1), rocket hits (*RH*) sounds and specific score or statistics boards of three types inserted by the producer between tennis points, games and sets respectively. At first we determine the template for the event of tennis serve or rally. When a serve/rally begins, a switch occurs to the camera providing global court view. When it finishes, the view is change so as to show, for instance, players' close-ups or the audience. So, in the

simplest case a serve/rally can be defined as a global court view shot. In order to distinguish a serve/rally from replay shots which correspond sometimes to the same or similar view, rocket hits event can be additionally used. In this case a serve/rally (*SR*) event is defined as a template of two primitive events *GCV* and *RH* related as *RH during GCV*, since rocket hits are not heard during replays. Let's consider the segmentation at semantic level 2 (level of sets) in Figure 3-1.I. We imply that a tennis set begins with its first serve/rally. Therefore the corresponding transition template is the beginning of event *SR*, denoted as *SR.begin* (the beginning of a template is defined as the earliest beginning of its constituent events/templates; the end is defined similarly). A unique score/statistics board is usually inserted a little time after the end of a set which is defined as the end of the last rally. Hence, we detect the beginning of a break as the end of a serve/rally event which *precedes* the beginning of the corresponding score board for sets (*SBS*), i.e. the template is written as {*SR.end precedes SBS.begin*} (the first event in this case is used to precise the beginning time of the break segment). Sometimes score boards stands on the screen all the playing time. In this case transitions to break segments could correspond to the changes of the printed score. The semantic segments and the corresponding templates for semantic level 3 and 4 (level of games and points) are defined in a similar way.

Note that the defined above templates are easily detectable with a computationally effective procedure. If the beginning and the end of detected events or lower-level templates are ordered in time and thus form an input sequence of instantaneous events, these templates can be recognized in one path using state variables for event tracking. For example, the *during* relation of score/rally event is easily checked at the end of a global court view by verifying that the beginning and the end of the rocket hits segment (if they exist) are between the beginning and the end of the global court view.

The general scheme of our parsing system is shown in Figure 3-2. First, relevant semantic events are detected from visual and audio sequences of an input video: score boards, global court views and rocket hits segments. These events are then looked for to distinguish transitional templates that are fed as the input to the content parser. Generally there are some constraints on possible chains of segments at each semantic level that are given by the corresponding grammar. In our case bi-grammars are employed that are sets of allowable transitions between two contiguous segments. A content table is finally generated by the content parser governed by the sequence of transition templates and by predefined grammar constraints.

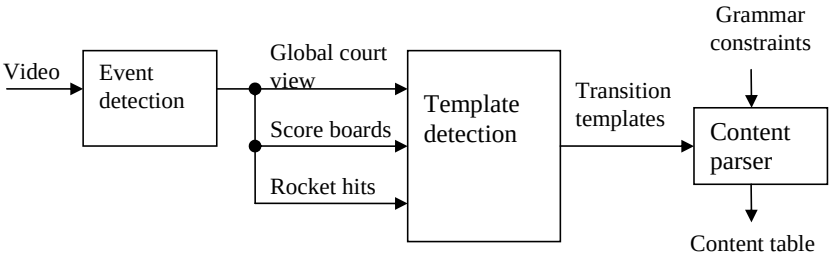


Figure 3-2. Parsing chain.

3.2.3 Segmentation Algorithm

The output content table is generated by a state machine whose states correspond to the appropriate semantic segments. The multilevel content structure of video is generated recursively, beginning at the highest semantic level. At each semantic level the parsing is driven by its grammar that imposes state transition constraints and transition template detectors that control the transition from one state to another. The corresponding parsing rules developed for the content structure of Figure 3-1.I are given in Table 3-1, Table 3-2 and Table 3-3. Column "Transition template" corresponds to the beginning of a state listed in the first column of the tables. The transition time specifies the precise transition moment for the corresponding template. In the general case it is supposed that the initial segment of a given video is unknown. That is why the state machine starts from initial undefined state at the second semantic level. For the lower semantic levels the initial machine state is chosen according to column "Initial state of the sublevel". Our recursive parsing algorithm for a given semantic level is the following:

- Detect transition templates from primary events.
- For each transition template extracted in the time order do:
    - Check whether the template corresponds to an allowed next machine state. If so, do:
        - If the semantic segment corresponding to the current machine state has to be further decomposed into the segments of the lower level, initialize the current state for that level accordingly and perform the parsing recursion for that segment.
        - Go to the next machine state according to the detected pattern.
- For the remaining semantic segment corresponding to the current machine state: if it has to be further decomposed into the segments of the lower level, perform the parsing recursion for this segment.

As it was mentioned above, transition templates can be detected from an input sequence of time ordered point events in one pass. Therefore the two first steps of the algorithm can be merged into one step performed in one pass as well.

State	Allowable next states	Initial state of the sublevel	Transition template	Transition time
Initial undefined	Set	-	-	-
Set	Break	Game	*SR.begin*	*SR.begin*
Break	Set	-	*SR.end precedes SBS.begin*	*SR.end*

Table 3-1. Parsing rules for semantic level 2 (of tennis sets).

State	Allowable next states	Initial state of the sublevel	Transition template	Time adjustment event
Game	Break	Point	*SR.begin*	*SR.begin*
Break	Game	-	*SR.end precedes SBG.begin*	*SR.end*

Table 3-2. Parsing rules for semantic level 3 (of tennis games).

State	Allowable next states	Initial state of the sublevel	Transition template	Time adjustment event
Point	Break	-	*SR.begin*	*SR.begin*
Break	Point	-	*SR.end precedes SBP.begin*	*SR.end*

Table 3-3. Parsing rules for semantic level 4 (of tennis points).

3.3 Event Detection

Our scheme of the automatic tennis video parsing requires a proper choose of events detected in the raw visual and audio streams at the preprocessing stage. The following is a description of algorithms developed for automatic detection of global court views and score boards.

3.3.1 Global Court View

Tennis video like a lot of other types of sport video is usually shot by a fixed number of cameras that give unique views for game segments. A transition from one such view to another is sometimes an important indicator of semantic scene change. In tennis video a transition to a global court view that shows the whole field area with the players commonly signifies that a point starts and a rally begins. When the rally finishes, a transition to another view such as a player close-up or the audience usually happens. Thus, court view recognition is important for rallies scenes detection.

The first step in the detection of a specific view is segmentation of the video into views taken by a single camera or, in the other words, segmentation into shots. Color histogram difference between consecutive frames is applied in order to detect shot transitions. We use 64-bins histograms for each 3 components of the RGB-color space and concatenate them into one 192-dimensional vector. The difference between histograms of two consecutive frames is given by the dissimilarity analogue of the cosine measure:

$$D(H_i, H_j) = 1 - \frac{\sum_k H_i(k) * H_j(k)}{\sqrt{\sum_k [H_i(k)]^2 * \sum_k [H_j(k)]^2}}, \quad (3\text{-}1)$$

where $H_i(k)$ indicates *k*-th bin of the color histogram of frame *i*.

A simple shot detection algorithm puts a shot boundary at a frame for which the difference climbs above some threshold value. It is suitable for abrupt shot transitions that yield strong maxima of the difference value. However, in order to detect gradual transitions we need to set a low threshold value that would lead to unacceptable level of false alarms caused by fast camera motion or a change in lighting conditions. That is why we use a twin-threshold algorithm capable to reliably detect both type of shot transition [DON 01]. Abrupt shot transitions (hard cuts) are detected using a higher threshold $T1$ applied to the histogram difference between two consecutive frames. In order to find a gradual shot boundary, a lower threshold $T2$ is used. If this threshold is exceeded, the cumulative difference is calculated and compared with the threshold $T1$.

In order to exclude false positives of the shot detection algorithm caused by flashlights, additional check is made for abrupt transitions. A flashlight usually changes the color histogram considerably for one or several frames, while the frames that follow right after the flashlight resemble the frames that are before it. We compare the frames lying to the left and to the right of a potential abrupt shot transition within a window T by computing the following value:

$$D_{flash}(t) = \min_{t-T \le i < t, t < j \le t+T} D(H_i, H_j), \quad (3\text{-}2)$$

where t – the time index of the potential shot transition, inter-frame difference D is defined according to expression (3-1). If this value is below a threshold, the shot transition is rejected. We also merge the shot boundaries that are too close to each other (they are usually generated when a gradual shot transition occurs) in order to exclude very short or false shots.

Color distribution of global court view shots does not change much during the tennis match. This allows us to detect them based on their comparison with sample frames of the court view that are selected manually at the learning stage. A shots is recognized as a court view if it is close enough (in the sense of the color histogram difference defined by the expression (3-1)) to

the appropriate sample view. Only homogeneous regions of the tennis field are taken from the learning frames in order to exclude players' figures and outliers. Several court samples and the corresponding rectangular tennis field areas selected at the learning stage of experimental evaluations are shown in Figure 3-3. Each learning sample is selected only once for a game or a series of games played at the same court (e.g. during the same championship).

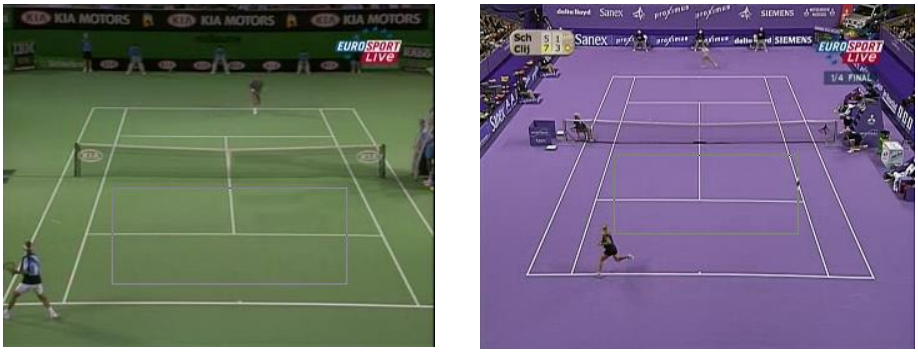


Figure 3-3. Global court view samples where the rectangular regions bounds learning areas.

In tennis video there are usually several types of shots that contain a big part of the tennis field at the background and, thus, resemble much the global court views. An example of such shots is players' close-up views; one such a view is shown in Figure 3-4 along with a court view sample. However, the court views usually take a longer part of the tennis video. Hence, we can enhance the robustness of the court view detection by grouping the shots into similarity clusters and, then, rejecting rare clusters. Let each cluster i be represented by its color histogram (which is an average histogram for all the shots of the cluster) H_i and the number of its shots M_i. In order to describe our clustering algorithm, denote the set of all the clusters as C and the total number of clusters - as N. Then the algorithm can be written as the following.

- Initialize C as an empty set.
- For each shot of the given tennis video do:
  - Calculate a mean histogram of the shot H_{shot}.
  - Find the number k of the cluster closest to the shot as $k = \arg\min_{i=1,\dots,N} D(H_{shot}, H_i)$, where $D(.)$ is the difference measure between the histograms defined by (1).
  - If the distance $D(H_{shot},H_k)$ is less than the threshold $t1$, then set $M_k = M_k + 1$ and $H_k = \frac{M_k - 1}{M_k} H_k + \frac{1}{M_k} H_{shot}$. Else create a new cluster $N+1$ that contains one shot and has the histogram H_{shot}, set $N=N+1$.
- Merge clusters that are close enough to each other.

So, we can resume the global court view detection algorithm as the following.

- Segment the tennis video into the shots.
- Combine visually similar shots into the clusters.
- Calculate the time duration of each cluster for the whole video; exclude from the further consideration the clusters that last less then a predefined fraction (0.2 in this work) of the maximally long cluster.
- Recognize as court views the shots that belong to the cluster closest to the learning court view frames.

Figure 3-4. Player's close-up and court view sample frames that have similar color distributions.

3.3.2 Score Board Detection

As reflecting the state of the game, score boards could provide useful information for tennis video parsing into its logical structure (shown in Figure 3-1.I). Since these boards are inserted regularly according to the game rules, the mere facts of their appearance/disappearance can be used as reliable indicators of the semantic segment boundaries. Moreover, they present important information about the game and, hence, we can choose the appropriate frames as the key frames of the corresponding semantic units and thus provide convenient visual interface for browsing through the content table.

The same tennis video usually has several types of score boards that can be used to separate the segments at different levels of the semantic hierarchy. Score boards of the same type have the fixed positions on the screen and similar color bitmaps near their boundaries. The only difference between them lies in their textual content, the horizontal size (which is changed so as to hold all required data) and somewhat in their color (caused by the partial transparency). Several sample frames which contain score boards along with their bounding rectangle are shown in Figure 3-5 and Figure 3-6. We detect score boards, if we find horizontal lines of enough

length placed near their upper and bottom borders. The Hough transform [HOU 59] is applied to edge points in order to detect the lines. The positions of the score boards borders are given manually during the learning – a user selects from sample tennis video the frames that contain required score tables and picks out their bounding rectangle (see Figure 3-5 and Figure 3-6). In order to enhance the robustness of detection results, smoothing is used – score boards scenes are pronounced only when the corresponding boards are detected in several frames during a period of time.

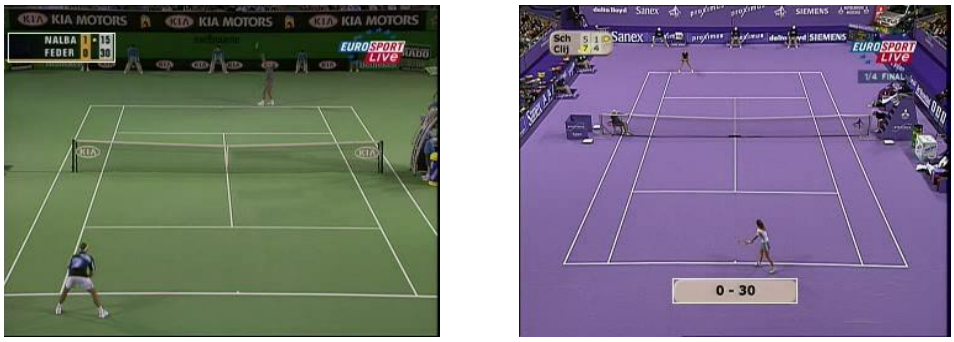


Figure 3-5. Samples of score boards inserted between tennis points and their bounding rectangle.

Figure 3-6. Samples of score boards inserted between tennis games and their bounding rectangle.

3.4 Experimental Evaluations

The performance of our parsing system was experimentally evaluated on three tennis video records captured from Eurosport satellite channel. One of them shows an excerpt of a tennis match of Australia Open (AO) 2003 championship, two others represent fragments of two matches of WTA tournament. The former lasts about 51 minutes, the rest two – 8.5 and 10 minutes. The two tournaments have different score board configuration and color distribution of

the court which can be seen from Figure 3-3, Figure 3-5 and Figure 3-6 representing these tournaments. So, we extracted two sets of learning samples for the events detectors.

In the parsing accuracy evaluations we used the content structure presented in Figure 3-1.I and parsing rules of Table 3-1, Table 3-2 and Table 3-3. Rocket hits detectors were not used in these evaluations, so a template for a score/rally event was represented by a single general court view. Automatically parsed videos were compared with manually labeled data where the segments were defined in the same way as those used to derive the transition templates above in this chapter: the segments "set", "game" and "point" begin with the first serve and end when the last rallies are over (we relate these moments to the beginning and the end of corresponding general court views). The results of segmentation performance evaluations are presented in Table 3-4. Semantic levels 3 and 4 (see Figure 3-1.I) were treated separately; level 2 was not considered as there are few set segments in the ground-truth. The values of recall, precision and F1 are calculated as

$$recall = \frac{N_c}{N_c + N_{miss}}, \tag{3-3}$$

$$precision = \frac{N_c}{N_c + N_{f.a.}}, \tag{3-4}$$

$$F1 = \frac{2 * recall * precision}{recall + precision}, \tag{3-5}$$

where N_c, N_{miss} and $N_{f.a.}$ are the number of correct, missed and false alarm boundaries respectively. A manually labeled boundary was considered as detected correctly if it coincided with an automatically obtained one within an ambiguity time window of 1 second. The value N_b in Table 3-4 stands for the number of tested boundaries in manually labeled video. In order to reduce the influence of "edge effects" on the segmentation evaluations results, the first and the last segments of the lowest semantic level were cut off by half from comparison intervals for each video record. The results of classification accuracy evaluations are given in Table 3-5. The value of recall and precision are computed in a similar way as expressions (3-4) and (3-5), where instead of the number of boundaries the time duration of the segments should be used. The "total duration" of segments in table 3 is measured in seconds.

Tournament	Semantic level	Recall	Precision	F1	N_b
AO	3	0.84	0.62	0.71	19
	4	0.82	0.91	0.86	153
WTA	3	1	0.83	0.91	10
	4	0.94	0.98	0.96	63
AO+WTA	3	0.90	0.68	0.78	29
	4	0.86	0.93	0.89	216

Table 3-4. Segmentation results.

Semantic Level	Segment	Recall	Precision	F1	Total duration
3	Game	0.97	0.99	0.98	3320
	Break	0.97	0.91	0.94	778
4	Point	0.83	0.98	0.90	1670
	Break	0.97	0.89	0.93	1650

Table 3-5. Classification results total for both the tournaments.

As for processing time, our parsing technique is quite fast provided that the events are already extracted and takes less than 1 second for a usual tennis match on modern personal computers. This is because the computational complexity is approximately proportional to the number of events and the number of semantic levels. The major computational power is required to decompress the video and detect the relevant events. On our Intel Pentium 4 1.8 GHz computer this task is performed nearly in real time for MPEG1 coded video, though we did not make a lot of optimizations.

The most of the segmentation errors are caused by unreliability of event detectors. High rate of false score boards result in relatively low precision of segmentation on games and breaks for AO tournament. It is caused by resemblance of the score board, which is a true indicator of the segment transitions, to a statistics board which was inserted in any place during games (sample frames are shown in Figure 3-7). One of the sources of the errors at semantic level 4 is a high false alarm rate for global court views which is caused by confusions with replay shots (they shift the transition between a point and a break). So, there is a need to improve the events detector or use additional ones. For instance, game and set score boards are often shown together with wide views (see the left frame of Figure 3-7). This allow us expect that their combining into a pattern would give a more reliable transition indicator.

Figure 3-7. Game score board (at the left) and its false counterpart.

In order to estimate the accuracy of our parsing engine without the influence of event detection errors, segmentation performance was evaluated on manually corrected events. We considered shots as global court views only if they were not replayed episodes. The evaluation results are given in Table 3-6. There are only few segmentation errors at the semantic level 4 for AO tournament that steam from the parsing rules. They are caused by the fact that sometimes the producer forget to show a score board or insert it after the first serve of a point.

Tournament	Semantic level	Recall	Precision	F1
AO	3	1	1	1
	4	0.91	0.95	0.93
WTA	3	1	1	1
	4	1	1	1
AO+WTA	3	1	1	1
	4	0.94	0.96	0.95

Table 3-6. Segmentation results for manually detected events.

3.5 Application: Tennis Analyzer

A computer program called “Tennis Analyzer” was developed and realized in C++ programming language using MS Visual C++ development environment. It is aimed at completely automatic generation of a content table for tennis video and provides a graphical user interface (GUI) for browsing. The block scheme of the program is depicted in Figure 3-8. Tennis video is given in the form of AVI or MPEG-code file. In order to extract visual and audio features that are to be used for content parsing and browsing through them, tennis video at first is split into a frame sequence and an audio samples stream. The frame sequence is segmented into shots using the twin-threshold method described above. For each shot it is calculated a key frame – the frame that has

the color histogram closest to the mean histogram of the shot. Key frames are used to visually represent the corresponding shots and to classify them into court views. Score boards are detected using the learning board samples extracted from the database which is prepared with the help of the learning module. The learning interface allows a user to select a sample frame with the score board of interest and to define its bounding rectangle. The audio stream is used to detect applauses segments. The applauses are used to generate an importance mark of semantic segments, so that the longer are the applauses, the higher is the mark. At first the audio classifier produces the applauses class probabilities for every sound chunk of one second length. Then, in order to reduce the rate of the false alarms, the smoothing module detects as applauses segments only the groups of several contiguous sound chunks with high probability. As the feature extraction is slow enough, all the features are computed only once and saved to the corresponding data files, whereupon they can be used for fast browsing.

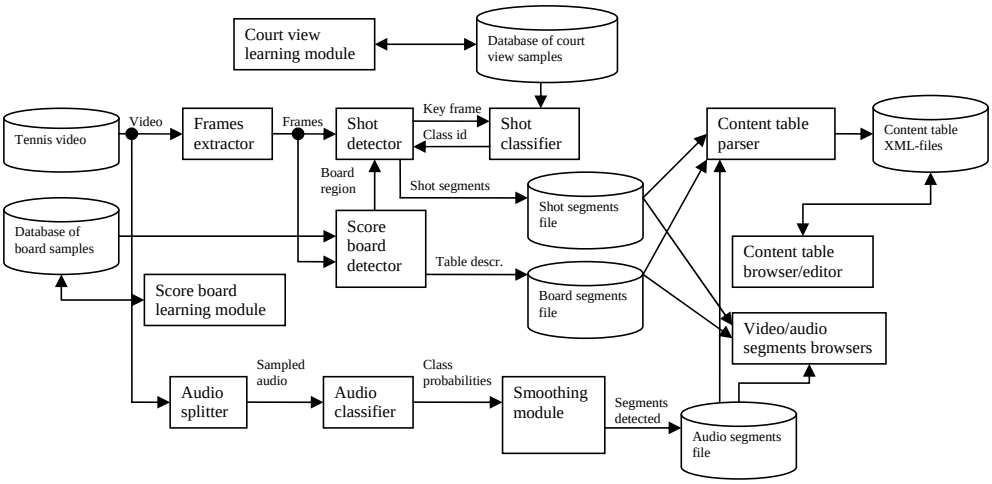


Figure 3-8. Block scheme of the Tennis Analyzer.

The Tennis Analyzer provides several views for tennis video browsing and analyses, as shown in Figure 3-9. The player window (shown at the upper right corner) allows for playing of the video using standard controls: play/stop and rewind buttons and a scrolling slider. The content view (shown at the upper left corner) represents the content table as a tree structure and allows for browsing through the content synchronously with the player window. For each selected semantic segment it represents a list of the nested segments with their attributes. The most interesting segments of the video can be filtered out based on the desirable range for the importance mark. In addition, the content view provides interface for entering the textual description for segments and for manual editing of the content structure that allows a user to correct automatically parsed structures and save them to persistent memory. The view shown at

the bottom of Figure 3-9 represents the key frames of the shots and the frames that contain a score table. It allows for synchronous browsing with the content view and the player window as well – for the content view it can represent only the segment selected in the content tree; the player window can be rewind to any selected key frame by a simple mouse click.

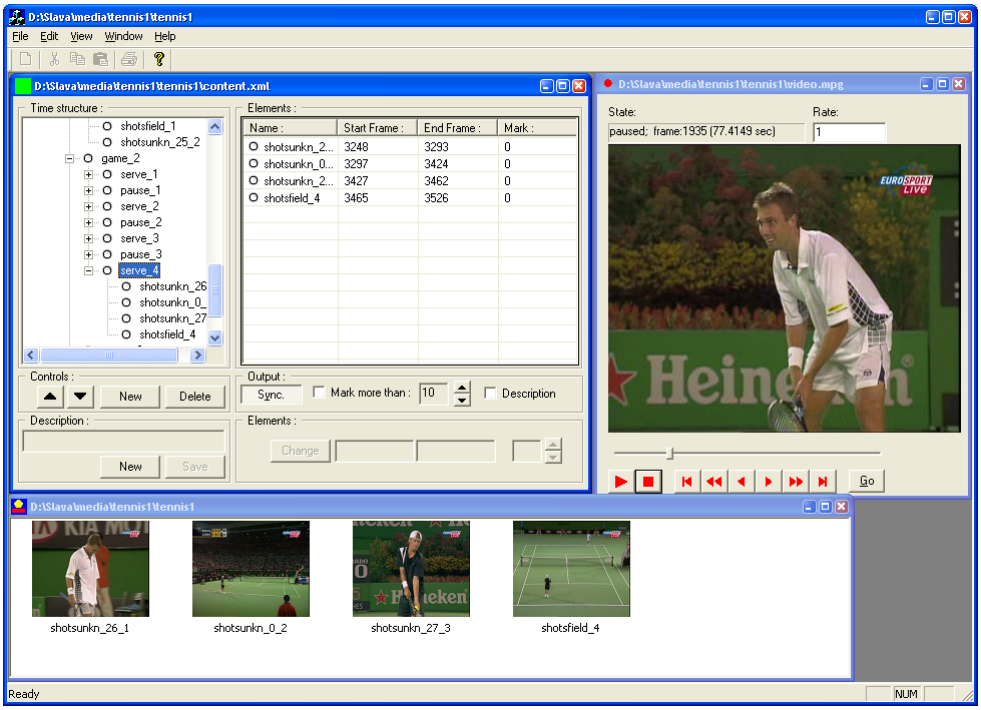


Figure 3-9. Tennis analyzer GUI.

3.6 Conclusions

A deterministic approach is proposed for hierarchical content parsing of video. It is adopted and tested for tennis video. The approach is based on some particular characteristics and production rules that are typically employed to convey semantic information to a viewer, such as specific views and score boards in tennis broadcasts. We use our notion of a tennis content structure to select unique template of events that indicate transitions to semantic segments of each type. These events along with grammar restrictions drive the parsing process.

The advantage of our approach is in its expressiveness and low computational complexity. Moreover, the experimental evaluations showed quite high segmentation accuracy, especially when high reliability of event detectors is provided. Further improvements of the proposed technique could be done in several directions. First, more robust event detectors could be elaborated, as the experimental evaluations showed that such an improvement would enhance

significantly the segmentation accuracy. Second, parsing rules could be extended to include additional informational sources such as rocket hits detection, time constraints, speech recognition. Third, the currently used semantic structure could be extended so as to contain a larger variety of semantics which could provide additional possibilities for content based navigation. For instance, the points could be split into several classes such as rallies, missed first serve, ace or replay.

4 Stochastic Approach

In practical applications it is difficult to find keys which would enable unambiguous segmentation of video. The ambiguity can be caused by the unreliability of the key detection or by the absence of the direct dependency. In the conventional deterministic approach this uncertainty is often ignored or is taken into account very roughly at the expense of the significant growth of system complexity. In this chapter we propose a statistical approach, enabling the keys to be treated in a probabilistic manner. This allows one to take into account "soft" grammar constraints imposed on the semantic structure and expressed in the form of probability distributions. Moreover, the multiple keys, being considered as statistical variables, can be more easily fused into one, more reliable decision in the case of their collisions. Based on the theory of hidden Markov models and their extensions, we consider a video as a stochastic automaton – statistical generalization of the finite state machine, proposed in the previous chapter. This enables us to take into account the correlation between semantic segments at different levels of abstraction (for hierarchical models) and the non-uniform distribution of segment duration.

Further in this chapter we first consider the general principles how to chose the optimal segments based on the corresponding probability estimates. In contrast to the conventional approach which chooses the single best path for the state variables, we focus on the state transitions so as to find the optimal segmentation in terms of recall and precision. Then we consider the video segmentation task based, more specifically, on a hidden Markov model and its extensions.

4.1 Segmentation Principles

4.1.1 Optimality Criterion

We consider video segmentation as detection of segment boundaries at discrete time moments given an input set of features extracted from raw video. These time moments or candidate points of segment boundaries can be chosen in various ways. In the tasks considered in this thesis they are camera shot boundaries since the semantic segments of interest are defined as groups of shots. Alternatively the candidate points might be determined by the boundaries of mid-level events or simply chosen at discrete times regularly spaced with an interval providing acceptable temporal resolution.

To indicate the absence or presence of a segment boundary at time index t we use a binary variable $s_t \in \{0,1\}$. So, the aim of segmentation of a video is to find an optimal

sequence $s \equiv \{s_1, s_2, ..., s_T\}$, where T is the number of candidate points within the video. If segments differ by their semantic meaning, we should also provide semantic labels $\{p_t, f_t\}$ of contiguous segments adjacent to each segment boundary at time *t*, where *p* is the type of the preceding segment and *f* – the type of the following one. Let's denote the sequence of *N* time indexes corresponding to scene boundaries as $b \equiv \{b_1, b_2, ..., b_N\}$. As each segment must have the same semantic label at the ends, the following constraints are imposed:

$$f_{b_i} = p_{b_{i+1}}, \forall i = 1,2,..., N-1. \quad (4\text{-}1)$$

In the general case of hierarchical content structure semantic segments are identified by their type defined at the current semantic level and by the type of the corresponding higher-level segments. For example, a break between tennis games according to Figure 3-1.I is represented as a pair {set, break}. We suppose in this case that all these nested identifiers for each segment are enumerated into one label.

To deal properly with the uncertainty of real observable data, we consider the segmentation task in a probabilistic manner by modeling the video as a stochastic process. The task is, then, to find optimal values of random variables s_t at each time index *t* as well as the corresponding segment labels given a set of observable data generated according to a probabilistic law. But what criterion of optimality should be used? The common approach is to find the most probable sequence of appropriate state variables related to an input video. In boundary-based segmentation methods, when semantic labels of segments are not of interest, these variables are our binary indicators of segment boundaries *s*, as it is the case for story segmentation in [HSU 03, HSU 04]. Alternatively, in segment-based segmentation methods, the temporal dynamics within segments are modeled by a sequence of states, often using hidden Markov models (HMM). For example, TV news broadcasts are segmented into story units in [LEK 02] using a four-states ergodic HMM; in [EIC 99] logical units of news programs are segmented and classified into six main types where each unit type is represented with a HMM. The most probable sequence of states is computed using computationally effective procedures based on dynamic programming, such as a Viterbi algorithm.

Let's consider this approach from the perspective of the measures used to numerically evaluate the segmentation performance. Recall and precision frequently serve as such measures. They are widespread in information retrieval [RIJ 79, LEW 91] and are standard in story segmentation [GUI 04]. The performance measures are obtained by comparing the actual and claimed segment boundaries of the same video. This is illustrated in Figure 4-1 where the chain of actual segments is represented by the upper stripe and that of claimed ones – by the lower; different segment types are encoded by different color. An actual boundary is defined to be

detected if there is at least one claimed boundary which lies in the vicinity measured by a temporal ambiguity τ and both the boundaries separate the segments of the same type. Otherwise the actual boundary is defined as a miss. Similarly a claimed boundary is defined as a correct one if there is at least one actual boundary within the limits of the ambiguity τ (which is assumed to be the same as for actual boundaries) and both the boundaries separate the segments of the same type. Otherwise the claimed boundary is defined as a false alarm. In fact, an ambiguity window 2τ (see Figure 4-1) is considered around each actual boundary – if one or several claimed boundaries, which separate the same segments, fall into this window, the corresponding boundaries are defined to be detected and correct (similarly we could place the ambiguity window around each claimed boundary, as the ambiguity time is the same for the actual and claimed segments). If the time interval between two consecutive claimed boundaries is less than 2τ, then it is possible that they are both correct and correspond to the same actual boundary and vice versa. Therefore the number of correct and detected boundaries is not generally the same.

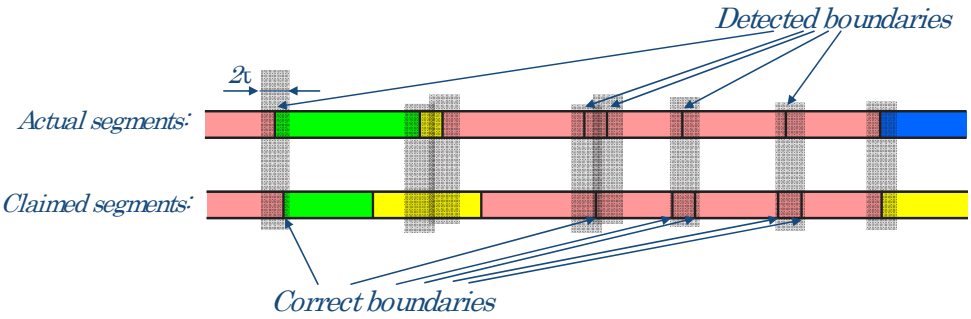


Figure 4-1. Comparison of segment boundaries.

Recall and precision measure the proportion of actual segment boundaries detected and the proportion of correct claimed segment boundaries respectively. Denoting the number of actual boundaries detected as $N_{a.d.}$, the number of correct clamed boundaries – as $N_{c.c.}$, the number of false alarms – as $N_{f.a.}$, the number of misses – as N_m, recall r and precision p are written as:

$$r = \frac{N_{a.d.}}{N_{a.d.} + N_m}, \tag{4-2}$$

$$p = \frac{N_{c.c.}}{N_{c.c.} + N_{f.a.}}. \tag{4-3}$$

System performance measured by recall and precision focuses on time indexes corresponding to segment boundaries. Thus there is no need to take into account all the candidate points at the same time, like in the methods where the most probable sequence of states is found

for the whole video. Moreover, in the most cases the moments of absence of segment boundaries are predominant, and the minor points of segment boundaries become negligible when optimizing the whole state sequence. This can deteriorate considerably the segmentation performance. Consider, for example, the situation where a segment boundary can be surely detected in a time range covering several candidate points, but the probability to find this boundary at each single point is quite low. Segmentation through finding the most probable state path for the whole video is likely to ignore the boundary, resulting in increase of number of misses and, hence, low recall.

Commentaire [LC1]: Ça se discute !!!

In this thesis we derive the optimal decision rule for the segment boundary detection based on recall and precision which are chosen to measure the system performance. Let's suppose that a fixed number N of distinct candidate points are claimed as segment boundaries and the total number of actual boundaries is N_a. It is not difficult to see that the denominator in expression (4-2) and (4-3) is equal to N_a and N respectively. Hence, in order to maximize recall and precision, N claimed boundary should be selected so that to provide the maximum values for $N_{a.d.}$ and $N_{c.c.}$. This minimizes the number of false alarms and the number of misses written as

$$N_{f.a.} = N - N_{c.c.}, \tag{4-4}$$

$$N_m = N_a - N_{a.d.}. \tag{4-5}$$

Let's further assume that segments cannot be of zero duration and that the coincidence between a claimed boundary and an actual one (allowing us to consider the claimed boundary to be correct and the actual one to be detected) is established only in the case where these boundaries occur exactly at one time (i.e. the time ambiguity τ is zero). Under these assumptions each correct claimed boundary correspond to one and only one actual boundary detected and, hence,

$$N_{a.d.} = N_{c.c.} \tag{4-6}$$

which is the only value to be maximized.

Let's now derive an expression for $N_{c.c}$. To distinguish the claimed (computed) segment boundaries the actual ones, we use a tilde. Thus, the result of computed segmentation for an input video is denoted as a sequence of tuples $\{\tilde{s}_t, \tilde{p}_t, \tilde{f}_t\}$ while the actual subdivision into segments is represented as $\{s_t, p_t, f_t\}$ where, as earlier, $s \in \{0,1\}$ is an indicator of the presence (s=1) or absence (s=0) of segment boundary, p and f – the labels of segments preceding and following the point under consideration, t – a time index. Then, since each claimed segment boundary b_i is considered to be correct if it coincides with an actual one, $N_{c.c}$ is written as

$$N_{c.c.} = \sum_{i=1}^{N} \delta(s_{b_i} - 1, p_{b_i} - \tilde{p}_{b_i}, f_{b_i} - \tilde{f}_{b_i}), \tag{4-7}$$

where the discrete delta function δ is defined for three arbitrary variables x, y, z as

$$\delta(x,y,z)=\begin{cases}1, \textit{if } x=0, y=0, z=0\\ 0 \textit{ otherwise}\end{cases} \tag{4-8}$$

As an input video is modeled as a stochastic process, $N_{c.c}$ is a random variable, and we consider its expected value instead:

$$E\{N_{c.c.}\}=\sum_{i=1}^{N} E\{\delta(s_{b_i}-1, p_{b_i}-\tilde{p}_{b_i}, f_{b_i}-\tilde{f}_{b_i})\}=\sum_{i=1}^{N} P(s_{b_i}=1, p_{b_i}=\tilde{p}_{b_i}, f_{b_i}=\tilde{f}_{b_i}), \tag{4-9}$$

where $P(s_i=1, p_i, f_i)$ denotes the posterior probability of the presence of a boundary between segments p_i and f_i at candidate point i.

Hence, assuming that the probability $P(s_t=1, p_t, f_t)$ of segment boundary is pre-calculated for each candidate point t and each segment labels pair $\{p_t, f_t\}$, *the optimal segmentation selects N segment boundaries so as to maximize the rightmost sum of expression (4-9)*. The more is N, the more points of low probability are generally selected and, hence, the less is the relative expected number of correct boundaries among them. On the other hand, the value N should be high enough to provide an acceptable level of misses. So, this value controls the trade-off between the number of false alarms and the number of misses and, hence, between precision and recall.

N can be chosen so as to provide the maximum of an integral performance measure. In this thesis it is a $F1$ measure which is a harmonic mean of recall and precision:

$$F1=\frac{2*r*p}{r+p}. \tag{4-10}$$

As it follows from experimental evaluations, F1 has a maximum when recall and precision are approximately equal. From expression (4-4) - (4-6) follows that equal recall and precision are provided when $N=N_a$, or, as N_a is considered as a statistical variable, N is selected as expected number of N_a:

$$N=E\{N_a\}. \tag{4-11}$$

By analogy with expression (4-9) the expected number of N_a is calculated as:

$$E\{N_a\}=\sum_{i=1}^{T} E\{\delta(s_i-1)\}=\sum_{i=1}^{T} P(s_i=1). \tag{4-12}$$

4.1.2 Computing Optimal Segment Boundaries

According to our optimal decision rule for segmentation we wish to select N segment boundaries so as to maximize expression (4-9). A straightforward exhaustive search over all possible boundary arrangements has an exponential computational complexity on N and thus is unfeasible in most cases. A simple and computationally effective algorithm can be proposed in the

particular case where the segments are not labeled. In this case the only input data are a sequence of segment boundary probabilities $\{P_1, P_2, ..., P_T\}$, where $P_i \equiv P(s_i = 1)$. N maximal values can be selected by scanning this sequence and extracting the maximal value N times, which yields the computational complexity on the order of $N \cdot T$. Alternatively, the sequence can be sorted in descending order of probability and N first values be related to segment boundaries, which yields the complexity on the order of $T \log(T)$ required for sequence sorting.

In the general case, where the segments are distinguished by their label, segment boundaries cannot be selected independently from each other because of constraints of expression (4-1) imposed on segment labels. To attain feasible computational complexity in this case, we propose the following procedure. Omitting variable s we denote the probability of transition from segment p_t to a segment f_t at time moment t as $P(p_t, f_t)$. Given this probability for each time point $t = 1, ..., T$ and for each pair of segment labels, the task is to select N distinct segment boundaries $\{b_1, b_2, ..., b_N\}$ and the corresponding segment labels p_{b_i} and f_{b_i} so as to maximize the sum

$$\sum_{i=1}^{N} P(p_{b_i}, f_{b_i}) \tag{4-13}$$

taking into account the constraints of expression (4-1). We define the following variable:

$$M(n, f, t) \equiv \max_{\substack{b_1,...,b_n \\ p_{b_1},...,p_{b_n} \\ f_{b_1},...,f_{b_{n-1}}}} \{\sum_{i=1}^{n-1} P(p_{b_i}, f_{b_i}) + P(p_{b_n}, f)\}, \tag{4-14}$$

where it is assumed that $1 \le b_1 < ... < b_n \le t$ and expression (4-1) holds true. $M(n, f, t)$ is the best score of expression (4-13) corresponding to n segment boundaries selected for first t candidate points given that the last segment is labeled as f. By induction we have

$$M(n, f, t) = \max_{1 \le f' \le m, 1 \le t' < t} \{M(n-1, f', t') + \max_{t' < b_n \le t} [P(p_{b_n} = f', f_{b_n} = f)]\}, \tag{4-15}$$

where m denotes the number of segment labels. To actually retrieve the sequence of optimal segment boundaries, we need to keep track of arguments which maximized expression (4-15). We do this via the arrays $L(n, f, t)$ and $B(n, f, t)$. The complete procedure for finding the best segment boundaries can now be stated as follows:

1) Initializaton:

$$M(1, f, t) = \max_{\substack{1 \le b_1 \le t \\ 1 \le p_{b_1} \le m}} \{P(p_{b_1}, f_{b_1} = f)\}, \; 1 \le f \le m, 1 \le t \le T \tag{4-16}$$

$$L(1, f, t) = \arg\max_{1 \le p_{b_1} \le m} \max_{1 \le b_1 \le t} \{P(p_{b_1}, f_{b_1} = f)\}, \; 1 \le f \le m, 1 \le t \le T \tag{4-17}$$

$$B(1,f,t)=\arg\max_{1\le b_1\le t}\max_{1\le p_{b_1}\le m}\{P(p_{b_1},f_{b_1}=f)\},\ 1\le f\le m,1\le t\le T \tag{4-18}$$

2) Recursion:

$$M(n,f,t)=\max_{1\le f'\le m,n-1\le t'<t}\{M(n-1,f',t')+\max_{t'<b_n\le t}[P(p_{b_n}=f',f_{b_n}=f)]\}, \tag{4-19}$$

$$L(n,f,t)=\arg\max_{1\le f'\le m}\max_{n-1\le t'<t}\{M(n-1,f',t')+\max_{t'<b_n\le t}[P(p_{b_n}=f',f_{b_n}=f)]\}, \tag{4-20}$$

$$B(n,f,t)=\arg\max_{t'<b_n\le t}\max_{1\le f'\le m,n-1\le t'<t}\{M(n-1,f',t')+P(p_{b_n}=f',f_{b_n}=f)\}, \tag{4-21}$$

$$2\le n<N,1\le f\le m,n\le t\le T.$$

3) Termination:

$$\{b_N,p_{b_N},f_{b_N}\}=\arg\max_{t<b_N\le T,1\le p_{b_N}\le m,1\le f_{b_N}\le m}\max_{N-1\le t\le T}\{M(N-1,p_{b_N},t)+P(p_{b_N},f_{b_N})\}. \tag{4-22}$$

4) Segment boundaries backtracking:

$$b_n=B(n,f_{b_{n+1}},b_{n+1}), \tag{4-23}$$

$$p_{b_n}=L(n,f_{b_{n+1}},b_{n+1}), \tag{4-24}$$

$$f_{b_n}=p_{b_{n+1}}, \tag{4-25}$$

$$n=N-1,N-2,...,1.$$

As calculation $M(n,f,t)$ requires on the order of $m\cdot T^2$ operations for each possible triple $\{n,f,t\}$, the resulting computational complexity of the procedure is on the order of $m^2N\cdot T^3$.

4.1.3 Ambiguity of Segment Boundary Position

In practical applications segmentation performance measures tolerate some temporal ambiguity τ between detected and actual boundaries when deciding whether there is correspondence between them [GUI 04]. Taking into account this ambiguity allows us to detect boundaries more reliably. In this subsection we propose a required extension to our optimal segmentation rule. For the purpose of simplicity we suppose hereafter in this subsection that labels of segments are not of interest and consider only their positions.

A typical value of τ is about 5 sec [GUI 04] which is normally less than segment length. We assume that segments cannot be shorter than 2τ. In this case it is not possible that two or more actual boundaries correspond to one claimed boundary. As so, if we wish to minimize the number of misses for a fixed number of claimed boundaries, these boundaries should be placed no closer than 2τ from each other as this provides the maximum number of potential correspondences. Several claimed boundaries, however, can still correspond to one actual

boundary. This can be used to "artificially" augment precision by claiming several boundaries in the vicinity of highly probable actual ones where the probability of false alarms is low. That is why we propose a stricter criterion of one-to-one correspondences between claimed and actual boundaries. The maximal number of these correspondences is the number of correct claimed boundaries $N_{c.c.}$ and the number of actual boundaries detected $N_{a.d.}$. As it was earlier, expression (4-6) holds true and our task is to select N boundaries so at to maximize $N_{c.c.}$. According to the stricter correspondence criterion these boundaries must be spaced no closer to each other than 2τ to minimize the number of misses and false alarms at the same time.

Given an input sequence of segment boundary probabilities $\{P_1, P_2, ..., P_T\}$ let's derive an optimal segmentation rule. Denote as G_i the set of candidate points lying in the vicinity $[t_i - \tau, t_i + \tau]$ of an arbitrary candidate point i occurring at time t_i. Under our assumption only one actual boundary can be found in this region. Hence, the probability of a single claimed boundary placed at point i to be correct is written as

$$P(c(i) = 1) = \sum_{j \in G_i} P_j \ , \tag{4-26}$$

where $c(i) \in \{0,1\}$ is indicator function which is equal to 1 when a boundary claimed at point i is correct and 0 otherwise. Since claimed boundaries are not closer to each other than 2τ and, hence, their corresponding regions G are not overlapped, the expected number of correct boundaries $N_{c.c.}$ is calculated as

$$E\{N_{c.c.}\} = E\{\sum_{i=1}^{N} \delta(c(b_i) - 1)\} = \sum_{i=1}^{N} P(c(b_i) = 1) = \sum_{i=1}^{N} \sum_{j \in G_i} P_j \ . \tag{4-27}$$

Optimal segment boundaries are chosen so as to maximize expression (4-27). We propose to do this iteratively. At each iteration step the sum of expression (4-26) is computed at each candidate point. The point i with the maximal sum is claimed then as a segment boundary and the points in G_i are excluded from the further analysis.

4.2 Hidden Markov Models

To obtain estimates of segment boundary probability which are required by our optimal segmentation rules considered above, we need to properly choose a model describing an input video. Hidden Markov models (HMM) are powerful tools for modeling the dynamics of different processes evolving in time, such as video [DIM 00, HUA 99, BOR 98] and speech signals [RAB 89, BEN 99]. In this section we provide basic definition and assumptions that underlying these models, consider their different variations suitable for the purpose of video modeling and derive expressions required for segmentation.

4.2.1 Basic Model

A basic HMM is a stochastic process which at any discrete time $t = 1,2,...,T$ is at one of a set of N distinct states $Q^* = \{1,2,...,N\}$. We denote the actual state at time t as $q_t \in Q^*$. The dynamics of the process is then described as a sequence $Q = \{q_1, q_2, ..., q_T\}$. At each time moment the model changes the state (or remains at the same state) according to the probability values associated with the state. A complete probabilistic description of a stochastic process requires specification of the current state depending on all the predecessor states. The HMM is defined as the special case of a first order Markov chain, where the probability to be in the current state q_t is determined completely by the predecessor state, i.e.

$$P(q_t = j \mid q_{t-1} = i, q_{t-2}, ...) = P(q_t = j \mid q_{t-1} = i) \equiv a_{ij}, \tag{4-28}$$

where a_{ij} denotes the probability of transition from state i to state j. It is supposed that the HMM is stationary and, hence, a_{ij} is independent on the time index. The initial state is chosen according to the probability denoted as

$$\pi_i \equiv P(q_1 = i). \tag{4-29}$$

We collect all state transition probabilities into one matrix $A = \{a_{ij}\}$ which satisfies the following stochastic constraints:

$$a_{ij} \geq 0,\ 1 \leq i \leq N,\ 1 \leq j \leq N, \tag{4-30}$$

$$\sum_{j=1}^{N} a_{ij} = 1,\ 1 \leq i \leq N. \tag{4-31}$$

Depending on applications, additional constraints can be imposed to matrix A. Forcing some coefficients to be zero we can forbid the corresponding transitions. Thus, different topologies can be defined that are usually depicted graphically so that the allowed state transitions are shown by arrows. One such model is presented in Figure 4-2. This is a left-right or Bakis model [BAK 76], for which low numbered states can only make transitions to higher number states or to themselves, i.e. $a_{ij} = 0$ for each $j < i$. This model is suitable for processes whose properties change over time, such as speech signals. If every state of the HMM could be reached from every other state in a single step, the corresponding topology includes all possible connections and is called an ergodic or circular model.

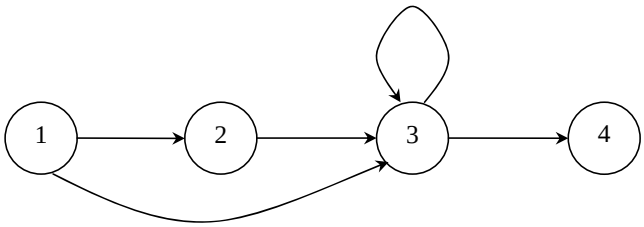


Figure 4-2. A 4-state left-right HMM.

The states of the HMM are not observable directly (i.e. "hidden") but generate a vector of measurable features according to a probabilistic function. We denote the feature vector observed at time t as D_t. It is assumed that this vector is conditioned only on the current state. We denote the corresponding probability distributions as $B = \{b_j(D_t)\}$, where

$$b_j(D_t) = P(D_t \mid q_t = j),\ 1 \le j \le N. \tag{4-32}$$

The presented above HMM describes double stochastic process. The primary process is not observable, or is hidden, and is determined as a first order Markov chain. The secondary process $D = \{D_1, D_2, ..., D_T\}$ is an observable representation of the primary process generated according to a probabilistic rule. The joint description of these two processes is given by defining the matrix of initial state probabilities $\Pi = \{\pi_i\}$, matrix of transition probabilities A and probability distributions for generating observations B. This description is a complete specification of a basic HMM.

A widespread approach to the task of video segmentation is to model an input video with a single HMM. The states of the HMM are stationary parts of the video, such as frames or camera shots. Semantic segments are then related to subsequences of the states. The HMM can be thought as an opaque box, where the sequence of features D is observable, while the sequence of the states is hidden. In the simplest case each segment is assigned to a unique state. For example, two different HMM topologies – a two-states ergodic and a left-right one (see Figure 4-3) – are explored in [ALA 01]. The aim is to separate dialog scenes from non-dialog scenes in movies. The elementary time units in this example are camera shots and state transitions are explored at shot change moments. The limitation of such an approach is that it is not general enough to separate several contiguous semantic segments of the same type. In the more general case each segment is represented by a sequence consisting of different HMM states. For example, news video is divided into story units of the same type in [LEK 02] using a four-states ergodic HMM. This model allows the authors to track dynamic patterns of shots corresponding to news stories.

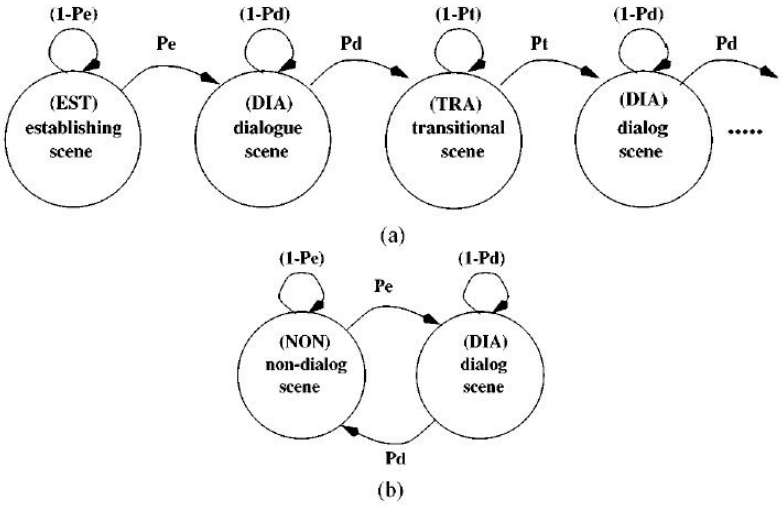


Figure 4-3. Left-right (a) and circular (b) HMM for modeling dialog scenes in movies [ALA 01].

Semantic segments are commonly detected through reconstructing the full sequence of the HMM states. If each segment is represented by a unique state, then the resulting segments are the corresponding groups of repetitive state labels. If segments are modeled as subsequences of states of several types, then segment boundaries are found as transition to or from unique states which begin or terminate the corresponding subsequences. The common criterion used to find the best sequence of HMM states is maximizing the posterior probability of the sequence $P(Q \mid D)$ which is equivalent to maximizing the joint probability $P(Q, D)$. This probability is written as

$$P(Q,D) = P(Q)P(D \mid Q) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T} \prod_{t=1}^{T} b_{q_t}(D_t). \tag{4-33}$$

The straightforward maximization of this expression using full search over all possible state sequences requires on the order of $2TN^T$ operations which is infeasible for the most applications. Fortunately, there exists a computationally effective technique for finding this best state sequence, based on dynamic programming, and it is called the Viterbi algorithm [VIT 67].

To write down the Viterbi algorithm, let's first define the following variable:

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P(q_1, q_2, ..., q_t = i, D_1, D_2, ..., D_t). \tag{4-34}$$

This variable is the highest probability for the first $t-1$ states. It allows one to find the probability of the whole optimal path recursively using the following rule:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(D_{t+1}). \tag{4-35}$$

In addition, we define for each t and j the variable $\psi_t(j)$ which is the argument maximizing expression (4-35). This variable is needed to retrieve the best state sequence after the maximum probability of the whole state sequence has been found. Denoting as $\tilde{P}$ the optimal value for the probability and as $\tilde{Q} = \{\tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_T\}$ the optimal state sequence, the Viterbi algorithm is resumed as follows:

1) Initialization:

$$\delta_1(i) = \pi_i b_i(D_1),\ \psi_1(i) = 0,\ 1 \le i \le N \tag{4-36}$$

2) Recursion:

$$\delta_{t+1}(j) = \max_{1 \le i \le N}\{\delta_t(i) a_{ij}\} b_j(D_{t+1}), \tag{4-37}$$

$$\psi_{t+1}(j) = \arg\max_{1 \le i \le N}\{\delta_t(i) a_{ij}\}, \tag{4-38}$$

$$1 \le t < T,\ 1 \le j \le N.$$

3) Termination:

$$\tilde{P} = \max_{1 \le i \le N}\{\delta_T(i)\}, \tag{4-39}$$

$$\tilde{q}_T = \arg\max_{1 \le i \le N}\{\delta_T(i)\}. \tag{4-40}$$

4) State sequence backtracking:

$$\tilde{q}_t = \psi_{t+1}(\tilde{q}_{t+1}),\ t = T-1, T-2, ..., 1. \tag{4-41}$$

It is easy to see that the computational complexity of the Viterbi algorithm is on the order of $N^2 \cdot T$.

As it was discussed above in this chapter, segmentation via reconstruction of complete state sequence does not necessarily lead to the optimal system performance. To find the optimal segment boundaries according to our optimality criterion, we need to estimate the posterior probability of segment boundaries at each candidate point. For this purpose we first define $\xi_t(i, j)$, the probability of transition from state i at time t to state j at time $t+1$, given the observation D:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j \mid D). \tag{4-42}$$

For computationally effective calculation of this value we use the forward-backward procedure [RAB 89] as follows. Consider the forward variable $\alpha_t(i)$ defined as the probability of the partial observation sequence until time t and state i at time t:

$$\alpha_t(i) \equiv P(D_1, D_2, ..., D_t, q_t = i). \tag{4-43}$$

This variable is calculated by induction as

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(D_{t+1}),\ 1 \le t < T,\ 1 \le j \le N, \tag{4-44}$$

where initial value is

$$\alpha_1(i) = \pi_i b_i(D_1),\ 1 \le i \le N. \tag{4-45}$$

In a similar manner a backward variable $\beta_t(i)$ is defined as

$$\beta_t(i) \equiv P(D_{t+1}, D_{t+2}, ..., D_T, q_t = i). \tag{4-46}$$

Initialized with

$$\beta_T(i) = 1,\ 1 \le i \le N, \tag{4-47}$$

it is calculated by the following induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(D_{t+1}) \beta_{t+1}(j),\ t = T-1, T-2, ..., 1\ ,\ 1 \le i \le N. \tag{4-48}$$

After applying the forward-backward procedure, variable $\xi_t(i, j)$ is calculated as

$$\xi_t(i, j) = \frac{P(q_t = i, q_{t+1} = j)}{P(D)} = \frac{\alpha_t(i) a_{ij} b_j(D_{t+1}) \beta_{t+1}(j)}{P(D)}, \tag{4-49}$$

where $P(D)$ can be calculated, for instance, as

$$P(D) = \sum_{i=1}^{N} \alpha_T(i). \tag{4-50}$$

Segment boundaries are related to transitions between the HMM states. Hence, the candidate points of these boundaries are $T-1$ potential transitions within the sequence of T states. If a segment boundary corresponds to a single pair of states i and j, as for instance in the case where each segment is represented by one state, then its posterior probability is $\xi_t(i, j)$. In the general case segments are modeled by subsequences consisting of different states. To separate these subsequences, one could mark their beginning or the end with a special state or model segments with non-overlapping sets of states. Let's denote the set of states which can end an arbitrary segment $s1$ as G_1 and the set of states which can begin an arbitrary set $s2$ – as G_2. Then the probability that a boundary between segments $s1$ to $s2$ corresponds to the transition between states q_t and q_{t+1} is computed as

$$\sum_{i \in G_1} \sum_{j \in G_2} \xi_t(i, j). \tag{4-51}$$

4.2.2 Hierarchical Model

The content of video is often organized in a hierarchical manner, e.g. a tennis match can be divided first into sets, then the sets are decomposed into games etc. In this subsection we present a generalization of the basic HMM, called a hierarchical HMM (HHMM) [SHA 98], which

models this organization directly. These models have found a wide use in many domains of application with hierarchical structure, such as image and video segmentation [PHU 05, ZHE 04], visual action recognition [NGU 05, MOO 01, HOE 01], spatial navigation [BUI 01, THE 01] and handwriting recognition [SHA 98]. The advantage of HHMMs is that they take into account statistical dependences existing between structural elements at multiple levels of coarseness, thus enabling to model long-term correlations between observable feature vectors.

A HHMM is a structured process defined as a Markov chain whose states are hidden and modeled with their proper lower-level Markov chains. At the lowest level of the hierarchy this process is an ordinary HMM, whose states generate observable feature vectors according to a probabilistic rule. The states of higher levels aggregate the lower-level state chains. Therefore they generally correspond to sequences of feature vectors. These sequences are generated in a recursive manner by activation the corresponding sub-models which may be composed of sub-models as well. This process terminates when states of the lowest-level are reached. The lowest-level states are called *production states* as they are the only states which emit observable data. The states of the higher-levels do not generate observable features directly and are called *internal* or *abstract states*.

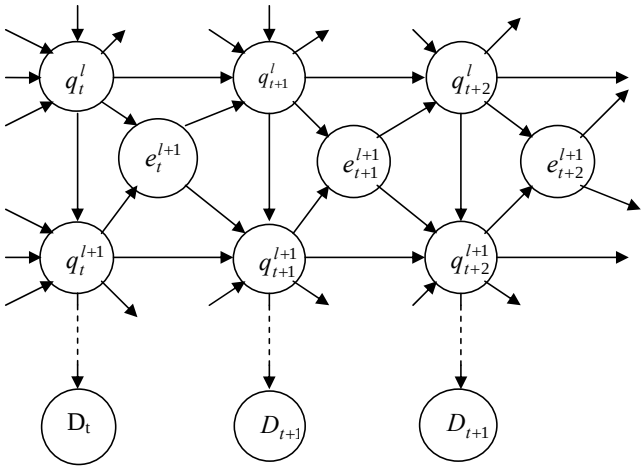


Figure 4-4. DBN representation of a HHMM at level l and $l+1$ at time t, t+1, t+2. q_t^l denotes the state at time t, level l; e_t^l is an indicator variable that the HMM at level l has finished at time t; D_t is the observable feature vector.

A HHMM can be graphically represented as a dynamic Bayesian network (DBN) [MUR 01], as shown in Figure 4-4. The state of the model at level l and time t is denoted as q_t^l. When the model enters the abstract state, the corresponding sub-model is activated in a recursive manner. This activation is called a *vertical transition*. When the sub-model is finished (which

may engender activations of lower level states recursively), the control returns to the upper-level state it was called from. Then a state transition within the same level, called a *horizontal transition*, occurs. A sub-model finishes when a special *end* state is reached. This state never emits observable data and immediately engenders the transition to the calling state. To indicate that the sub-model is about to enter the end state, the corresponding indicator variable of the DBN representation $e_t^l \in \{0,1\}$ is set to 1, otherwise it is equal to 0.

The calling context of vertical transitions is stored in a depth-limited stack. Any HHMM can be converted to an ordinary HMM by enumerating all possible states in the stack, from the highest model level up to the lowest one. Assuming that the HHMM has L levels and that all production states are at the lowest level L, the states of the equivalent HMM are encoded by mapping the calling context $q_t^{1:L} = \{q_t^1,...,q_t^L\}$ of each production state into integers. The same sub-model of the HHMM can be shared by several sub-models of the upper level. In the HMM representation this shared sub-model must be duplicated for each calling context, which generally results in a larger model. So, the power of the HHMMs is in the ability to reuse its substructures. As a result, they have a more compact representation, and the less number of parameters simplifies their learning. The hierarchical representation of HHMMs also allows us to specify their topology or constraints on possible state transition in a more natural way. Consider, for example, the HHMM topology for a tennis video shown in Figure 2-5 (note that the underlying semantic structure of the video is slightly different from that used in our deterministic segmentation approach described above in this work, e.g. points are not separated by break segments). Using a chain of sub-models allows the authors to impose a constraint on the minimum number of the corresponding semantic segments, e.g. a game segment consists of no less than 4 points. At the same time, these sub-models are not duplicated superfluously.

In order to give a strict formal definition of the HHMM, let's specify conditional probability distributions of each node type in the corresponding DBN representation (see Figure 4-4). Consider first the lowest level L of the model. The states of this level follow the rules of a regular HMM, whose parameters are determined by its position in the HHMM encoded by the vector of higher state variables $q_t^{1:L-1} = \{q_t^1,...,q_t^{L-1}\}$. For simplicity of notations we represent this vector by the integer k. When the HMM is activated, its initial state j is selected according to the prior distribution $\pi_k^L(j)$ defined for the parent state vector encoded by k. Then at subsequent time moments it undergoes a change of state according to the state transition matrix A_k^L until the *end* state is reached. In the DBN representation the system never enters the *end* state, but the

corresponding variable e_t^L is set to 1 instead, indicating that the higher-level sub-model can now change its state. Thus the conditional probability of a state at level L is written as

$$P(q_t^L = j \mid q_{t-1}^L = i, e_{t-1}^L = f, q_t^{1:L-1} = k) = \begin{cases} \tilde{A}_k^L(i,j), \text{if } f = 0 \\ \pi_k^L(j), \text{if } f = 1 \end{cases} \tag{4-52}$$

where it is assumed that $i, j \neq end$. Matrix $\tilde{A}_k^l$ is the state transition matrix at level l given that the parent variables are in state k and the *end* state is never reached, i.e. it is defined from the following equality:

$$\tilde{A}_k^l(i,j)(1 - A_k^l(i,end)) = A_k^l(i,j). \tag{4-53}$$

The conditional probability for e_t^L is determined as

$$P(e_t^L = 1 \mid q_t^{1:L-1} = k, q_t^L = i) = A_k^L(i,end). \tag{4-54}$$

The observable feature vector D_t is generated according to a probability function conditioned on the whole stack configuration $q_t^{1:L}$.

To write down the conditional probabilities for intermediate level l, we need also to take into consideration the variable e_t^{l+1} indicating whether the sub-model has finished or not. If this variable is 0, which means that the sub-model has not finished, the state transition at level l is forbidden. Hence, the conditional probability of state q_t^l is written as

$$P(q_t^l = j \mid q_{t-1}^l = i, e_{t-1}^{l+1} = b, e_{t-1}^l = f, q_t^{1:l-1} = k) = \begin{cases} \delta_{ij}, \text{if } b = 0 \\ \tilde{A}_k^l(i,j), \text{if } b = 1 \text{ and } f = 0 \\ \pi_k^l(j), \text{if } b = 1 \text{ and } f = 1 \end{cases} \tag{4-55}$$

where δ_{ij} is the Kronecker delta. The variable e_t^l can be set to 1 only when the state q_t^l is allowed to enter a final state. Therefore, its conditional probability is written as

$$P(e_t^l = 1 \mid q_t^l = i, q_t^{q:l-1} = k, e_t^{l+1} = b) = \begin{cases} 0, \text{if } b = 0 \\ A_k^l(i,end), \text{if } b = 1 \end{cases} \tag{4-56}$$

The conditional probabilities for the top level of the HHMM are written similarly to expression (4-55) and (4-56). The only difference that the no parent states are to be specified, that is why the conditioning on $q_t^{1:l-1} = k$ must be omitted.

4.2.3 State Duration Modeling

The proper modeling of semantic segments of video should account for their duration constraints which can be formulated as the corresponding probability distribution. If a segment is modeled with a single state of a HMM, the inherent duration probability density is always meet a

geometric distribution. Indeed, the probability of the Markov chain to remain at a state *i* during first *d* time moments is written as

$$P(q_1 = i,...,q_{d-1} = i, q_d \neq i) = (a_{ii})^{d-1}(1-a_{ii}). \tag{4-57}$$

This geometric distribution is often not appropriate. For example, segments of short duration are unlikely as they have not enough time to convey the semantics to a viewer, while according to this distribution they should be of the highest probability (see the left part of Figure 4-5).

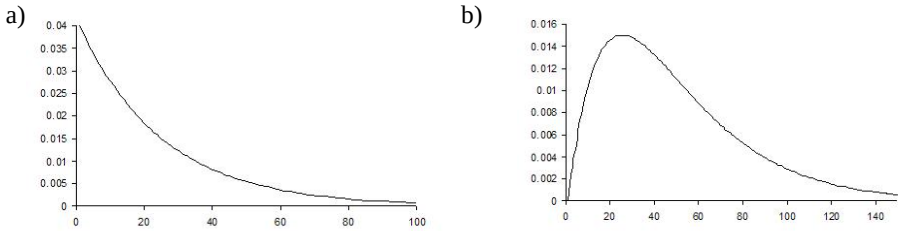


Figure 4-5. A sample plot of the inherent duration probability for the 1-state (a) and 2-state (b) Markov chain (*a*=0.96).

The duration distribution can be fit more freely if the segment is modeled by a chain consisting of several different states. To make this distribution to be decreasing when the duration approaches zero, two state are enough. Consider the two-state chain presented in Figure 4-6. Denoting as $P_1(x)$ and $P_2(x)$ the probability of remaining x times in state 1 and 2 respectively, the probability of remaining in the whole chain is written as

$$P(d) = \sum_{x=1}^{d-1} P_1(x)P_2(d-x) = \sum_{x=1}^{d-1} (a_{11})^{x-1}(1-a_{11})(a_{22})^{d-x-1}(1-a_{22}), \tag{4-58}$$

where the second equality follows from expression (4-57). Assuming for simplicity that $a_{11} = a_{22} \equiv a$, expression (4-58) is continued as

$$P(d) = \frac{1-a^2}{a^2} a^d \sum_{x=1}^{d-1} 1 = (d-1)(1-a)^2 a^{d-2}. \tag{4-59}$$

This is a second-order Erlang distribution, a discrete counterpart of the gamma-distribution, which, for instance, has been shown to be good fit for the probability density function of shot duration in [VAS 97]. A sample plot of this distribution is shown in the right part of Figure 4-5.

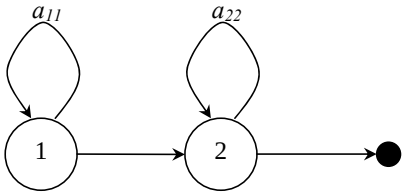


Figure 4-6. A two-state HMM.

Markov chains of sufficient size can model general probability distributions [CRY 88]. Hence, in order to properly realize the state duration, the HMM can be expanded so that its states are expanded to sub-models which have their own topology and transition probability. The resulting structure is called the expanded state HMM (ESHMM) [RUS 87]. The lower-level sub-models are regular HMMs whose states have the same emission probability functions. They usually have a compact left-right topology. The transition coefficients can be learned with the Baum-Welch procedure [RAB 89], an EM-algorithm commonly used for HMM parameters estimation. Alternatively, these coefficients can be calculated directly from the estimated statistical moments [BON 96].

In many applications the state duration distributions in the ESHMM are fitted with quite compact sub-models, thus not increasing crucially the computational complexity with respect to the original HMM. For instance, in [BON 96] three states are assumed to be enough for modeling phone durations in the task of speech recognition. Since the complexity of the probability computations for the regular HMM is quadratic with respect to the total number of the states, the resulting three times growing in the total size of the model engender at most a nine times increase in the computational burden.

The ESHMM is suitable for the tasks where the segment duration distributions are fixed and can be fitted only once during the preliminary learning. Sometimes, however, there is a need to recalculate these distributions at each time step. These recalculations with the ESHMM lead to unacceptable computational complexity. Such a need in the re-estimation of the duration probability arises, for example, when the time units corresponding to the states are not of regular duration, while the distributions of segment duration are defined in the domain of natural time measured in regular units. This is the case in our task of narrative video segmentation into logical story units, or scenes. The elementary time units are camera shots whose duration is not regular and can change from 1-2 seconds to half a minute or even more. The shot length can change considerably from one scene to another, depending on the conveyed semantic, while the time distribution of scene duration remains more or less stable. We estimate the probability of a scene change as a function of shots length and the time duration of the scene. The resulting state

transition probabilities of the corresponding model are dependent from these terms as well and change from one candidate point to another. Such a non-stationary system seems to be modeled more effectively with an extension to the regular HMM where the state duration probability is modeled explicitly. This kind of a model is called a variable duration HMM [RAB 89] or a hidden semi-Markov model (HSMM) [RUS 85].

The functional difference of the HSMM in respect to the regular HMM, is that in the HSMM the transitions from the states back to themselves are prohibited, i.e. the diagonal elements of the state transition matrix $a_{ii} = 0$. Instead of the value of a_{ii}, which implicitly define the state duration in the regular HMM, the occupancy of the state is now determined by an explicit probability distribution. For the practical aspects discussed above, in this thesis we extend the HSMM to be non-stationary in the sense that state duration distributions are defined at each time step. The evolution of the process described by the HSMM is defined as follows. An initial state q_1 is chosen according to the initial state distribution π_i. Once activated, each state *i* remains unchanged during *x* consecutive time moments, where *x* is chosen according to the state duration density $p_i^t(x)$, which is supposed to be non-stationary and dependent on the state activation time *t*. It is assumed that the duration density $p_i^t(x)$ is defined to be non-zero up to a maximum possible duration value τ_i^t. When state *i* is finished, the sequence of observable feature vectors is generated according to the joint observation density $b_i(D_{t:t+x-1})$. The next state j ($j \neq i$) is chosen then according to the state transition probabilities a_{ij}.

To be applied to the HSMM, the forward-backward procedure, used for computationally effective calculation of the posterior state transition probabilities, is modified as follows. We assume that the first state begins at time $t = 1$, and the last state ends at $t = T$, i.e. the model comprises only entire state duration intervals. The forward variable $\alpha_t(i)$ is now defined as

$$\alpha_t(i) = \begin{cases} P(D_{1:t}, q_t = i, q_{t+1} \neq i), \text{ if } t < T \\ P(D_{1:t}, q_t = i), \text{ if } t = T \end{cases}, \; 1 \le t \le T, \tag{4-60}$$

where $D_{i:j}$, $j > i$, denotes the sub-sequence of observable data $D_i, D_{i+1}, ..., D_j$. In the other words, the forward variable defines the probability of observing *t* first data vectors and the state *i* finishing at time *t*. The variable is initialized as

$$\alpha_1(i) = \pi_i p_i^1(1) \cdot b_i(D_1), \; 1 \le i \le N. \tag{4-61}$$

For the subsequent time moments $t = 2, ..., T$ we have the following induction:

$$\alpha_t(j) = \pi_j p_j^1(t) \cdot b_j(D_{1:t}) + \sum_{i=1}^{N} \sum_{\substack{1 \le k \le t-1 \\ k \ge t-\tau_j^k}} \alpha_k(i) a_{ij} p_j^k(t-k) b_j(D_{k+1:t}),\ 1 \le j \le N. \tag{4-62}$$

The first term of this expression disappears when time t exceeds the maximum possible state duration τ_j^1. The value τ_j^k limits the range for the second sum of the second term for time t so that the state duration does not exceed its maximum allowed value (in algorithmic realization this limit can be effectively tracked with a queue of values τ_j^k, whose elements are discarded when $t > \tau_j^k + k$). The probability of observing the whole sequence of feature vectors is written in terms of the α's as

$$P(D_{1:T}) = \sum_{i=1}^{N} \alpha_T(i). \tag{4-63}$$

We also define two backward variables as

$$\beta_t(i) = P(D_{t+1:T} \mid q_t = i, q_{t+1} \ne i),\ 1 \le i \le N, \tag{4-64}$$

$$\beta_t^*(i) = P(D_{t+1:T} \mid q_t \ne i, q_{t+1} = i),\ 1 \le i \le N, \tag{4-65}$$

i.e. $\beta_t(i)$ and $\beta_t^*(i)$ are the probabilities of partial feature vector sequence $D_{t+1:T}$ given that state i ends at time t and given that state i begins at time $t+1$ respectively. We initialize the recursion as

$$\beta_T(i) = 1,\ 1 \le i \le N. \tag{4-66}$$

Then for $t = T-1, T-2, ..., 1$ by induction we have

$$\beta_t^*(i) = \sum_{x=1}^{\min\{\tau_i^t, T-t\}} \beta_{t+x}(i) p_i^t(x) b_i(D_{t+1:t+x}),\ 1 \le i \le N, \tag{4-67}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_t^*(j),\ 1 \le i \le N. \tag{4-68}$$

The posterior probability of state transitions are computed based on the forward-backward variables as

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j \mid D_{1:T}) = \frac{\alpha_t(i) a_{ij} \beta_t^*(j)}{P(D_{1:T})}. \tag{4-69}$$

To find the most probable sequence of HSMM states, the Viterbi algorithm must be modified so that to account all possible durations of states. Defining the variable $\delta_t(i)$ to be the probability of the best state sequence such that the last state i ends at time t, by induction we have

$$\delta_t(j) = \max_{1\le i\le N} \max_{\substack{1\le k\le t-1 \\ k\ge t-\tau_j^k}} \{\delta_k(i) a_{ij} p_j^k(t-k) b_j(D_{k+1:t})\} + \pi_j p_j^1(t) b_j(D_{1:t}),\ 1 \le j \le N. \tag{4-70}$$

The observable feature vectors are usually assumed to be conditionally independent on each other. Therefore the joint probability of these vectors measured at an arbitrary time run from j to k at a model state i is calculated as

$$b_i(D_{j:k}) = \prod_{l=j}^{k} b_i(D_l). \tag{4-71}$$

Taking into account this equality, the comparisons of the expressions for the forward-backward variables for the basic HMM (4-44)-(4-48) and the HSMM (4-61)-(4-68) allows us to conclude, that the HSMM requires about $\tau^2/2$ times the computation, where τ denotes the average value of τ_j^k. The same is true for the Viterbi procedure as well. This increase in computational burden is, however, not crucial in our task of narrative video segmentation, since the model is applied only once for an input video and the main computational efforts are still required for the feature vector extraction. A pruning theorem is proposed in [BON 93], which reduces significantly the search space in the Viterbi induction (4-70). The resulting increase of computational effort is reported to be about 3.2 times with respect to a conventional HMM, which is usually considerable lower than the use of the original technique. The pruning theorem requires, however, that the state duration distributions be log-convex, which is difficult to provide for our non-stationary model.

4.2.4 Autoregressive Model

The conventional HMM assumes that the observable feature vectors are statistically dependent only on the current states. However it is often the case that there is a strong inherent correlation between consecutive feature vectors, which breaks this assumption. To deal properly with unwanted dependencies, we could consider the joint probabilities of several consecutive feature vectors. But this would require expanding the dimension of the probability functions, which would make more difficult their learning. Alternatively, we could fit the time series of feature vectors with some model, which would allow us to get rid of the information redundancy and pass to a sequence of independent data. An extension to the conventional HMM, where the initial sequence of feature vectors is considered as an autoregressive process, is called an autoregressive HMM (ARHMM). This model was initially proposed for speech signals [JUA 85].

A time series $d_1, d_2, ..., d_T$ is said to represent an autoregressive process, if it can be written as

$$d_t = \mu - \sum_{k=1}^{p} a_k d_{t-k} + e_t, \tag{4-72}$$

where a_k are p constant coefficients, μ is the process mean, e_t is assumed to be a white noise process with mean zero and variance δ^2. The functional difference of the ARHMM is that it does not assume any longer the conditional independence of the current observable feature vector from the past observations, i.e. in the general case

$$P(D_t \mid q_t, q_{t-1}, ..., q_1; D_{t-1}, D_{t-2}, ..., D_1) \neq P(D_t \mid q_t). \tag{4-73}$$

We assume that observable vector D_t consists of K statistically independent components, i.e. $D_t = \{d_t^1, d_t^2, ..., d_t^K\}$. Thus, an autoregressive model can be applied independently for each component and, hence, its upper index is hereafter omitted. As it follows from expression (4-72), an observable feature can be written as

$$d_t = \hat{d}_t + e_t, \tag{4-74}$$

where $\hat{d}_t$ denotes the predicted value calculated as

$$\hat{d}_t = \mu - \sum_{k=1}^{p} a_k d_{t-k}. \tag{4-75}$$

In the other words, values a_k can be considered linear prediction coefficients. Then the independent statistical variable e_t is written as the difference between the real and predicted values of the feature:

$$e_t = d_t - \hat{d}_t = d_t - \mu + \sum_{k=1}^{p} a_k d_{t-k}. \tag{4-76}$$

In the ARHMM e_t is assumed to be conditionally dependent only on the current state and, in fact, replaces the feature value of the conventional HMM. We additionally assume that this value has a Gaussian distribution. The probability of observing e_t at model state i is substituted in the ARHMM by the following value

$$b_i(e_t) \equiv P(e_t \mid q_t = i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2\sigma_i^2} \left(d_t - \mu_i + \sum_{l=1}^{p} d_{t-l} a_l^i \right) \right), \tag{4-77}$$

where σ_i and μ_i are the deviation and the mean corresponding to state i, a_l^i - l-th autoregressive coefficient corresponding to state i. It is assumed that in the general case the observation are generated by different mechanisms at different states. That is why the autoregressive parameters in expression (4-77) are defined depending on the current state.

To estimate the autoregressive parameters of the ARHMM, we use the maximum likelihood learning criterion. As our final task is the video segmentation, it is assumed that the

model is trained on a pre-segmented set of videos. We further assume that each semantic segment corresponds to a single state, i.e. the learning videos are, in fact, marked up into states. Given a training video of length *T*, the maximum likelihood estimates are selected so as to maximize the log-likelihood of the observable sequence written as

$$P_{ML} \equiv \log P(d_1, d_2, ..., d_T \mid q_1, q_2, ..., q_T) = \sum_{t=1}^{T} \log b_i(e_t). \tag{4-78}$$

Substituting expression (4-77) for $b_i(e_t)$, we write the log-likelihood as

$$P_{ML} = \alpha - \beta \sum_{t=1}^{T} \left(\log \sigma_{q_t}^2 + \frac{1}{\sigma_{q_t}^2} \left(d_t - \mu_{q_t} + \sum_{k=1}^{p} a_k^{q_t} d_{t-k} \right)^2 \right), \tag{4-79}$$

where α and β are inessential constants. The optimal autoregressive parameters can be found independently for each state by equaling the partial derivatives to zero:

$$\partial P_{ML} / \partial a_l^i \propto \sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \left(d_t - \mu_i + \sum_{k=1}^{p} a_k^i d_{t-k} \right) d_{t-l} = 0, \tag{4-80}$$

$$\partial P_{ML} / \partial \mu_i \propto \sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \left(d_t - \mu_i + \sum_{k=1}^{p} a_k^i d_{t-k} \right) = 0. \tag{4-81}$$

The resulting system of linear equations can be rewritten as

$$\sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \begin{bmatrix} d_t d_{t-1} \\ d_t d_{t-p} \\ .... \\ d_t d_{t-p} \\ d_t \end{bmatrix} = \sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \begin{bmatrix} d_{t-1} d_{t-1} & d_{t-2} d_{t-1} & d_{t-p} d_{t-1} & d_{t-1} \\ d_{t-1} d_{t-2} & d_{t-2} d_{t-2} & d_{t-p} d_{t-2} & d_{t-2} \\ & & & \\ d_{t-1} d_{t-p} & d_{t-2} d_{t-p} & d_{t-p} d_{t-p} & d_{t-p} \\ d_{t-1} & d_{t-2} & d_{t-p} & 1 \end{bmatrix} \begin{bmatrix} a_1^i \\ a_2^i \\ \\ a_p^i \\ \mu_j \end{bmatrix}. \tag{4-82}$$

After solving this system with respect to a_k^i and μ_i, these parameters can be used to estimate variation σ_i^2 by equaling the corresponding partial derivative to zero, which yields

$$\sigma_i^2 = \frac{\sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \left(d_t - \mu_i + \sum_{k=1}^{p} a_k^i d_{t-k} \right)^2}{\sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} 1}. \tag{4-83}$$

Expression (4-82) and (4-83) can be easily generalized for the case where several learning videos are provided by extending the sums on *t* to all the available data.

4.3 Conclusions

A statistical framework has been proposed for the task of video segmentation which focuses on the detection of segment boundaries. The common approach to the task is to select the single best model of the whole video. This does not necessarily lead to the optimal segmentation performance which is commonly measured in terms of recall and precision. In our approach we select segment boundaries so as to maximize the performance metrics directly. The approach is based on the posterior probabilities of the boundaries estimated at each candidate point. It is finally formulated as a task of constrained optimization, for which a computationally feasible algorithm, applicable to the general case of multiple semantic segments, is proposed.

The posterior probabilities of segment boundaries can be estimated in different ways, depending on the particular model of the video. In this chapter we describe a hidden Markov model and its modifications which have been shown to be effective tools for modeling the dynamics of time sequences, such as video. A basic model is first defined, and its application to the video segmentation task is considered. Several modifications of this model are presented then, which allow us to overcome some inherent limitations: a hierarchical extension used to model multi-level semantic structure; a hidden semi-Markov model which enable the use of arbitrary distributions of state duration; an autoregressive version which deals properly with statistical interdependencies existing between consecutive feature vectors.

5 Narrative Video Segmentation

In this chapter we adopt and experimentally evaluate our stochastic segmentation approach to the task of narrative films segmentation into scenes, sometimes called also logical story units or sequences (hereafter in this chapter we reference them as scenes, which is the more specific term, used for narrative video, than more general term “logical story units”). These units are meaningful semantic elements of the whole story told by a narrative video. They can be identified based on features extracted from both the image sequence and the audio track of the video. We propose feature extraction techniques which have been shown to provide a high segmentation performance. To apply our statistical approach, we treat these features as statistical variables and describe them with two models, depending on the assumptions on their conditional dependencies. As the result, two statistical segmentation methods are derived, called a maximum likelihood and a hidden Markov model-based one. We also develop a statistical segmentation technique which selects scene boundaries sequentially and is called a sequential segmentation algorithm. It performs scene segmentation in one pass and has surprisingly high performance.

This chapter is organized as follows. First we specify the segmentation task, giving a strict definition of a scene and describing our database of ground-truth video used for performance comparisons. Then we propose audio-visual features which provide evidence about scene boundaries. After this we propose and evaluate a deterministic rule-based segmentation technique so as to provide benchmarking data for subsequent comparisons with the statistical methods which are described and evaluated in the last three subsections.

5.1 Segmentation Task

5.1.1 Scene Definition

Scenes are distinguished by viewers intuitively as temporal units showing an action or an interaction between characters, such as a dialog and a chasing episode. However, to give an objective and quite general definition of a scene is not a simple task. In cinematography a scene is defined as “a series of shots that communicate a unified action with a common locale and time” [BOG 00]. We accept this definition and give some more precise specifications based on particular video editing techniques so as to avoid the subjectivity as much as possible.

The unity of action, place and time, being expressed as the same objects, background or settings and lighting conditions, causes the visual resemblance of shots composing a scene. In addition, in order to facilitate the perception of scenes as unified segments, video producers hold

some common principles or editing rules, aiming to provide temporal and spatial continuity. One of these principles concerns the positioning of the cameras and reads that they should be placed at one side of an imaginary line. As a result, the action is shown at the same background, and the perceived relative location of the characters remains unchanged. According to another principle, a scene is usually taken by several cameras simultaneously, yielding parallel long shots which are then cut and juxtaposed into one image sequence during the montage. The resulting scene can be distinguished as a sequence of interleaving visually similar shots which correspond to the same locale taken from different points or to different locales. For example, a conversation between two persons is typically shown by switching the camera periodically from one talking person to the other. This conversation can occur at the same locale, or it can be a dialogue on the telephone where the personages are at different places. A scene transition sometimes can be visually distinguished by a change in the lighting conditions, corresponding to a change in time, or by a change of the color tone, used to underscore the specific mood of the scene.

To introduce the overall space and the main characters, an establishing shot is often inserted at the beginning of a scene. This shot often shows an outside view of the building where the action takes place and can be interpreted broadly whether it has the common locale with the following shots, so as to be related to the same scene, or not. We consider the establishing shot as a part of a scene for which they establish the settings and the main characters. Sometimes several auxiliary shots can precede the main action, e.g. they can show characters coming up and then entering the building. Also a re-establishing shot can be added at the end of the action, describing the overall space again. We always merge these shots with the main action which they precede or finish up.

Commentaire [LC2]: needs some images and examples to illustrate all these principles

Several semantically related events, which take place at the same time but at different locales, can be shown simultaneously using a parallel cutting technique. This results in a sequence of interleaving segments which are changed quite fast to be perceived as one scene. However, it is often difficult to say objectively, whether these segments are independent scenes or not. In spite of the fact that such parallel events do not occur at the same place, we merge them into one scene if the segments corresponding to the same locale are quite short, i.e. their duration is under a threshold value, set to 25 seconds in our case. We also merge a sequence of short retrospective episodes and the shots showing the current dramatic incident into one scene using the same threshold of 25 seconds. This threshold is chosen somewhat arbitrarily to provide the maximum objectivity of scene definition. Note, however, that it is not applied frequently.

5.1.2 Ground Truth Video and Performance Evaluation Criteria

For the lack of common benchmark data, a database of four ground-truth movies of different genres – drama “A beautiful mind”, mystery “Murder in the mirror”, French comedy “Si j’etais lui” and romance “When Harry met Sally” – was prepared and manually segmented into semantic scenes, providing reference data. The comparative performance of different segmentation techniques is measured hereafter in this chapter in terms of recall and precision and the integral measure F1 defined by expression (4-2), (4-3) and (4-10) respectively. Detected scene boundary is considered as correct if it coincides with a manual scene boundary within ambiguity of 5 sec (the same ambiguity of 5 sec was admitted in TRECVID evaluations for the task of news video segmentation into stories [GUI 04]). Otherwise it is considered as a false alarm. A manual scene boundary is considered as missed if it does not coincide with any of automatically detected boundaries within the same ambiguity of 5 sec. The beginning of the first scene and the end of the last one are assumed to be given a priory and are excluded from consideration. Clamed scene boundaries are related to reference ones within a time interval which begin at the middle of the first scene and ends at the middle of the last one. The performance comparisons are made inside time intervals which have the total duration of about 22000 seconds and include 234 manually labeled scene boundaries.

As a scene is defined as a continuous sequence of camera shots, the candidate points of scene boundaries are chosen at the shot transitions. We do not assign any specific semantic label to scenes and consider them as being of the same type. An input raw video is supposed to be pre-segmented into shots using an automatic twist-threshold method [DON 01] based on color histogram measure of inter-frame similarity.

5.2 Feature Extraction

In this section we consider the basic ideas underlying the segmentation separately in the visual and audio domains and propose visual and audio features - video coherence and audio dissimilarity - providing evidence of the presence or absence of a video scene boundary. These features form the input data sequence for the segmentation techniques considered later in this chapter.

5.2.1 Video Coherence

To derive our video coherence measure we start from description of two conventional methods of video segmentation scenes – the scene transition graph-based and short memory-based ones. This provides us with motivations for our proper visual feature and with the reference enabling performance comparisons which are carried out using the ground-truth data.

5.2.1.1 Graph-Based Method

According to the editing rules, scenes of narrative video are shot by a small number of cameras. The position of each camera usually does not change much during a scene. Therefore the background and often the foreground objects shot by one camera are mostly static and, hence, the corresponding shots are visually similar to each other. In the clustering-based approach [MAH 00, YEU 96] these shots are clustered into equivalence classes and are labeled accordingly. As a result, the shot sequence is transformed into a chain of labels identifying the cameras. Within a scene this sequence usually consists of the repetitive labels. Consider, for example, a typical scene which shows a dialog of two persons. As a rule, such a scene is mostly produced by two cameras; each of them shots a view of the corresponding person. Let's denote these cameras as *A* and *B*. Then the sequence might be looked as *ABABAB*.

When a transition to another scene occurs, the camera set changes. For example, a transition from a scene shot by cameras *A* and *B* to a scene produced by cameras *C* and *D* could be represented by a chain *ABABCDCD*. If within a scene a shot change can be followed then by return to a shot of the same cluster, after a scene transition such return is impossible. Hence, the only possible transitions between the shots that precede a scene boundary and the shots that follow it are "before" and "meets" according to Allen's definitions [ALL 83], whereas the possible relationships between shots within a scene are "overlaps" or "during". So, scene transition can be detected through classification of the temporal relations between the shot clusters. In [MAH 00] these relations are generated through a temporal-clusters graph built for the cluster chain.

In practice two shots belonging to different scenes can be found visually similar because of their accidental resemblance or a reuse of the same locale, e.g. several scenes can take place at the same room. The graph-based method fails to separate scenes in this case. In order to reduce the probability of this undesirable situation, an additional constraint is imposed on clustering: the shots which belong to different sequences (narrative units combining several scenes) [MAH 00] or are temporally far apart [YEU 96] are considered to be non-similar and are never merged at the same cluster.

5.2.1.2 Short Memory Model

The segmentation method based on a short memory model [KEN 98, SUN 00] allows for scene boundary detection through continuous coherence measure. In comparison with the discrete clustering-based approach it is more flexible and takes into consideration shot length and spacing. The approach based on the short memory model views segmenting of video as the

ability to recall the past data stored in a memory buffer, given the present data stored in an attention span. The recall between two shots *a* and *b* is formalized as follows:

$$R(a,b) = Sim(a,b)T_a T_b (1 - \frac{\Delta t}{T_M}), \quad (5\text{-}1)$$

where T_a and T_b are the ratio of the length of shots *a* and *b* to the total memory size T_M, Δt is the time difference between these shots (it is supposed that $\Delta t < T_M$, otherwise the recall is equal to 0), *Sim(a,b)* is their visual similarity.

Commentaire [LC3]: needs to indicate the intuitive indea behind such a formulas : the correspondence between this formulas and basic scene editing rules ; attention span is it included in memory buffer ?

Scene boundaries are detected in the local minima of the visual coherence curve. The coherence is defined as a measure of how two segments stored in the attention span and the memory buffer are similar to each other. It can be written as follows:

$$C_{MM}(i) = \frac{\sum_{a \in \{T_M \setminus T_{as}\}} \sum_{b \in T_{as}} R(a,b)}{C_{MM\max}(i)}, \quad (5\text{-}2)$$

where $C_{MM}(i)$ is the coherence value at the shot boundary *i*, T_{as} means the duration of the attention span; the normalizing denominator $C_{MM\,max}(i)$ is obtained by setting the similarity to its maximum possible value when computing recall *R* using (5-1) (the normalization is needed to take into account the different number of terms when computing C_{MM}).

Computational complexity of expression (5-2) grows quadratically with the number of shots contained in the memory and can become crucial for short shots or when the parts of the shots are used (so called shot-lets proposed in [SUN 00]). In practice, in order to reduce this complexity, we propose to employ the method of storing partial sums. A matrix *Sum* of partial sums is calculated for each shot of the given video according to the following expression:

$$Sum(i,b) = \sum_{k=1}^{b} R(i,k), \; b \in T_M, \quad (5\text{-}3)$$

where the sums for each *b* are computed using simple recursion.

Then the coherence value is calculated as

$$C_{MM}(i) = \sum_{a \in \{T_M \setminus T_{as}\}} [Sum(a, i + N_{as}) - Sum(a,i)], \quad (5\text{-}4)$$

where N_{as} denotes the number of shots in the attention span. The normalizing denominator is calculated in a similar manner. The use of the partial sums reduces the cost of computation of one coherence value so as it becomes linearly proportional to the number of shots contained in the memory.

Commentaire [LC4]: really ??

5.2.1.3 Our Measure of Video Coherence

The shot clustering implies comparison of the inter-shot similarity measure with some threshold value in order to decide which shots are similar enough to be grouped into the same cluster. This threshold should be low enough to allow for variability of visual appearance of shot frames ~~shot~~ from one camera position. On the other hand, it should be as high as to separate the shots taken ~~form~~ from different camera positions. In practice, however, it is usually difficult to choose the threshold satisfying both of these contradicting requirements at the same time. In this thesis we propose continuous generalization of the clustering-based method which does not require the quantization of shot similarity measure and, hence, it is more flexible and less dependent on its parameters. Like the segmentation method which uses the short-memory model, it detects scene boundaries at the local minima of a continuous curve. In our method we, however, do not accumulate visual similarity for the shots which are probably taken from different camera positions, as these shots usually differ much from each other and, thus, add non-regular noise to the coherence measure. Being direct generalization of the clustering-based technique, our method yields local minima that are better pronounced and has better segmentation performance with respect to the memory model-based method.

Let's consider the following shot clustering technique. First, similarity matrix for the given video is built, each element s_{ij} of which is the similarity value between shots *i* and *j*. Then each pair of shots which are similar enough (i.e. their similarity is higher then a threshold) is merged into one cluster until the whole matrix is exhausted. This is almost a conventional clustering procedure except that the radius of the clusters is not limited.

Since scenes are usually composed of repetitive shots, similarity matrix elements of a high value are grouped at the intersections of the corresponding rows and columns that form square regions in the similarity matrix. Let's quantize the similarities into binary values that can be equal to 0 or 1, so as the value 1 corresponds to a pair of similar shots (which are related to the same cluster) and the value 0 means that the corresponding shots are non-similar. Then an example of the resulting matrix could be looked as the following:

$$
\begin{array}{cc}
 & \begin{matrix} & & & & i & & \end{matrix} \\
\begin{matrix} \\ \\ \\ \\ i \\ \\ \\ \end{matrix} &
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}
\end{array}
$$

In this example a scene boundary occurs at the beginning of shot *i* represented by the corresponding row and column. As the shots situated before this boundary never repeat in the future, their temporal relation with the following shots are just “before” or “meets”. So, the clustering-based algorithm can be reformulated as searching of such square regions along the main diagonal of the quantized similarity matrix and detecting scene boundaries at the points of their contact.

As in the graph-based method [YEU 96], we impose a temporal constraint on the shot similarity: the shots which are temporally far apart are considered to be non-similar, i.e. their similarity is set to 0 (which is the minimum possible value). We do not pre-segment an input video into sequences to separate shot clusters belonging to different sequences as proposed in [MAH 00] since such pre-segmenting in the general case is quite a difficult task itself. In [MAH 00] the sequence boundaries are distinguished as gradual inter-shot transitions, which is not the general case where these transitions can be inserted within scenes.

In addition, if we suppose that the similarity measure is symmetric, the similarity matrix is symmetrical too and its elements lying on the main diagonal are equal to one (i.e. the maximal possible value, since the shot is considered similar to itself). Hence, only non-zero values of the similarity matrix which are concentrated near the main diagonal and lie only below or above it (due to the symmetry) have to be calculated and stored.

Let’s define for each shot *i* of given video the following variable:

$$C^0(i) = \max_{a<i,b\geq i} Sim(a,b), \tag{5-5}$$

According to our clustering-based segmentation scheme this variable, being computed for the quantized similarities, takes on a value 0 for shots whose beginning corresponds to a scene boundary and a value 1 otherwise. Hence, the segmentation procedure can be reformulated as follows. At first the value C^0 defined by expression (5-5) is calculated for each shot of the given video based on its shot similarity matrix. Then scene boundaries are detected at the beginning of the shots for which this value is below the similarity quantization threshold.

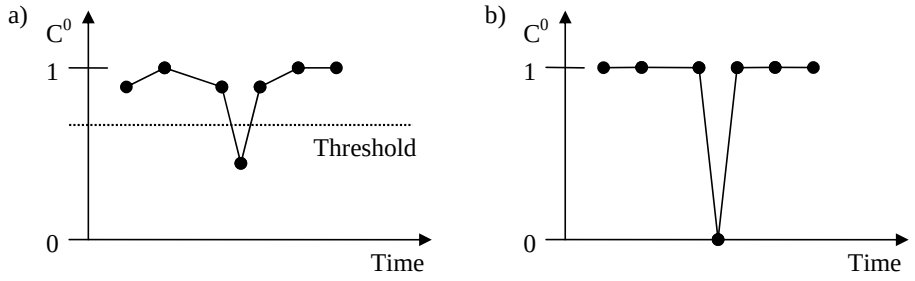


Figure 5-1. C^0 curve sample for real-value shot similarity (a) and for quantized similarity (b).

The shift from the clustering to the curve-based technique makes the segmenting simpler and more illustrative. In addition, it does not require recalculating of the shot clusters in case of a change of the quantization threshold. An example of a curve described by the variable C^0 with time and its quantized analogue are given in Figure 5-1. Since scenes usually consist of several contiguous shots, this curve falls below the threshold mostly in the single points that form sharp local minima. Hence, scene detection can be implemented as searching of such minima. This allows us not to use a quantizing threshold and, as a result, enhance segmentation accuracy. In practical implementation another threshold can be used in order to reject weak minima. This threshold, however, does not govern the segmenting procedure so crucially and can be selected less accurately.

In real movies visual similarity between shots within the same scene often is not high enough, especially near scene boundaries due to the use of establishing shots and in action films, where there are many dynamic episodes. Because of this, minima of the variable C^0 are often badly pronounced and it can happen that a shot from some scene resembles the shot from the previous or the next scene. In this case the segmenting procedure can miss scene boundaries. Consider, for example, two scenes represented by a shot clusters chain *ABABCDADCD*, where a real scene boundary occurs before the first shot of cluster *C* and, because of accidental similarity, one of the shots from the second scene was misclassified as *A*. Since the shot clusters in this example cannot be divided into two non-intersecting groups, clustering-based segmenting procedure fails to detect the scene boundary.

In order to enhance the robustness of the segmenting procedure, we can try to implicitly exclude isolated misclassified shots from consideration. At first glance, the next maximal value after C^0 could be taken in expression (5-5). However, if a single shot is similar to a shot from another scene, it is likely to resemble other shots of the same cluster. In the previous example of a cluster chain the shot from the second scene, misclassified as cluster *A*, is likely to be similar to two shots of this cluster for the first scene. Hence, exclusion of a single pair of maximally similar

shots does not definitely exclude the influence of a single misclassified shot. So, in addition to this pair, we propose to not take into consideration all the maximally similar shots that follow or precede it and define for each shot *i* the following variable:

$$C^1(i) = \min\{\max_{a<i,b\geq i.a\neq a_0(i)} Sim(a,b), \max_{a<i,b\geq i,b\neq b_0(i)} Sim(a,b)\}, \quad (5\text{-}6)$$

where the variables a_0 and b_0 are the shot numbers, for which expression (5-5) attains the maximum:

$$\{a_0(i), b_0(i)\} = \arg\min_{a<i,b\geq i} Sim(a,b), \quad (5\text{-}7)$$

By recursion we can derive variables to exclude the influence of the second misclassified shot, the third one etc:

$$C^n(i) = \min\{\max_{a<i,b\geq i,a\notin\{a_0(i),\ldots,a_{n-1}(i)\}} Sim(a,b), \max_{a<i,b\geq i,b\notin\{b_0(i),\ldots,b_{n-1}(i)\}} Sim(a,b)\}, \quad (5\text{-}8)$$

$$a_n(i) = \arg\max_{a<i,b\geq i,a\notin\{a_0(i),\ldots,a_{n-1}(i)\}} Sim(a,b), \quad (5\text{-}9)$$

$$b_n(i) = \arg\max_{b\geq i,b\notin\{b_0(i),\ldots,b_{n-1}(i)\},a<i} Sim(a,b). \quad (5\text{-}10)$$

The variable C^k has sharp local minima at scene boundaries only if they correspond to *k* misclassified shots. Otherwise these minima are not well pronounced. Generally, as the same pair of maximally similar shots can correspond to several contiguous shots, the defined above variables *C* can remains constant during a period of time. If this period corresponds to a local minimum, the scene boundary position cannot be located precisely. In order to use all the variables *C* together and reduce the probability of wide local minima, an integral variable is defined:

$$C_{\text{int}}(i) = \frac{1}{N}\sum_{k=0}^{N-1} C^k(i), \quad (5\text{-}11)$$

where *N* denotes the number of variables *C* determined by expression (5-5) - (5-10).

By analogy with [KEN 98] we refer variable $C_{\text{int}}(i)$ as *video coherence* and consider its single local value as the visual feature. This value provides flexible evidence of the presence or absence of a scene boundary at the beginning of shot *i* so that the lower is this value, the higher probability of the scene boundary.

5.2.1.4 Inter-Shot Similarity Measure

The similarity *Sim(a,b)* between shots *a* and *b* involved in expression (5-5) - (5-10) can be calculated in different manners, depending on what frames are chosen to be the shot representatives, what features are used to represent the frames etc. The following measure was found experimentally to work quite well in this thesis.

The visual similarity measure between two arbitrary video shots *a* and *b* is defined as the maximum similarity S_{fr} between any key frame f_{ak} in the shot *a* and any key frame f_{bl} of the shot *b*:

$$Sim(a,b) = \max_{k,l} S_{fr}(f_{ak}, f_{bl}), \tag{5-12}$$

Frame-to-frame similarity S_{fr} is calculated as a difference measure between the color histograms representing the frames. The color histograms are defined in HSV-color space and have three dimensions. They have 18 bins for hue, 4 – for saturation and 3 – for value and additionally include 16 shades of grey.

The measure of similarity between two frames f_i and f_j is defined to be:

$$S_{fr}(f_i, f_j) = 1 - \sum_b \left| h_{ib} - h_{jb} \right| / n, \tag{5-13}$$

where h_{ib} is bin *b* in the histogram of frame *i*, and *n* is the total number of pixels in each frame. The second term in this expression is l^1 distance measure normalized to a range [0, 1]. So, the similarity measure takes the values from the same range.

In order to take into consideration dynamic nature of shots, we divide them into quasi-stationary contiguous segments called sub-shots using a sequential one-pass clustering algorithm. For each shot this algorithm is the following:

- Set the beginning of the first sub-shot to the time position of the first frame of the shot.
- For each resting frame *f* taken in the time order from the shot do:
    - Calculate similarity between frame *f* and the first frame of the current sub-shot.
    - If this similarity exceeds a clustering threshold, begin the new sub-shot starting from *f*. Else add this frame to the current sub-shot.
- Remove all short sub-shots. If no segments are left, consider the whole shot as one sub-shot.

The key frames in expression (5-12) are representatives of sub-shots. In this work each sub-shot is represented by a frame whose color histogram is the closest to the mean histogram of all the frames in this sub-shot. A result of shot segmenting into sub-shots is illustrated in Figure 5-2, where color histogram is schematically presented along a single color axis.

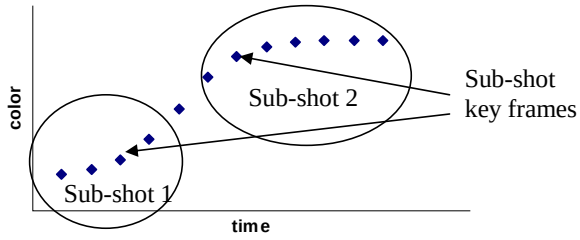


Figure 5-2. A schematic example of color dynamics in a shot divided into two quasi-stationary sub-shots.

5.2.1.5 Experimental Evaluations

The effectiveness of our video coherence was experimentally tested using our ground truth video database. In these experiments we did not apply the statistical approach, as a conventional deterministic one, which seeks for scene boundaries at the local minima of the video coherence curve, works quite well when a single curve is used as the input data (the statistical approach can use implicitly the timing information included as scene duration prior). The simplest scene segmentation algorithm which uses a video coherence curve detects scene boundaries when its point just falls below a threshold. Experimental evaluations show, however, that better segmentation precision can be attained if scene boundaries are detected in local minima. We found that the following algorithm works well enough. First, all the local minima are detected as potential scene boundaries. Two contiguous windows w1 and w1 that adjoin each local minimum to the left and to the right are defined (see Figure 5-3). They typically contain 3 shot boundaries. Two parameters a and b are related to the local minima: they are respectively the difference between the maximum in the windows w1 and w2 and these minima. A scene boundary is detected in a local minimum, if the following conditions hold true:

- The given local minimum is a global one in the windows w1 and w2.
- It is less than a threshold $t1$.
- The value min(a.b) exceeds a threshold $t2$.

Commentaire [LC5]: what is the rational of this ?

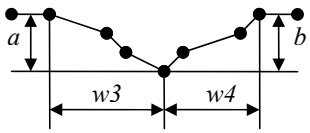


Figure 5-3. Local minimum parameters used in the scene segmentation algorithm.

The performance of the segmentation algorithm described above was experimentally tested for our coherence measure C_{int} (5-11) and that of the short memory model C_{MM} (5-2). The

performance evaluations results total for all the ground-truth video are given in Table 5-1. Threshold values $t1$ and $t2$ of the local minima search algorithm was selected so as to maximize integral performance measure F1 individually for each type of coherence curve. As it can be seen from Table 5-1, the use of the video coherence C_{int} yields the gain in recall and F1 measure. The results obtained for coherence C_{MM} look worse than those reported in [SUN 00] (in this work the authors apply the video coherence measure to so-called shot-lets – 1 sec parts of shots, which, however, does not improve the performance for our database). Besides some possible differences in technical realizations of the segmentation procedure, it might be explained by the fact that we used different ground truth video (for the lack of a common reference database) and that in our case the scenes were defined in a different way, being considered as semantic ones rather than as only groups of visually similar or repetitive shots. Typical behavior of the curves described by the two coherence measures is presented in figure 3. As it could be expected they are somewhat correlated with each other. The measure C_{int} however is more stable, more regular which results in higher segmentation performance.

Commentaire [LC6]: ???

Video coherence	Precision, %	Recall, %	F1, %
C_{int}	54.1	64.3	58.8
C_{MM}	55.8	49.6	52.5

Table 5-1. Segmentation performance comparison for different video coherence measure

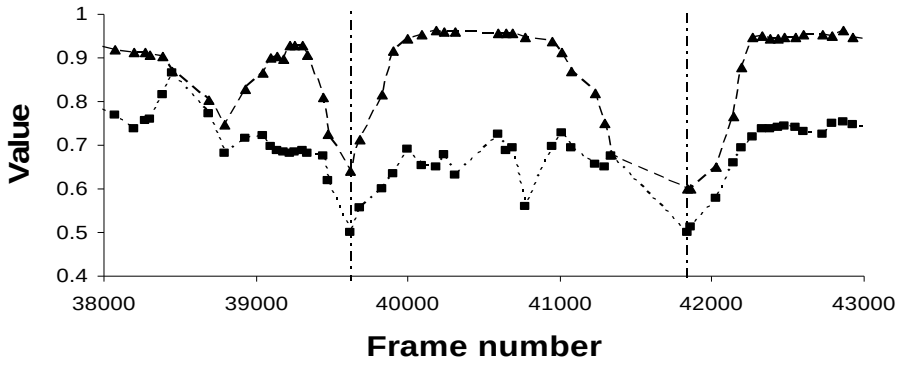


Figure 5-4. Video coherence C_{int} (the upper curve) and C_{MM} (the bottom curve) defined by expression (5-11) and (5-2) respectively for the film “Murder in the mirror”. Two vertical dash-dot lines delimit scenes.

5.2.2 Audio Dissimilarity

A scene transition in video usually entails abrupt change of some audio features caused by a switch to other sound sources and, sometimes, by film editing effects [CHE 02, SUN 00, CAO 03]. Hence, this change can be used as an indicator of the presence of a scene boundary. Since short-term acoustic parameters often are not capable to represent properly the sound environment, we accumulate these parameters within a long-term window. Comparing the resulting descriptors for two adjacent time windows at the point of a potential scene change (shot transition in this thesis) provides us with the evidence whether the scene change really occurs or not. The measure of the difference between these descriptors is referenced hereafter as the audio dissimilarity or an audio feature and is calculated as follows.

To calculate the short-term acoustic feature vector for a sound segment we divide the spectrum obtained from Continuous Wavelet Transform (CWT) into windows by application of triangular weight functions W_i with central frequencies f_i in Mel scale as it is done in the case of Mel Frequency Cepstrum Coefficients calculation (see Figure 5-5). Unlike the FFT, which provides uniform time resolution, the CWT provides high time resolution and low frequency resolution for high frequencies and low time resolution with high frequency resolution for low frequencies. In that respect it is similar to the human ear which exhibits similar time-frequency resolution characteristics [TZA 01].

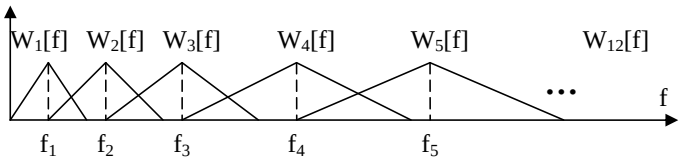


Figure 5-5. Triangular weight functions with central frequencies in Mel scale.

Then energy values E_i in each spectral window are computed and finally, the matrix of spectral band ratios is obtained as

$$K_{ij} = \log(E_i / E_j), \tag{5-14}$$

The elements lying above or below the main diagonal (i.e. the top-right or bottom-left triangular) of the matrix K are taken as our acoustic features. The resulting acoustic feature vector is not affected by main volume change unlike spectral coefficients. At the same time it allows us to detect changes in acoustic environment.

The procedure of audio dissimilarity curve calculation is done by moving of two neighboring windows (with size 8 and step 0.5 seconds in our experiments) along the audio stream and obtaining the distance between the distributions of the corresponding acoustic

features. Various measures may be used as a distance or dissimilarity for the task of acoustic segmentation: Bayesian Information Criterion [CHE 98], Second-Order Statistics [BIM 95], Kullback-Leibler (KL) distance applied directly to distribution of spectral variables [HAR 03a].

The KL-measure is a distance between two random distributions [COV 03]. In the case of Gaussian distribution of random variables the symmetric KL distance is defined as:

$$KL(X_1, X_2) = \left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2}\right) + (\mu_1 - \mu_2)^2 \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right), \tag{5-15}$$

where μ and σ are the mean value and the variance of compared distributions.

Instead of multi-dimensional KL applied to a feature vector of spectral bands ratios a sum of KL distances applied to each element of the vector is used in this work as audio dissimilarity measure:

$$D = \sum_{ij} KL(K1_{ij}, K2_{ij}), \tag{5-16}$$

where K1 and K2 – feature matrices for the neighboring windows. As an observable feature of a scene boundary in the audio domain in this work we extract the maximal value of audio dissimilarity in a time window of about 4 seconds centered in the corresponding candidate point so as to tolerate small misalignments between the audio and image streams of video.

5.3 Rule-Based Segmentation

In real applications neither the video coherence nor the audio dissimilarity can determine unambiguously the presence or absence of a scene boundary. Low values of the video coherence can be encountered within scenes due to, for example, intensive camera movements or the use of establishing shots, while the high values do not necessarily signify the absence of the scene transition (which is more rare though) because of accidental coincidences. As for the audio dissimilarity, abrupt changes in the sound environment can occur naturally within scenes or be the result of editing effects, such as music. So, to detect scene boundaries more reliably, both the audio and video features should be taken into account. The conventional rule-based approach seeks for potential scene boundaries based on one of these features and then uses the other to confirm or reject them in the final decision. The same principles underlie our rule-based segmentation which is described and experimentally evaluated further in this section.

5.3.1 Segmentation Algorithm

We find the potential scene boundaries based on the video coherence curve. This curve, however, provides more information for combined scene segmentation than just potential scene boundaries. High values of such measure signify a high level of repetitiveness of the

corresponding shots that are likely to be produced in the same physical settings. Therefore, these values correspond to low probability of scene transition. That is why in our segmentation algorithm shot boundaries having video coherence higher than a threshold *v1* are always rejected. On the other hand, the value of video coherence in local minima is an indicator of reliability of the corresponding scene transition candidates. The lower are these local minima, the more probably they are accompanied by scene transitions. Therefore, low minima falling below a threshold *v2* are always accepted as reliable scene transitions.

We use audio the audio dissimilarity to confirm or reject potential scene boundaries that correspond to intermediate values of video coherence lying between thresholds *v2* and *v1*. Much as for video coherence, the parameters of audio dissimilarity picks can be used for estimating of scene transition probability. The most significant quantity is the value of these picks – the higher is this value, the higher is the probability. Experimental evaluations of scene segmentation based solely on audio data show that better performance is achieved if scene boundaries are detected using simple comparison of audio dissimilarity with a threshold value at shot boundaries. In the combined segmentation algorithm potential scene transitions are confirmed if the corresponding dissimilarity value exceeds a threshold *a,* else they are rejected.

The resulting segmentation algorithm for a given video is referenced hereafter as a three-threshold algorithm. It is written as follows:

1. Preset the threshold parameters – *v1*, *v2* and *a*.
2. Calculate the value of video coherence at the shot boundaries.
3. For each shot boundary *B* and the corresponding value *C* of video coherence do:
   - Compare *C* with the threshold *v1*. If greater, continue with the step 3.
   - Check whether *C* is a strong local minimum detected using two-window approach described above (see Figure 5-3). If no, continue with the step 3.
   - If *C*<*v2*, add the shot boundary *B* to the set of scene boundaries and continue with the step 3.
   - Calculate audio dissimilarity *A* for the shot boundary *B*. If *A*>*a*, add *B* to the set of scene boundaries.

An example of real video coherence and audio dissimilarity curves, normalized to a comparable value ranges, are shown in Figure 5-6. Both the scene boundaries in this figure are detected correctly. The first one corresponds to a strong minimum of the video coherence value which is below the threshold *v2*. The second boundary is detected because its video coherence value takes a local minimum between the thresholds *v1* and *v2* and is confirmed by a high value of audio dissimilarity.

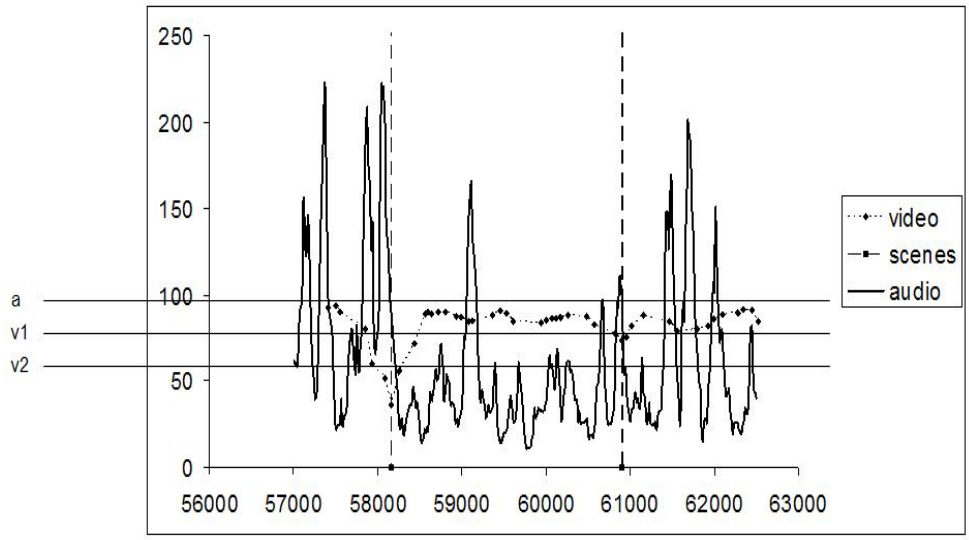


Figure 5-6. An example of video coherence ("video") and audio dissimilarity ("audio") curves. "Scenes" lines mark the scene boundaries.

5.3.2 Performance Evaluation Results

In this subsection we describe the results of experimental evaluations of our rule-based segmentation algorithm. To provide the best performance, the video coherence was computed according expression (5-11) included 3 terms, i.e. in N was equal to 3; the video similarity was calculated within two contiguous groups of 5 shots adjoining the point under consideration.

The results obtained for each of the four films of the database are presented in Table 5-2. The threshold values in this trial were chosen so as to maximize (through the full search) the integral measure F1 total for all the films. As it can be seen from the table, the precision and recall has sometimes quite different values, resulting in the decrease of the integral measure F1. This is caused by the various behavior of the audio-visual features depending on the specific film, which does not allow us to choose the thresholds providing the balanced values of recall and precision for all the films at the same time. Quite a low recall for film "Murder in the window", for example, is caused by too low threshold values for the video coherence which are more suitable for films where the scene changes are often accompanied with changes in the color tone and, hence, with low video coherence.

Film	Precision, %	Recall, %	F1, %
A beautiful mind	64.4	67.1	65.7
Murder in the mirror	81.8	42.9	56.3
Si j'etais lui	56.8	68.9	62.2
When Harry met Sally	70.2	57.9	63.5
Total	65.6	59.4	62.3

Table 5-2. Performance of the three-threshold segmentation algorithm. The thresholds are chosen so as to maximize F1 measure for all 4 films: $v1$=0.78, $v2$=0.64, a=130.

The capability of our segmentation approach to fuse audio-visual features can be revealed from Table 5-3. The results are total for all 4 films. The first row presents the segmentation performance when only the visual feature was used and scene boundaries are claimed at the local minima of the coherence curve. The performance of the segmentation algorithm which is based solely on the audio dissimilarity is given at the second row. In this algorithm scene boundaries are claimed when the audio dissimilarity exceeds a threshold value. The performance of the three threshold algorithm fusing both the video and audio features is presented at the third row. The threshold values are chosen separately for each algorithm so as to maximize the measure F1. As it follows from the table, fusing the visual and audio features enhances the integral performance.

Feature used	Precision, %	Recall, %	F1, %
Visual	54.1	64.3	58.8
Audio	29.6	64.1	40.5
Visual + audio	65.6	59.4	62.3

Table 5-3. Audio-visual data fusion capability of the three-threshold segmentation algorithm.

To estimate how general are the threshold values obtained for a separate set of the training data, cross-validation tests have been carried out. The learning set, used to choose the optimal thresholds, included three films and the test set consisted of the resting forth. The performance evaluation results are given in Table 5-4 along with the corresponding threshold values. The comparison with Table 5-2 allows us to notice a considerable deterioration in the performance when the thresholds are chosen for a separate data set. One of the reasons of this is an even more disproportion between recall and precision caused by inappropriate threshold values which are strongly dependent on the particular learning data.

Film	Precision, %	Recall, %	F1, %	$v1$	$v2$	a
A beautiful mind	65.6	60.9	63.2	0.74	0.61	70
Murder in the mirror	78.6	18.0	29.3	0.69	0.64	130
Si j'etais lui	56.8	41.7	48.1	0.75	0.61	60
When Harry met Sally	66.7	29.6	41.0	0.75	0.55	70
Total	64.4	38.5	48.2	-	-	-

Table 5-4. Performance of the thee-threshold algorithm in cross-validation tests.

5.4 Maximum Likelihood Ratio Segmentation

A deterministic segmentation algorithm performs quite well if it is adapted in a heuristic manner to a specific feature, such as video coherence. This algorithm, however, is hardly extensible to multiple data sources, which is required for further performance enhancements. As we could see it above, the fusion of multiple data in a deterministic manner leads to the use of numerous thresholds which are difficult to be selected properly. Moreover, the data, being compared with the thresholds, are coarsened excessively. The different nature of the features makes difficult their fusion into a single measure. In our stochastic approach we assume the features to be random variables and treat them in the same terms of conditional probabilities, which allows us to fuse the features in a flexible unified manner. The posteriori probabilities of scene boundaries in this approach can be estimated in different ways, depending on the assumption on the conditional dependencies and the prior distributions. In this section we derive quite a segmentation procedure, called a maximum likelihood ratio algorithm, which, however, performs well in the task of scene segmentation provided that the features are chosen properly.

5.4.1 Segmentation Algorithm

At each candidate point of scene boundary (in this work it is a shot change moment) let's consider a posterior probability $p(s|D)$, where $s \in \{0,1\}$ is a random variable corresponding to the presence (s=1) or absence (s=0) of a scene boundary, D – a locally observable feature providing information about s. According to the Bayesian rule

$$p(s=1 \mid D) = \frac{p(D \mid s=1)p(s=1)}{p(D \mid s=1)p(s=1) + p(D \mid s=0)p(s=0)} = 1/\left[1 + \frac{1}{L}\frac{p(s=0)}{p(s=1)}\right], \qquad (5\text{-}17)$$

where $L \equiv \frac{p(D \mid s=1)}{p(D \mid s=0)}$ is likelihood ratio, p(s) – the prior probability of s. Let's assume that feature vectors are conditionally dependent only from the value s at the current time moment and that the prior probabilities of scene boundaries are fixed for a given video. Then the posterior probabilities defined by expression (5-17) can be estimated independently at each candidate point. Suppose that our segmentation algorithm claims N distinct scene boundaries at the points of their maximum posterior probability $p(s=1 \mid D)$ which provide the optimal performance according to our optimality criterion. As the posterior probability is an increasing function of L, N points with maximal value of likelihood ratio L can be selected instead.

Remind that in our stochastic approach N is chosen to be equal to the expected number of actual scene boundaries, so as to provide approximately equal recall and precision. In real applications the given above assumptions seem to be too strong to estimate this number correctly. Therefore we propose the following estimate:

$$N = \frac{T}{S}, \tag{5-18}$$

where T denotes the duration of a video to be segmented, S is the mean scene duration evaluated from a learning data set. In this expression it is assumed that a mean duration of scenes does not depend much on the specific film (the per-film mean scene duration for our ground-truth video changes from 78 to 117 sec while the total mean scene duration is 96 sec).

Experimental evaluations of the proposed segmentation algorithm have shown that its performance is greatly improved if it is constrained to select scene boundaries which are temporally apart from each other at least by some threshold value S_{min}. This can be explained by the fact that each observable local feature vector D used in this work is in fact conditionally dependent from its context and a high value of likelihood ratio in a point corresponding to an actual scene boundary is often accompanied by high likelihood ratio values at surrounding candidate points which should be excluded from consideration (see an example of likelihood ratio curve for video coherence and audio dissimilarity features presented in Figure 5-7).

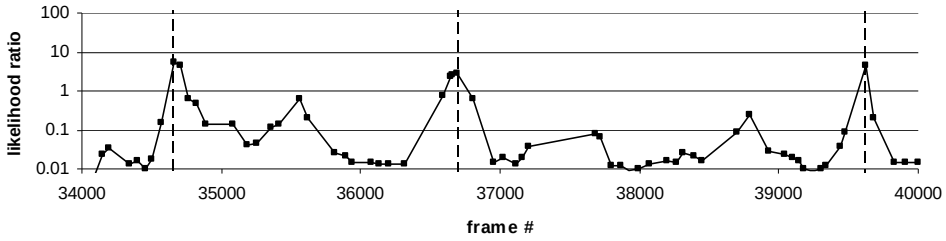


Figure 5-7. Log-scale likelihood ratio versus frame number. Vertical dashed lines delimit scenes.

So, the scene segmentation algorithm is finally formulated as follows.

1. Segment an input video into shots and select shot transition moments as candidate points of scene boundaries.
2. At each candidate point calculate the likelihood ratio for the corresponding observable feature vector.
3. Pronounce N scene boundaries at the points with maximal likelihood ratio separated from each other at least by the temporal interval S_{min}, where N is calculated according to expression (5-18).

In multimodal segmentation observable feature vector D integrates M sources of information each described by its own vector d_i, i.e. $D = \{d_1,...,d_M\}$. We suppose that these sources are conditionally independent given the value s. So, we can write

$$p(D \mid s) \equiv p(d_1,...,d_M \mid s) = \prod_{i=1}^{M} p(d_i \mid s), \tag{5-19}$$

and, hence, likelihood ratio of the whole data D is calculated as the product of likelihood ratio values l_i evaluated for each i-th informational source independently:

$$L \equiv \frac{p(d_1,...,d_M \mid s=1)}{p(d_1,...,d_M \mid s=0)} = \prod_{i=1}^{M} \frac{p(d_i \mid s=1)}{p(d_i \mid s=0)} \equiv \prod_{i=1}^{M} l_i \ . \tag{5-20}$$

Note that this expression provides extensibility of the segmentation algorithm since it allows us to easily add new features as they are available.

In this work we integrate two types of evidence about the presence or absence of scene boundaries – video coherence and audio dissimilarity measure. To provide low dependence of the video coherence feature from the surrounding values, only one term is included in its definition given by expression (5-11). In the other words this feature is defined as variable C^0 by expression (5-5). In the ideal case this variable has a low value in a single point of a scene boundary and is not dependent on the values at the surrounding points. To calculate the likelihood ratio of the features, we use non-parametric estimates of the corresponding

probabilistic distributions [DUD 73]. The resulting dependences between feature and likelihood ratio values for video coherence and audio dissimilarities, obtained for the learning set of all 4 films, are shown in Figure 5-8 and Figure 5-9. The estimates of likelihoods were obtained using the Gaussian kernel with a fixed standard deviation parameter (0.04 for video coherence and 40 for audio dissimilarity).

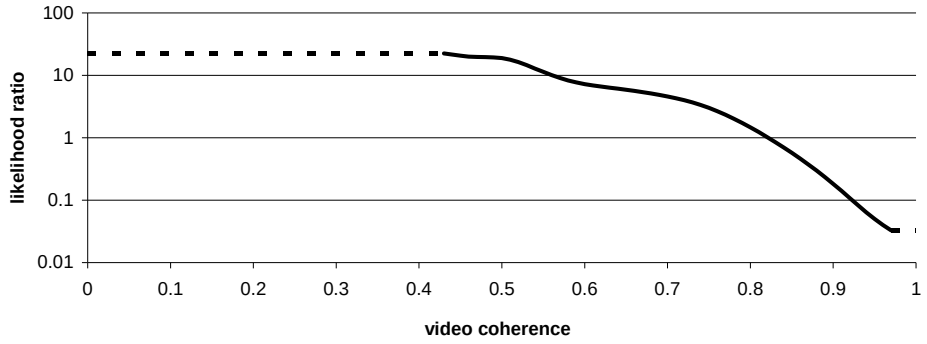


Figure 5-8. Log-scale likelihood ratio versus video coherence. The horizontal dotted line depicts extrapolated values which fall beyond the domain of stable estimate.

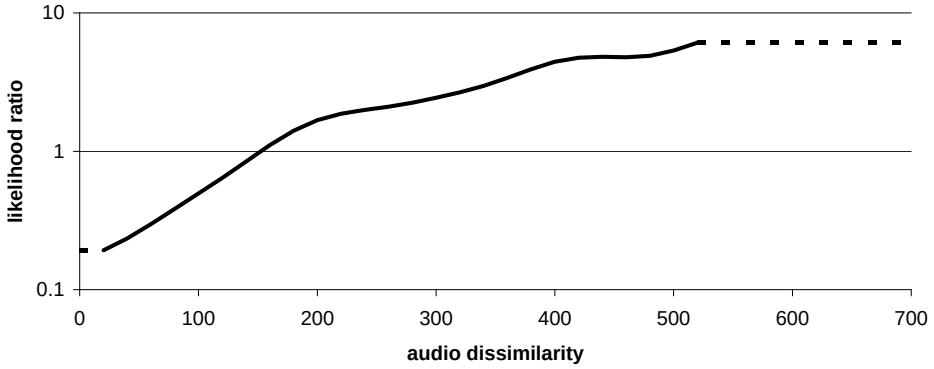


Figure 5-9. Log-scale likelihood ratio versus audio dissimilarity. The horizontal dotted line depicts extrapolated values which fall beyond the domain of stable estimate.

In fact the proposed segmentation algorithm detects scene boundaries at local maxima of likelihood ratio curve and thus reminds conventional unimodal techniques searching for extremums of some scene consistency measure [KEN 98, CHE 02]. From this point of view expression (5-20) can be considered as a way of combining several measures calculated independently for each mode into a single curve.

5.4.2 Experimental Evaluations

In this subsection we report the results of experiments designed to test the proposed maximum likelihood ratio algorithm. Likelihood ratio values for both audio and video features were calculated based on nonparametric estimates of the corresponding conditional probabilities using Gaussian kernel function. Table functions with linear interpolation were used to speed up the calculations. To account for small misalignments between the manually labeled scenes and the actual ones in a learning set, the feature statistics conditioned on the presence of a scene boundary were collected in the time window of 2 sec centered at the position of the corresponding scene boundary label. In the domains (fixed through all experiments described below) where probability estimates became unstable due to the lack of data the likelihood ratio values were extrapolated as constant functions. Only a small portion of data fell in these domains and experimental evaluations demonstrated that their choice was not crucial for segmentation performance.

The evaluated performance of the proposed segmentation algorithm total for all ground truth video available is reported in Table 5-5. The tests were conducted both inside the learning set and using cross-validation technique. The cross-validation allows us to evaluate the generalization capability of the parameters estimation approach as it concerns both the likelihood ratio and mean scene duration. It was performed using the learning set included three films and the test set consisted of the resting forth one until all data were tested. Table 5-5 reports the results which were obtained using the fusion of visual and audio features and those obtained for one of these features only. In the last case the performance would not change if the feature curve itself were used without its transform to likelihood ratio, which is expected since such transform is monotonous.

As it can be seen from Table 5-5, the feature fusion yields significant improvements both in recall and precision. Minor degradation caused by applying cross-validation suggests that these improvements steams from proper modeling rather than parameter overfitting. Note that the video coherence was calculated in a different way with respect to that of the previously described experiments (Table 5-1, Table 5-2, Table 5-3) so as to better fulfill the underlying assumptions on the conditional dependencies. This video coherence has a worse performance when being used alone but is more effective when being combined with the audio dissimilarity.

Features used	Using cross-validation	Precision, %	Recall, %	F1, %
Visual + audio	No	63.2	63.2	63.2
	For LR only	61.1	61.1	61.1
	Yes	60.5	61.5	61.0
Visual only	No	51.7	51.7	51.7
	Yes	50.0	50.9	50.4
Audio only	No	41.5	41.5	41.5
	Yes	39.9	40.6	40.3

Table 5-5. Performance of the maximum likelihood ratio segmentation algorithm, total for all ground-truth video. Abbreviation LR means "likelihood ratio".

5.5 Hidden Markov Model

In this section we adopt and experimentally evaluate a hidden Markov model (HMM) which is used to get the estimates of the posteriori probability of scene boundaries, required in our stochastic segmentation approach. In comparison with the maximum likelihood ratio algorithm, described above, the resulting segmentation technique is more complicated but is based on less restrictive assumptions. In particular, this technique takes into account the non-uniform nature of scene duration priors and allows for dependencies of the observable data on the context, adopting an autoregressive HMM and explicit state duration modeling.

5.5.1 Conditional Dependence Assumptions about the Features

Let's consider an observable audio-visual feature vector D_i measured at a scene boundary candidate point i independently from the rest of vectors. In the general case this vector is conditioned on the fact of presence or absence of a scene boundary not only at this point but at the neighboring points as well. Indeed, in the visual domain the corresponding feature represents visual similarity between two groups of shots adjoining to the point under examination. If a scene boundary appears exactly between these groups, the similarity measure usually has a local extremum. But if a scene boundary lies inside one of these groups, the similarity measure takes an intermediate value which is the closer to the extremum, the closer is the scene boundary (see, for example, Figure 5-6). The similar considerations hold true for the audio data too.

For the purpose of simplification we assume that local features are conditionally dependent on the distance to the closest scene boundary and are independent from the position of the rest of scene boundaries. As the visual feature used in this work is a measure of similarity

between shots which changes only at the points of shot transitions, it is reasonable to assume the conditional dependence of this feature on the distance expressed in the number of shots. Let's denote a time-ordered sequence of scene boundaries as $B = \{b_1, b_2, ..., b_n\}$, where each boundary is represented by the order number of the corresponding candidate point. As the scene boundary closest to an arbitrary candidate point i is one of two successive boundaries b_{k-1} and b_k surrounding this point so as $b_{k-1} \le i < b_k$, the likelihood of video feature v_i measured at point i given partitioning into scenes B can be written as

$$P(v_i \mid B) = P(v_i \mid b_{k-1}, b_k) = P(v_i \mid \Delta_i), \tag{5-21}$$

where Δ_i is the distance (measured in the number of shots) from point i to its closest scene boundary b_c defined as

$$\Delta_i = i - b_c, \tag{5-22}$$

$$b_c = \begin{cases} b_{k-1}, \text{ if } i - b_{k-1} \le b_k - i \\ b_k \text{ otherwise.} \end{cases} \tag{5-23}$$

We define the audio feature as a change in acoustic parameters measured between two contiguous windows of the fixed temporal duration. Therefore, we assume conditional dependence of this feature on the *time* distance to the closest scene boundary. Denoting the time of i-th candidate point as t_i, the temporal distance from point i to its closest scene boundary - as τ_i, we write the likelihood of audio feature a_i measured at point i as

$$P(a_i \mid B) = P(a_i \mid b_{k-1}, b_k) = P(a_i \mid \tau_i), \tag{5-24}$$

where

$$\tau_i = t_i - t_c, \tag{5-25}$$

$$t_c = \begin{cases} t_{b_{k-1}}, \text{if } t_i - t_{b_{k-1}} \le t_{b_k} - t_i \\ t_{b_k} \text{ otherwise.} \end{cases} \tag{5-26}$$

Taking into account expression (5-21) and (5-24), the likelihood of the total feature vector $D_i = \{v_i, a_i\}$ given partitioning into scenes B can be reduced to

$$P(D_i \mid B) = P(D_i \mid b_{k-1}, b_k), \tag{5-27}$$

We assume conditional independence of the components of D_i given B:

$$P(D_i \mid B) = P(v_i \mid B) P(a_i \mid B) = P(v_i \mid b_{k-1}, b_k) P(a_i \mid b_{k-1}, b_k). \tag{5-28}$$

If more observable data are available, expression (5-28) can be easily extended to include additional feature vector components.

We calculate likelihood values $P(v_i \mid \Delta_i)$ and $P(a_i \mid \tau_i)$ using the corresponding probability density functions (pdf) considered to be stationary (i.e. independent of time index i).

It is assumed that observable features are dependent on the closest scene boundary only if the distance to it is quite small, i.e. is lower than some threshold which is on the order of the length of the time windows used to calculate these features. This assumption facilitates to learn parameters of pdf estimates based on a set of learning data.

To evaluate the likelihood of the video coherence, we use a non-parametrical estimate of the corresponding pdf based on a Gaussian kernel and obtained for a set of pre-segmented ground-truth data. It is calculated separately for each possible value of the distance to the closest scene boundary Δ. We assume that this distance is limited by a range $[-n_1, n_2]$, where n1 and n2 are natural numbers of the order of number of the terms N included in expression (5-11). If it happens that $\Delta < -n_1$, we set $\Delta = -n_1$, and if $\Delta > n_2$, we set $\Delta = n_2$.

The likelihood of the audio dissimilarity feature is calculated from the joint probability as

$$P(a \mid \tau) = \frac{P(a, \tau)}{P(\tau)}, \tag{5-29}$$

where, as earlier, a stands for the feature value, τ - for the time distance to the closest scene boundary. We approximate the joint probability with a non-parametric estimate of pdf using a Gaussian kernel on a set of learning data. Just as for the visual feature, we limit the range of τ by a value having the order of duration of the neighboring time windows used to calculate the audio dissimilarity.

5.5.2 HMM Specification and Optimal Scene Boundaries Selection

We assume that a priori probability of the presence of a scene boundary at any candidate point is determined by the position of the previous scene boundary. Hence, each scene, beginning at a time t, can be described by a state of a hidden semi-Markov model, whose duration τ is chosen according to a probability distribution $p^t(\tau)$, which is supposed to be estimated from learning data. To detect scene changes as corresponding state transitions we need to assign to scenes different states of the model or use several states to model each scene. As this leads to an undesirable growth of the model, we propose to generalize it so as to make possible the use of only one common state. The resulting model is presented in Figure 5-10. To distinguish the transition from one scene to another, an auxiliary state q_{end} is used. This state does not emit observable data, but acts as an indicator of the transition and switches immediately to the regular model state 1. We define a transition indicator variable s_t which is set to 1 if a transition takes place at time t and 0 otherwise.

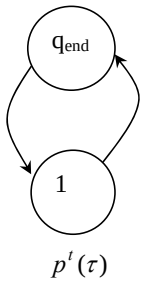


Figure 5-10. A generalization of a hidden semi-Markov.

In order to reduce the statistical dependency between contiguous feature vectors, we adopt an autoregressive model for the video coherence samples. Therefore the linear prediction error of the video coherence is assumed to be the visual component of the feature vectors. As our model has only one state which emits observable data, all the video coherence values are generated by a single autoregressive process. Therefore the linear prediction error can be obtained at the preprocessing stage and then be used as a regular feature. We don't apply an autoregressive model to the audio data, assuming that the time distance between contiguous audio features is quite large and their dependence is not so crucial.

The posterior probability of the presence of scene boundaries is calculated using the forward-backward procedure, which is rewritten for our model as follows. As earlier, we assume that there are T candidate points $\{1,2,...,T\}$ where a scene transition can occur, the first scene begins at time t=1, and the last scene ends at t=T. The forward variable α_t is defined as

$$\alpha_t = P(D_{1:t}, s_t = 1), \tag{5-30}$$

where $D_{1:t}$ denotes the subsequence of observable features $D_1, D_2, ..., D_t$. The variable is initialized as

$$\alpha_1 = 1. \tag{5-31}$$

For the subsequent time moments $t = 2,...,T$ we have the following induction:

$$\alpha_t = \sum_{\substack{1 \le k \le t-1 \\ k \ge t-\tau^k}} \alpha_k p^k(t-k) \prod_{i=k+1}^{t} P(D_i \mid s_k = 1, s_{k+1} = 0, ..., s_{t-1} = 0, s_t = 1), \tag{5-32}$$

where τ^k is the maximum possible scene duration. It is assumed in this expression that the feature vectors are conditioned only on positions k and t of the two surrounding scene boundaries, as it follows from (5-27). The probability of observing the whole sequence of feature vectors is written in terms of the α as

$$P(D_{1:T}) = \alpha_T. \tag{5-33}$$

The backward variable β_t is defined now as the probability of partial feature vector sequence $D_{t+1:T}$ given that a scene transition occurs at time *t*:

$$\beta_t = P(D_{t+1:T} \mid s_t = 1). \tag{5-34}$$

This variable is calculated recursively, initialized first as

$$\beta_T = 1 \tag{5-35}$$

and then by induction

$$\beta_t = \sum_{x=1}^{\min\{\tau^t, T-t\}} \beta_{t+x} p^t(x) \prod_{i=t+1}^{t+x} P(D_i \mid s_t = 1, s_{t+1} = 0, ..., s_{t+x-1} = 0, s_{t+x} = 1). \tag{5-36}$$

The posterior probability of a scene transition at time *t* is finally written as

$$P(s_t = 1 \mid D_{1:T}) = \frac{P(D_{1:T}, s_t = 1)}{P(D_{1:T})} = \frac{\alpha_t \beta_t}{\alpha_T}. \tag{5-37}$$

We assume that the minimum possible scene duration is limited by the value which exceeds the time length of the ambiguity window of 10 seconds (5 seconds in each direction) admitted for reference scene boundaries. At most only one scene boundary can correspond to a reference boundary under this assumption. Therefore, to provide the optimal values of recall and precision, we select *N* scene boundaries so as to maximize the expected number of the correct ones using expression (4-27), where *N* is calculated according to expression (4-12).

5.5.3 Scaling

It is easy to see that α_t, defined by expression (5-30), consists of the sum of terms which are written as

$$P(s_{1:t-1}, s_t = 1) P(D_i \mid s_{1:t-1}, s_t = 1), \tag{5-38}$$

where $s_{i:j}$, $i \le j$, denotes the subsequence of scene transition indicator variables $s_i, s_{i+1}, ..., s_j$. Since the likelihood of the feature vector $P(D_i \mid s_{1:t-1}, s_t = 1)$ often differs considerably from 1, each term takes the value too high or too low to be within the limits of the precision range of standard floating-point number representations; the same is true for variable α_t as well. To tackle this problem, log-values of this variable could be used instead. However, as expression (5-32) includes summation, this would lead to additional computational efforts required to transform and normalize the log-values to and from their regular representation at each step of the recursion. Therefore instead of the use of log-values we propose to perform the computation by applying a scaling procedure.

The scaling consists of multiplying the feature likelihood values by scaling coefficients c_t dependent on time index *t*. This multiplying does not change the posterior probability of a

scene transition $P(s_t = 1 | D_{1:T})$. Indeed, denoting the sequence $\{s_1, s_2, ..., s_T\}$ as *S*, this probability is written as

$$P(s_t = 1 | D_{1:T}) = \frac{P(D_{1:T}, s_t = 1)}{P(D_{1:T})} = \frac{\sum_{S \; s.t. \; s_t = 1} P(S) \prod_{i=1}^{T} P(D_i | S)}{\sum_{S} P(S) \prod_{i=1}^{T} P(D_i | S)}. \quad (5\text{-}39)$$

The multiplying of the data likelihood by an arbitrary $c_t \neq 0$ evidently does not change this ratio.

We choose the scaling coefficients so that the scaled version of α_t, denoted as $\hat{\alpha}_t$, becomes equal to 1. For this purpose we use first the recursion (5-32) to calculate α_t. Then we multiply the likelihood of the feature vector measured at time *t* by scaling coefficient c_t calculated as

$$c_t = 1 / \alpha_t . \quad (5\text{-}40)$$

It can be easily seen that $\hat{\alpha}_t$, calculated from expression (5-32), becomes equal to 1 and can be omitted in the following recursions both for β_t and α_t. The same scaling coefficient c_t is used at each subsequent recursion for $\hat{\alpha}_t$ and $\hat{\beta}_t$ given by expression (5-32) and (5-36), where the feature likelihood $P(D_i | S)$ is substituted by the value $\hat{P}(D_i | S)$ written as

$$\hat{P}(D_i | S) = c_t P(D_i | S) . \quad (5\text{-}41)$$

As it follows from expression (5-37), the posterior probability that a scene boundary is present at time *t* is written finally as

$$P(s_t = 1 | D_{1:T}) = \hat{\beta}_t . \quad (5\text{-}42)$$

5.5.4 Prior Probability Estimate

We assume that the duration of scenes has a stable probabilistic distribution at the domain of regular time (as opposed to the time measured in number of shot units whose duration varies in quite a large range) and does not depend much on a specific input video. Therefore the prior probability of scene transition $p^t(\tau)$ is calculated based on the probability density function (pdf) of scene duration, denoted as $p_s(\delta)$. To obtain the expression for the prior probability we first make two hypotheses.

According to the first hypotheses h_1 the prior probability of the presence of a scene transition at a candidate point *i* is proportional to pdf $p_s(\delta)$ and is written as

$$h_1 = \alpha \cdot p_s(\delta_i), \tag{5-43}$$

where δ_i is the time elapsed from the previous scene transition, α - normalizing coefficient. Denoting the index of the previous scene transition point as *j*, we choose α so that the total probability of scene transition at the subsequent candidate points $\{j+1, j+1, ..., T-1\}$ is equal to the integral probability of scene duration, i.e.

$$\alpha \cdot \sum_{i=j+1}^{T-1} p_s(t_i - t_j) = \int_{t_i}^{t_{T-1}} p_s(t - t_j)dt, \tag{5-44}$$

where t_i denotes the time of *i*-th candidate point. To reduce the computational burden, we limit the maximum possible scene duration by a value $\delta_{\max}$ (which is of about 5 minutes). Therefore the summation at the left side of expression (5-44) is stopped when $t_i - t_j > \delta_{\max}$ (in this case the integral of the right side is equal to 1).

The second hypothesis accounts for the variability of the shot duration which defines the time interval between contiguous candidate points so that the larger is this interval, the more probable is the scene transition. We suppose that a scene ends somewhere within a shot but is really observed only at the shot transition. Therefore the probability of a scene boundary at point *i* according to the second hypothesis is calculated as integral value

$$h_2 = \int_{t_{i-1}}^{t_i} p_s(t - t_j)dt. \tag{5-45}$$

To obtain the final estimate of the prior probability, we combine both the hypotheses into one using a weighted sum. The resulting prior probability $p^j(i-j)$ that a scene boundary is present at point *i* given that the previous scene boundary occurs at point *j*, $j<i$, is written as

$$p^j(i-j) = a \cdot h_1 + (1-a)h_2, \tag{5-46}$$

where *a* is a weight coefficient, $0 \le a \le 1$.

We choose *a* so as to obtain the best probability estimate based on the maximum likelihood criterion. For this purpose let's assume that there is a set of learning instances $L = \{(j_1, i_1, s_1), ..., (j_m, i_m, s_m)\}$ which are selected from manually marked up videos so that j_k are the candidate points where a scene transition occurs, while i_k are the subsequent candidate points where s_k indicate the presence or absence of a scene boundary given that the previous boundary occurs at point j_k. More strictly, these instances are defined as follows: j_k and i_k are all the pairs of candidate points so that a scene boundary is present at point j_k, $i_k > j_k$, and the time difference between points j_k and i_k does not exceed the maximum possible scene duration $\delta_{\max}$; s_k – a binary variable which is set to 1 if there is a scene boundary at point i_k and there no scene

boundaries between points j_k and i_k, otherwise this variable is 0. Assuming that each learning instance is drawn independently, the likelihood of the all data is written as

$$P(L \mid a) = \prod_{k=1}^{m} P(j_k, i_k, s_k \mid a). \tag{5-47}$$

It can be shown [MIT 96] that the maximization of this expression leads to the maximization of the cross entropy E written as

$$E = \sum_{k=1}^{m} s_k \ln p^{j_k}(i_k - j_k) + (1 - s_k)\ln(1 - p^{j_k}(i_k - j_k)). \tag{5-48}$$

The dependence between the averaged value of E and a, experimentally obtained for the total set of 4 ground-truth videos, is shown in Figure 5-11. As it can be seen from this curve, the best estimate of the scene boundary prior is obtained when hypothesis h_1 and h_2 are combined. The optimal value of a is chosen to be 0.4.

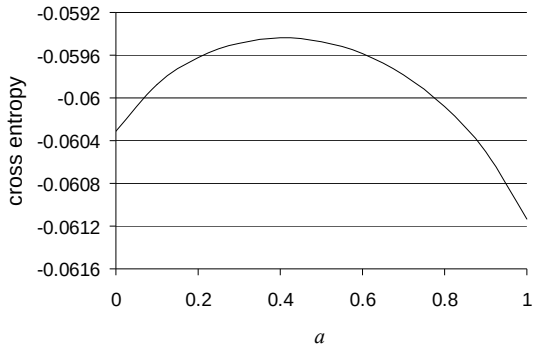


Figure 5-11. Cross entropy versus a.

The pdf of scene duration p_s is calculated using non-parametric estimate with Gaussian kernel and limit its range of definition by lower and upper boundaries. A sample plot of the resulting estimate, obtained for the 4 films of the ground truth is shown in.

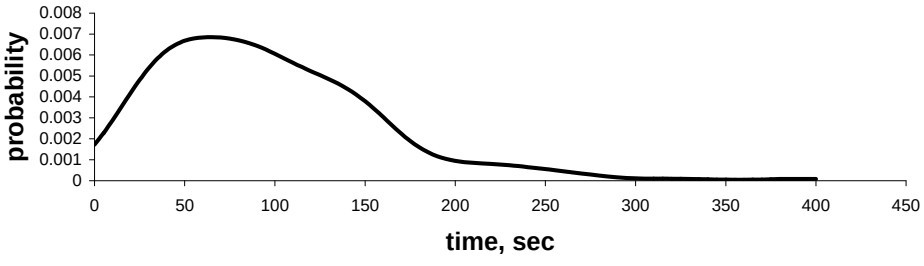


Figure 5-12. Scene duration pdf.

5.5.5 Experimental Evaluations

In this subsection we present the results of performance evaluations of the HMM-based segmentation technique. In these evaluations the observable data likelihoods were estimated non-parametrically at the domains where the learning samples provide a sufficient statistics. Outside these domains the likelihoods were extrapolated as constant values so as to diminish the influence of the outliers (this breaks the normalization of the likelihoods, which is not crucial as only their ratio is taken into account). Table functions with linear interpolation were used to speed up the calculation of likelihood values, so the segmentation procedure itself was considerably faster with respect to the computations required to extract feature vectors and took several seconds for an one-hour video on our Intel Pentium M 1.8 GHz computer. To provide the best performance, the video coherence feature was computed according expression (5-11) included 3 terms, i.e. in N was equal to 3; the video similarity was calculated within two contiguous groups of 5 shots adjoining the point under consideration.

The segmentation performance, obtained for each film from the ground-truth database, is given in Table 5-6. The probabilistic distributions for observable data and scene duration in this trial were obtained from the same set including all 4 films of the ground truth. The generalization capability of the algorithm was tested using cross validation where the learning set included three films and the test set consisted of the resting forth one until all data were tried. The results of the cross-validation tests are given in Table 5-7. The comparisons with Table 5-6 allows us to conclude that using of new data does not degrade the performance considerably. In the test results reported hereafter in this subsection we suppose that the learning is performed for all 4 films of the ground-truth.

Film	Precision, %	Recall, %	F1, %
A beautiful mind	58.0	72.3	64.3
Murder in the mirror	90.2	61.7	73.3
Si j'etais lui	53.2	57.9	55.5
When Harry met Sally	64.2	65.4	64.8
Total	63.7	64.5	64.1

Table 5-6. Performance of the HMM-based segmentation algorithm. The probabilistic distribution estimates were learned once for the same set of the 4 films.

Film	Precision, %	Recall, %	F1, %
A beautiful mind	54.7	72.3	62.3
Murder in the mirror	88.9	53.3	66.7
Si j'etais lui	49.3	57.9	53.2
When Harry met Sally	63.5	63.5	63.5
Total	62.0	60.2	61.1

Table 5-7. Results of the cross-validation tests for the HMM-based segmentation algorithm.

The capability of the HMM-based technique to fuse audio-visual data can be evaluated from Table 5-8 which presents the segmentation performance for 3 trials: first one is based only on visual data, the second – on audio dissimilarity only, the third trial fuses the audio and visual features. The comparisons with the results obtained for the rule-based technique (see Table 5-3) and the maximum likelihood ratio segmentation (see Table 5-5) shows that the HMM provides the better performance for visual data, which can be explained, in particular, by the fact that this model includes additionally the priory information about the scene duration. In contrast, the audio feature yields relatively low performance and does not contribute much when both audio and visual data are fused. One of the reasons of this is the neglect of the probabilistic dependencies between adjacent audio features, which, in particular, causes significant overestimates or underestimates of the posterior probability of scene transitions. So, the further improvements might include the better modeling of these dependencies.

Feature used	Precision, %	Recall, %	F1, %
Visual	66.2	60.3	63.1
Audio	34.3	35.5	34.9
Visual + audio	63.7	64.5	64.1

Table 5-8. Audio-visual data fusion capability of the HMM-based algorithm total for 4 films.

We have also tested our HMM for the conventional Viterbi algorithm which performs scene segmentation by finding the most probable sequence of scene transition indicator variables $S = \{s_1, s_2, ..., s_T\}$ for the entire set of candidate points. The results are reported in Table 5-9. Significant degradations can be remarked with respect to our segmentation technique (see Table 5-6) which, remind, is based on the optimality criterion aimed to maximize the performance

metric directly. The most considerable performance deterioration is observed for recall, as the Viterbi algorithm tends to produce long scenes, claiming scene boundaries only at the points where there is strong evidence of their presence. Another drawback of this algorithm is that the number of claimed boundaries cannot be changed so as to provide the desirable ratio between recall and precision.

Film	Precision, %	Recall, %	F1, %
A beautiful mind	43.8	21.6	28.9
Murder in the mirror	57.1	13.3	21.7
Si j'etais lui	50.0	28.1	36.0
When Harry met Sally	66.7	23.1	34.3
Total	52.1	21.4	30.3

Table 5-9. The performance of the Viterbi segmentation algorithm.

5.6 Sequential Segmentation Algorithm

According to our statistical approach a video is segmented in two stages: first the posterior probability of scene boundaries is computed at each candidate point, and only then the optimal boundaries are finally selected. In this section we derive and test a segmentation algorithm according to another statistical approach where scene boundaries are selected sequentially in one pass. The algorithm can be used in real-time systems where the result is obtained as new data available with a delay of the maximum possible scene duration.

5.6.1 Segmentation Principles

As earlier, we assume that feature vector D_i, measured at a time point i, is conditionally dependent from the position of the closest scene boundaries b_{k-1} and b_k which surround this point, i.e. expression (5-27) holds true. Furthermore, the posterior probability of a scene boundary b_k at point i assumed to be conditionally dependent solely on local feature vector D_i given the position b_{k-1} of the previous scene boundary. This assumption agrees with the intuition that evidence of the presence or absence of a scene boundary at an arbitrary point is determined by the feature vector measured at the same point. Indeed, this feature vector reflects the degree of change in the visual and audio environment of a scene and the larger is this change, the higher

is the probability of a scene change. Using Bayes rule, the posterior probability of *k*-th scene boundary at point *i* given b_{k-1} is written as

$$P(b_k = i \mid D_i, b_{k-1}) = \frac{P(D_i \mid b_{k-1}, b_k = i)P(b_k = i \mid b_{k-1})}{P(D_i \mid b_{k-1}, b_k = i)P(b_k = i \mid b_{k-1}) + P(D_i \mid b_{k-1}, b_k \neq i)P(b_k \neq i \mid b_{k-1})}$$

$$= 1/\left[1 + \frac{P(D_i \mid b_{k-1}, b_k \neq i)P(b_k \neq i \mid b_{k-1})}{P(D_i \mid b_{k-1}, b_k = i)P(b_k = i \mid b_{k-1})}\right], \ \forall i > b_{k-1}. \quad (5\text{-}49)$$

In expression (5-49) we further assume that the next scene boundary b_{k+1} takes place long time after boundary b_k, so that the likelihood of D_i given $b_k < i$ is always conditioned on b_k when computed according to expression (5-21) - (5-26). We denote this assumption as $b_{k+1} = +\infty$. It is also supposed that scene boundary duration is limited in time by a threshold value $\delta_{\max}$. Then a possible position of *k*-th scene boundary is limited by a value m_k defined as

$$m_k = \max\{l \mid t_l - t_{b_{k-1}} \leq \delta_{\max}\}. \quad (5\text{-}50)$$

Under these assumptions expression (5-49) is continued as

$$P(b_k = i \mid D_i, b_{k-1}, b_{k+1} = +\infty) =$$

$$= 1/\left[1 + \frac{\sum_{l=b_{k-1}+1}^{i-1} P(D_i \mid b_k = l, b_{k+1} = +\infty)P(b_k = l \mid b_{k-1}) + \sum_{l=i+1}^{m_k} P(D_i \mid b_{k-1}, b_k = l)P(b_k = l \mid b_{k-1})}{P(D_i \mid b_k = i)P(b_k = i \mid b_{k-1})}\right]. \quad (5\text{-}51)$$

We propose to segment an input video into scenes sequentially, choosing each next scene boundary based on the position of the previous one. So, the video can be segmented in real-time with a time delay of the order of the maximal scene duration $\delta_{\max}$. Knowing the position of scene boundary b_{k-1}, we select the next boundary b_k using the posterior probability estimated at each candidate point *i*, $i > b_{k-1}$, on time length $\delta_{\max}$ according to expression. In this paper the boundary b_k is placed at the point of the maximal probability, as such a decision criterion has appeared to work well in experimental evaluations. This criterion is based on a relative comparison of the evidence of a scene boundary at each point under consideration provided by the feature vector measured at the same point. In this manner, the resulting segmentation procedure resembles the conventional techniques which pronounce scene boundaries at the points of local extremum of some visual or audio similarity curve, expression (5-51) being considered as a way of fusing multiple data into one cumulative measure. Four posterior probability curves along with audio dissimilarity and video coherence curves obtained for a ground-truth film are depicted in Figure 5-13. The probability curves are shown partly overlapped; each curve begins at the first candidate point inside a scene, achieves the global

maximum at the point of transition to the next scene and is interrupted at the middle of the next scene (in order not to encumber the figure). As it can be seen from the figure, the probability curves have peaks which are better pronounced and, hence, better detectable with respect to the feature curves.

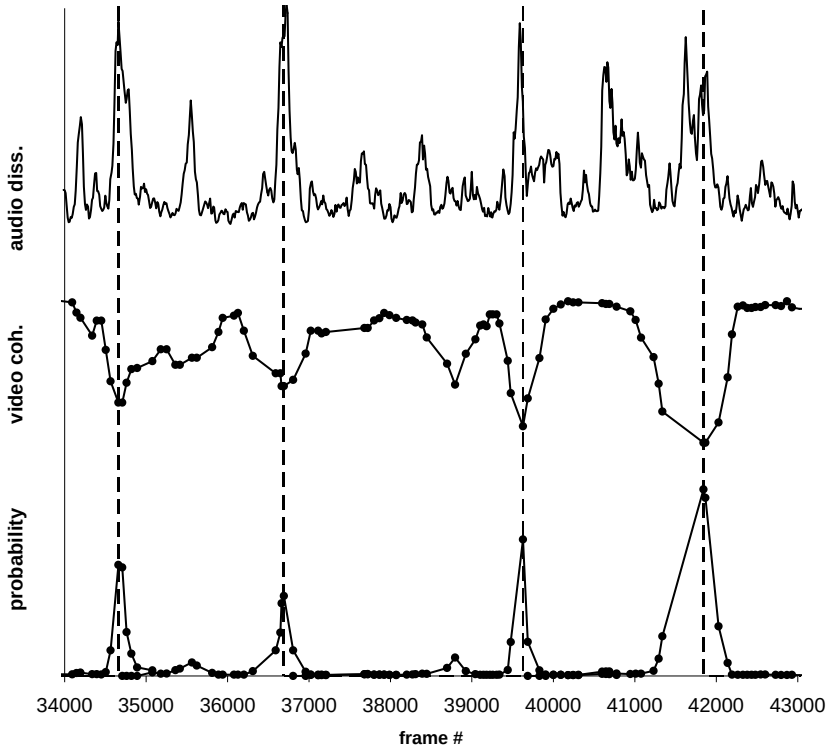


Figure 5-13. Audio dissimilarity (upper curve), video coherence (middle curve) and scene boundary posterior probability in sequential segmentation approach (partially overlapping curves in the bottom) versus frame number. Vertical dashed lines delimit scenes.

It is assumed that in expression (5-51) the prior probability $P(b_k \mid b_{k-1})$ of scene boundary b_k is determined by the duration of the scene which ends up at this boundary and is calculated using pdf of scene duration p_s as

$$P(b_k \mid b_{k-1}) = \alpha p_s(t_{b_k} - t_{b_{k-1}}), \tag{5-52}$$

where t_i is the time of candidate point i. Normalizing coefficient α can be omitted when this expression is substituted in expression (5-51) as only the ratio of probability values is taken into account. We do not take into account the shot duration in this expression, as it was done in our

HMM-based segmentation technique, since this would make the probability curve more irregular because of the significant difference in shot length.

We deliberately include only one local feature vector D_i in expression (5-51) and exclude surrounding data from consideration. Otherwise there would be a need to treat properly the strong dependence which usually exists between contiguous observable data. This would complicate the proposed approach and would possibly require more learning data. Experimental tests on a more complicated model which includes the complete set of observable data up to the point under examination, much as the model proposed in [VAS 97] for the task of shot segmentation, suggest that simple neglect of this dependence in such a model degrades considerably the segmentation performance, let alone the increase of the computational complexity. The problem of dependence between feature vectors is avoided in our model, as the single feature vector D_i in expression (5-51) is usually placed far enough from boundary b_{k-1} at the most points under examination and, thus, does not strongly depends on the feature vector measured at this boundary.

5.6.2 Final Algorithm

The final segmentation algorithm used in this work is resumed as follows.

- Segment an input video into shots and assign candidate points of scene boundaries to be the shot transition moments. Estimate feature vector D_i at each point i.
- Place the initial scene boundary b_0 at the beginning of the first scene (which is supposed to be given). Select recursively each subsequent scene boundary b_k based on the position of the previous one b_{k-1} through the following steps:
    - Calculate the posterior probability of k-th scene boundary at each candidate point i of set $\{b_{k-1}+1,...,m_k\}$ according to expression (5-51), where m_k is defined by expression (5-50) and is limited by the last candidate point.
    - Place the next scene boundary b_k at the point of the highest posterior probability.
    - If a stopping criterion is fulfilled, exit the algorithm.

The stopping criterion is used mostly to keep inside the narrative part of the input video. We suppose that the position of the last scene boundary is given and the stopping criterion is fulfilled when the current scene boundary b_k appears to be closer in time to the last scene boundary than a predefined threshold value which is approximately equal to the mean scene duration.

5.6.3 Experimental Evaluations

In this subsection we report the results of experiments designed to test the proposed video scene segmentation algorithm. As earlier for the HMM-based segmentation technique, the feature

likelihoods and the scene duration pdf were estimated non-parametrically using a Gaussian kernel. To provide the best performance, in all the experiments the video coherence feature was computed according expression (5-11) included 3 terms, i.e. in N was equal to 3; the video similarity was calculated within two contiguous groups of 5 shots adjoining the point under consideration.

Segmentation performance of the proposed sequential segmentation algorithm relative to different films entered into our database is compared in Table 5-10. The feature likelihoods and the scene duration pdf were estimated on the learning set including all 4 films. The highest integral performance F1 for film "Murder in the mirror" was caused mainly by the most stable behavior of the video coherence curve as the scenes were shot by relatively slow-moving or static cameras. In contrast, the outsider film "Si j'etais lui" was characterized by intensive camera movements. A reason of a relatively low performance for film "A beautiful mind" was less accurate shot segmentation for gradual shot breaks which merged sometimes shots contiguous to a scene boundary. Comparison with the results given in the sections above (Table 5-2, Table 5-5, Table 5-6) allows us to conclude that the sequential segmentation approach has the best performance measured by both the precision and recall.

Film	Precision, %	Recall, %	F1, %
A beautiful mind	67.7	67.7	67.7
Murder in the mirror	88.9	66.7	76.2
Si j'etais lui	66.7	63.2	64.9
When Harry met Sally	69.8	71.2	70.5
Total for 4 films	72.4	67.1	69.6

Table 5-10. Performance of the sequential segmentation algorithm for different films.

In order to evaluate the generalization capability of the segmentation approach learned on a set of pre-segmented data, the cross-validation tests were carried out. The learning set included three films and the test set consisted of the resting forth. The overall results for all 4 films are given in Table 5-11. Three trials were made: the first one did not used cross-validation at all, serving as a reference; the second used a separate set to learn only the pdf estimates for the audio and visual features while the scene duration pdf was estimated on a common set including all 4 films; the third trial supposed separate learning and test sets for all the pdf estimates. As it follows from Table 5-11, our segmentation approach does not suffer much from parameters over-fitting, providing quite a general model for video scene segmentation. The perceptible sensitivity to the estimate of scene duration pdf suggests importance of taking into account of prior

information about scene duration. The results given below in this section assume the same learning and test set which includes all 4 films of the ground truth.

Using cross-validation	Precision, %	Recall, %	F1, %
Non	72.4	67.1	69.6
For the feature pdf only	69.9	67.5	68.7
Total for the feature pdf and the scene duration pdf	67.6	65.0	66.2

Table 5-11. Performance of the sequential segmentation algorithm in cross-validation tests.

The capability of our sequential segmentation approach to fuse audio-visual features can be revealed from Table 5-12, where the first row presents the segmentation performance when only the visual feature was used, the second one gives the performance only for the audio feature and the third – for both the features. As it follows from the table, fusing the visual and audio features enhances both the recall and precision.

Feature used	Precision, %	Recall, %	F1, %
Visual	61.7	64.1	62.9
Audio	39.9	48.7	43.8
Visual + audio	72.4	67.1	69.6

Table 5-12. Performance of the sequential segmentation algorithm for audio-visual feature fusion.

As for computational time required by our sequential segmentation algorithm, it is quite fast given that audio-visual features are pre-computed and takes less than a second on our Intel Pentium M 1.8GHz computer for one film. This is because the computational complexity is approximately linear with respect to the film length due to limited time search for each scene boundary. The main computational burden for a raw video file stems from its decoding and feature extraction which, though, can be done in real time without much optimization for MPEG 4 video format.

5.7 Conclusions

In this chapter we have adopted our stochastic approach to the task of narrative video segmentation into semantic scenes. Several particular segmentation techniques were derived based on different assumptions about the feature dependencies and the priori distribution of scene duration. Because of the lack of common benchmarking data, the performance of the proposed techniques was tested comparatively using our database of 4 ground-truth videos. To

reliably detect scene boundaries, we proposed two informational sources providing evidence about possible scene transitions – video coherence and audio dissimilarity. It was experimentally shown that our video coherence measure leads to a better segmentation performance with respect to the conventional measure which is based on a short memory model.

While the conventional rule-based segmentation algorithm attains the performance improvements when fusing multiple data sources, it suffers from excessive coarseness and is too sensitive to the chose of its threshold parameters. The multi-modal data are fused more effectively in our statistical maximum likelihood ratio algorithm. The cross validation tests showed that this algorithm generalizes learning data quite well and can be applied to new data without significant losses in performance. Further improvements were made in our HMM-based segmentation algorithm which models the dependences of the observable data from the scene boundary positions and takes into consideration the non-uniform priori distribution of scene duration. Being based on our optimality criteria, this algorithm has better performance than the conventional Viterbi procedure. We also proposed a statistical algorithm, called a sequential segmentation one, which segments video in one pass, selecting scene boundaries sequentially as new data are available, and, hence, is suitable for real-time applications. The algorithm is not sensitive to the statistical dependencies between adjacent feature vectors and has the best performance.

6 Video Summarization

6.1 Introduction

Being compact representations of the content, video summaries provide a fast way to get acquainted with the main points at a glance, without the need to see the entire video. The input video can be summarized at a whole and can be used, for example, in the form of trailers to allow users to choose quickly an interesting movie from a huge collection, or in the form of personalized summaries for mobile devices. Alternatively, video summaries can be used together with content tables as convenient interface for navigation within a video, providing compact visual representation of the semantic units.

Video summary can be produced in the form of a static storyboard called sometimes a pictorial summary. In this case it is represented as a collection of still images arranged in time order to convey the highlights of the content. These images can be simply the most representative frames called key-frames that are extracted from the video stream. In the more complicated case they are produced synthetically, e.g. mosaics which represent the panoramic views capturing the moving of the camera [PEL 00]. The static storyboards, being rendered on a screen, may provide to a user the possibility to grasp at a glance the concise information about the whole content or the moments of interest. Another commonly used style of video summary is a video skimming which is a trimmed video consisting of a collection of image sequences. This type of summary usually requires more memory space and longer time to be viewed but its advantage is that it conveys the audio information and the motion as well.

In this chapter we propose a video summary using a shot-based approach that allows generating both a static storyboard and a video skim in the same manner. The video is decomposed into consecutive shots and the most important of them are left to compose a summary. For each shot we choose the most representative frame (i.e. key frame) which is used to build a pictorial summary and to calculate some features like the similarity of the different shots. This approach is justified when the shots are static enough and are shot by a still camera.

A lot of summarization approaches proposed by today are rule-based. They use sets of rules to identify important moments in the video combining the different characteristics extracted from both the audio and video streams [SMI 98, LIE 97]. Pattern recognition algorithms are often employed to detect the events of interest to be included in the summary, especially in the domain-specific tasks such as sport video summarization [MUR 03]. The drawback of these approaches is that they qualitatively select the important moments and thus do not allow tuning

of the compression ratio. This is not the case for another class of summarization methods which use mathematical criteria to quantitatively evaluate the importance of video segments. For example, Uchihashi et al. [UCH 99] numerically estimate the importance score of video segments based on their rarity and duration; Yihong Gong and Xin Liu [YIH 00] use for this purpose singular value decomposition.

We propose a generalized quantitative criterion that includes some quantitative parameters and simple rules defined by a user as desirable constraints on different features of video segments: their classification into day or night, exterior or interior, shot duration, sound classification into speech, noise, silence and music etc. So, our method can be considered as the generalization of the rule-based approaches as well. It is however more flexible and allows customizing to the needs of a user. Indeed, a video summary is generated according to the user's preferable configuration of constraints and the desirable compression ratio or/and the threshold set on the importance scores.

The using of a global arrangement of the shots of a video according to their importance score leads sometimes to the fact that several important semantic segments are not represented in the summary at all. To overcome this we additionally propose a summarization approach that prevents the semantic structure of the video that consists of the higher level segments than shots. In this approach each higher level segment (scenes in our case) is given the equal opportunity to be presented in the target summary. This type of summary we will call hereafter a "digest".

6.2 Summarization Principles

6.2.1 System Architecture

The general architecture of our summarization system is shown in Figure 6-1. To build a summary of a video, the shot segmentation is first applied to its frame sequence. Then the different features describing the shots are computed. For the video stream these features currently are: time of a day for outdoor shots – day or night – which is determined using percentage of the illuminated parts in the key frame image [MAH 02]; place of the action – exterior or interior – that is defined based on color temperature of the selected parts of the key frame [MAH 01]; percentage of the human skin pixels calculated based on the pixel spectral distribution [HAM 03]; clustering of the shots based on the color histogram dissimilarity described above; average quantity of motion. The audio stream of the video is used to calculate the expected duration of the semantic segments in the shots (speech, music, silence, and noise) based on the approach described in [HAR 03b]. The multiple features of each shot are then fused into the importance score measure according to the user preferences. The target summary is built

from the shots that have the maximum score until the compression ratio limit is exceeded or shots having acceptable score are exhausted. The details of the score assignment are given further in this section.

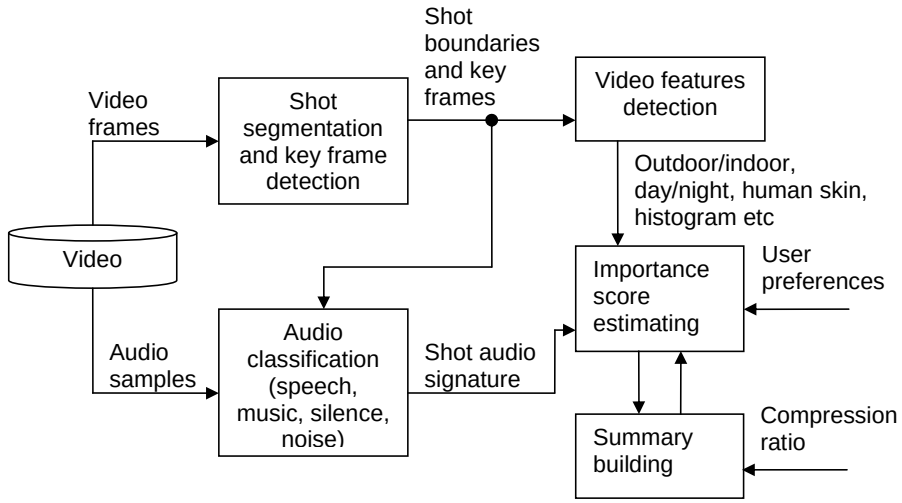


Figure 6-1. Architecture of the summarization system.

6.2.2 Importance Score Estimation

It seems difficult to propose a numerical importance score estimation approach that would be quite general to comprise all conceivable combination of the shot features on the one hand, and simple and expressive to be easily tuned by a user on the other hand. The possible decision is to formulate the problem as the function approximation task and to use the automatic learning techniques to tune the coefficients of the function representation formula. In this case, however, the user will have to provide the learning set each time when he decides to accustom the system to his specific needs. The problem of the most appropriate function representation remains anyway.

It is often the case when the user finds some difficulties in numerical estimating of the importance score, but he can express his wishes concerning the desirable content of the video summary in the form of simple assertions (negative and positive) on the features. That is why it seems reasonable to combine these assertions into the importance estimation formula in such a simple way as a weighted sum. In this case the score represents a simple calculation of the points gained (or lost) by the positive answers to the simple tests. Similar technique is used to build the summary of the key-word annotated video by the video semantic summarization systems of IBM

[IBM], where the tests check the presence of the key-words in the annotation. Xingquan Whu et al. [XIN 03] put into a video summary the shots that have the maximum number of key-words (the idea is that these shots are most representative). Their approach is a particular case of our concept where tests check the presence of the key-words and all the weights are positive and equal.

In our case the user formulates his preferences specifying the inequalities ("greater", "less" or "equal" relations) applied to the wide-range shot parameters (duration of shots, percent of human skin, quantity of movement, expected duration of sound semantic segments) and desirable classification result (day/night, exterior/interior). Each such a preference number *i* we represent by the binary value denoted as b_i which is set to 1 if the corresponding constraints is true and to 0 otherwise. We also give to the user opportunity to formulate his favor to some of the numerical shot features by adding terms which are functions of these features. We denote these terms as f_i. The resulting expression for importance score *S* estimation is written as

$$S = \sum_i w_i^b b_i + \sum_i w_i^f f_i, \tag{6-1}$$

where w_i^b and w_j^f are the weights of the corresponding terms b_i and f_j.

In this work the terms *f* are:

- The "coverage" of the shot which express the relative length of the video belonging to the same cluster. We define it as $\log(\frac{\sum_{i \subset C} Li}{L_e})$, where L_i denotes the duration of the shot *i*, *e* – the index of the estimated shot, *C* – set of the shot numbers that belongs to the same cluster as the shot *e*.
- The "originality" of the shot which is decreasing function of the number of shots belonging to the same cluster which are already included in the summary. In our work it is a reverse value of this number. As this term is depended of the process of the summary building, the connection between the stages of importance score estimation and of summary building in Figure 6-1 is bidirectional.

These terms may be used to select the original shots of the clusters and the most repetitive clusters as the most representative ones. Note that they have approximately the same value area as the binary values, e.g the minimum possible value of the coverage term is 0 (for the unique shots) and the maximum value is limited due to the logarithm function. So, they give approximately the same contribution to the importance score.

6.2.3 Video Digest

The summarization method described above selects the shots globally for the whole video and does not take into consideration its high level semantic structure. Therefore, some high level semantic segments may not be presented in the summary at all, especially when the high compression is desirable. We propose to a user an additional summarization approach aimed to build a digest – the summary which, in fact, is compounded of the summaries of each its high level semantic segments. To each high level semantic segment, which we reference hereafter in this chapter as a scene, we give the equal opportunity to be presented in the digest. The following algorithm is used:

1. Perform shot segmentation and their features extraction according to the scheme described above.
2. Perform scene segmentation and collect the detected scenes (represented by groups of corresponding shot descriptors) into the set SCENES.
3. **While** the set SCENES is not empty **do**:
   a. **For** each scene in the set SCENES **do**:
      i. Extract the shot with the highest importance score from the scene and add it to the set CANDIDATE_SHOTS.
      ii. If the scene does not contain the shots with the importance score higher then the score threshold, remove the scene from the set SCENES.
   b. **For** each shot extracted in the decreasing order of its importance mark from the set CANDIDATE_SHOTS **do**:
      i. Add the shot to the digest.
      ii. If the size limit of the digest is achieved, exit.

6.3 Implementation and Experiments

The summarization algorithms described above has been implemented as a computer program that was used for their experimental evaluations. Three windows of the graphic user interface (GUI) provided by the program are shown in Figure 6-2. The window of the player allows to a user to view a summary or a digest (and the original video as well) by playing the corresponding video skim clip using rewind and positioning controls for navigation along the time line. The window of the key frames control presents a summary or a digest as a static storyboard that contains the key frames of the video shots. The window of the filter allow to a user to specify his preferences concerning the shot features which are used to calculate the importance score of the video shots. These preferences are expressed by imposing “less”, “equal” and “greater” relations on the shot features and by setting of the corresponding weights. In the window of the filter the

user can set a threshold value on the importance score as well so that to leave in the summary only the most important video segments.

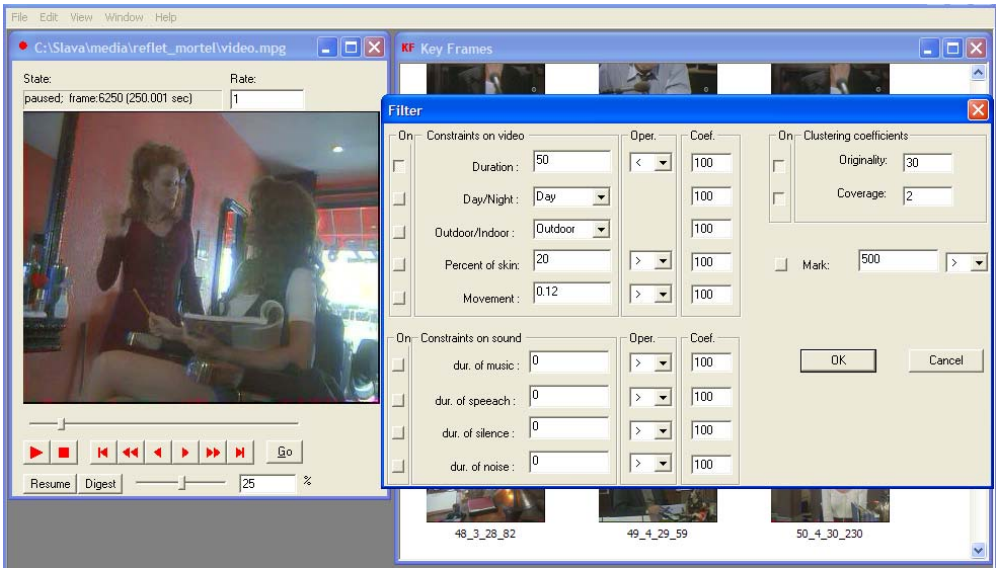
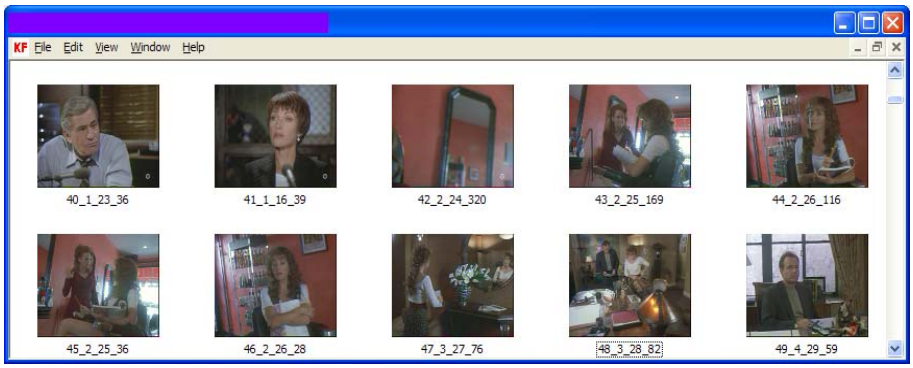


Figure 6-2. GUI of the summarization program.

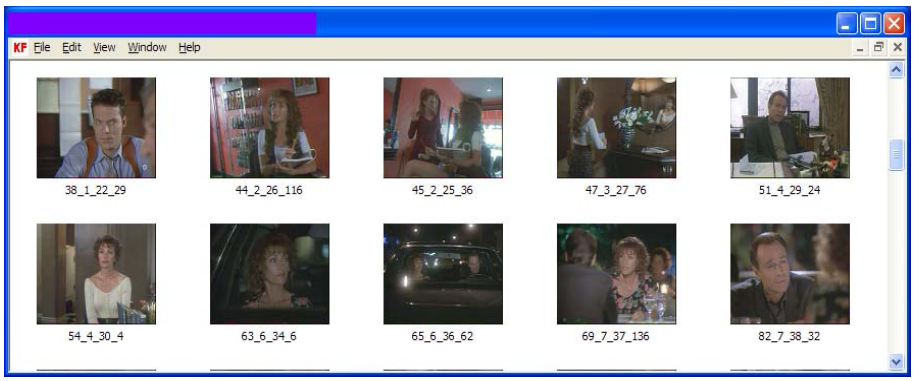
The summarization program makes a summary or a digest from a video file at two steps. First, the video file is segmented into the shots and their audio and visual features are extracted, i.e. the stages shown before the importance score estimation in Figure 6-1 are executed. The shot grouping into scenes are implemented as well, and the result is saved to a persistent storage. This step is slow enough and, therefore, is executed only once. The summary or the digest is built at the second step which is very fast (it requires less than a second for a one-hour video on a modern personal computer) and may be executed many times without making the user to wait. As the configuration of the user's preferences influences only on this second step, he can execute it many times to interactively select the best combination of his preferences for the loaded video.

A lot of configuration of the preferable constraints can be proposed to adapt the system to specific needs. The user can highlight emotional episodes in the film choosing the shots containing musical tracks, dialog scenes including long speech and a portion of skin, try to select "erotic" shots containing a lot of skin etc. At the same time he can suppress some undesirable moments, e.g. annoying commercial inserts choosing the shots with long duration and high coverage (because of the fact that the commercials are often compounded of the short and non-repetitive shots).

a)



b)



c)

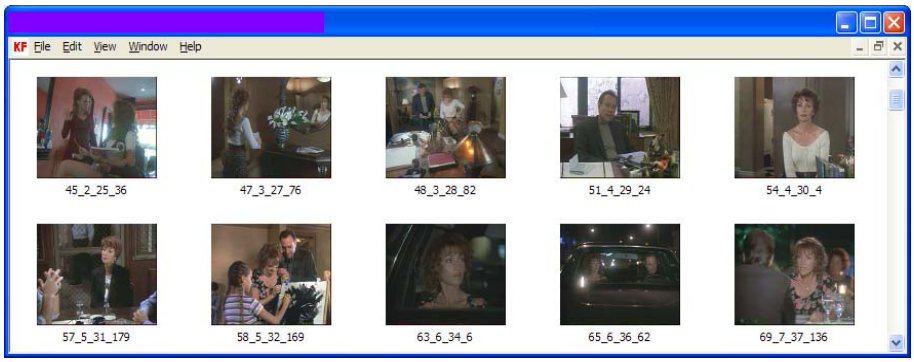


Figure 6-3. Storyboard fragments for an original video (a), its summary (b) and digest (c). Their images are the shot key frames. Three first numbers in the textual labels are the ordinal shot number, the scene number and the cluster number.

Selecting the "originality" and "coverage" terms the user can build a "semantic" summary briefly depicting the main shot types of the longest episodes. Figure 6-3b shows a static storyboard of such a "semantic" summary of an excerpt of a detective film captured from one of the French TV channel. The originality term weight was set to 30, the coverage term weight – to 2; the minimum shot duration was limited by 2 second with the weight set to 100; the compression ratio was set to 15%. Comparing this summary with the original video (Figure 6-3a)

we can see that it is capable to represent more semantic episodes on the same screen surface. Figure 6-3c shows a storyboard for a digest corresponding to the same part of the video. It rigorously prevents the semantic structure of the video, uniformly representing all the scenes.

6.3.1 Conclusions

In this chapter we have proposed a versatile approach which can be used to create summaries that are customizable to specific user's preferences to different type of video. A high versatility of the approach is based on a unified importance score measure of video segments which fuses multiple features extracted from both the audio and video streams. This measure provides the possibility to highlight the specific moments in a video and at the same time to select the most representative video shots using the "coverage" and "originality" terms. Its coefficients can be interactively tuned due to a high computational speed of the approach.

Additional terms can be easily added to the importance score estimation formula to extend our approach. For example, they might be assertions of a new form concerning the video shot features or additional features not mentioned in this work. Our digest building algorithm can be extended as well to prevent the structure of the video on the levels higher than scenes.

7 Conclusions and Future Work

Automatic video segmentation into semantic units is important to organize an effective content based access to long video. The review of related work in the field of semantic video segmentation has revealed a large diversity of the processing techniques stemming from the variety of genres and sub-genres of video; this is especially the case for sports programs where domain-specific fine-tuned event detectors are often applied. In spite of this, we could notice that much of this work is based on the common idea of using production rules that are followed during creation of video. In this thesis we proposed several segmentation techniques relying on common characteristics of video stemming from production rules by the following reasons. First, they provide us with quite a general basis to deal with the diversity of video properties in a unified fashion. Second, such characteristic can be reliably detected using common signal processing techniques that may require some learning to be adapted to a particular type of video. Moreover, instead of detection of semantic segments of just one or several types, that is often the case in the related work, in this thesis we aimed at reconstructing the total content structure of video.

In the case where an input video has a well-defined temporal content structure whose segments can be unambiguously related to mid-level events, we proposed a deterministic approach which is based on a finite state automaton. This approach provides a regular basis which allows one to formulate video content parsing rules as grammar constraints and feature templates that control transitions between semantic segments. It is suitable for video having complex hierarchical content structure for which reliable feature templates can be specified. In this thesis we adopted and tested this approach for the task of tennis video segmentation. The resulting segmentation technique is based on production rules that are typically employed to convey semantic information to a viewer, such as specific views and score boards in tennis broadcasts. In this technique we used our notion of a tennis content structure to select unique template of events that indicate transitions to semantic segments of each type. These events along with grammar restrictions drive the parsing process. The advantage of our approach is in its expressiveness and low computational complexity. Moreover, the experimental evaluations showed quite high segmentation accuracy, especially when high reliability of event detectors was provided.

For the task where sufficient learning data can be provided, we proposed a statistical segmentation approach. Treating an input video in a probabilistic manner we can take into

account “soft” grammar constraints imposed on the semantic structure and expressed in the form of probability distributions. Moreover, the multiple keys, being considered as statistical variables, can be more easily fused into one, more reliable decision in the case of their collisions. In contrast to the common statistical approach which selects the single best model of the whole video, in our approach we claim segment boundaries so as to maximize the performance metrics directly. The approach is based on the posterior probabilities of the boundaries estimated at each candidate point. These probabilities can be estimated in different ways, depending on the particular model of the video. In particular, we adopted the theory of hidden Markov models and their extensions and considered a video as a stochastic automaton – statistical generalization of the deterministic finite state machine.

We adopted our stochastic approach to the task of narrative video segmentation into semantic scenes. Several particular segmentation techniques were derived based on different assumptions about the feature dependencies and the priori distribution of scene duration. Experimental evaluations showed that the multi-modal data are fused more effectively in our statistical approach with respect to the conventional rule-based one. Based on the cross validation tests we also showed that the derived algorithms generalize learning data quite well and can be applied to new data without significant losses in performance. As for our HMM-based segmentation algorithm, the tests allowed us to conclude that the use of our optimality criterion leads to significantly better segmentation performance than the conventional Viterbi procedure.

In addition to the video segmentation, we also proposed a versatile approach to the video summarization task which is customizable to specific user’s preferences to different type of video. A video summary can have an independent meaning aimed to quickly get acquainted a viewer with the content of video or it can be generated for each semantic segment of a content table forming so called digest. Pictorial digests provide a convenient interface for navigation with content tables where each unit is visually represented with one or just several key frames. The high versatility of the approach is based on a unified importance score measure of video segments which fuses multiple features extracted from both the audio and video streams. The coefficients of this measure can be interactively tuned due to a high computational speed of the approach.

Further improvements of the proposed techniques could be done in several directions. To provide higher performance, we could extend the particular applications by adding new features. For example for the tennis segmentation these features could include additionally the results of rocket hits detection, time constraints and the output of speech recognition. Additional useful information for the narrative video segmentation could be provided by automatic person

tracking, as the same scene usually includes the same personages. The inclusion of new features could require dealing properly with possible dependencies between them, as currently in our statistical approach we fuse the multiple features assuming that they are independent. In addition, the currently used semantic structure could be extended so as to contain a larger variety of semantics which could provide additional possibilities for content based navigation. For instance, the points of tennis video could be split into several classes such as rallies, missed first serve, ace or replay. Also we are going to apply our approach to other types of video, e.g. sports broadcasting, news programs or documentary video.

References

[ALA 01] Alatan A.A., Akansu A.N., Wolf W., “Multi-Modal Dialogue Scene Detection Using Hidden Markov Models for Content-Based Multimedia Indexing”, Multimedia Tools and Applications, Vol. 14, No. 2, pp. 137-151, 2001.

[ALL 83] Allen J.F., “Maintaining Knowledge about Temporal Intervals”, Communications of the ACM, Vol. 26, No. 11, pp. 832-843, 1983.

[ARD 00] Ardebilian M., Chen L., Tu X.W., "Robust Smart 3-D Clues based Video Segmentation for Video Indexing", *Journal of Visual Communication and Image Representation*, Vol.11, Numéro 1, pp.58-79, Mars 2000.

[BAK 76] Bakis R., “Continuous speech recognition by statistical methods”, *Proc. ASA Meeting*, Washington DC, 1976.

[BEN 99] Benjio Y., “Markovian Models for Sequential Data”, *Neural Computing Survey*, Vol.2, pp. 129-162, 1999.

[BIM 95] Bimbot F., Magrin-Chagnolleau I., Mathan L., “Second order statistical measures for text-independent speaker identification”, *Speech Communication*, Vol. 17, No. 1-2 , 177-192, 1995.

[BOG 00] Boggs J.M., Petrie D.W., “The art of watching films”, Mayfield Publishing Company, Mountain View, CA, 5th edition, 2000.

[BON 93] Bonafonte A., Ros X., Marino J.B., “An Efficient Algorithm to Find the Best State Sequence in HSMM”, *Proc. of Eurospeech*, pp. 1547-1550, 1993.

[BON 96] Bonafonte A., Vidal J., Nogueiras A., “Duration Modeling with Expanded HMM Applied to Speech Recognition”, *Proc. Int. Conf. on Spoken Language Processing*, USA, pp. 1097-1100, 1996.

[BOR 97] Bordwell D., Thompson K., “Film Art: An Introduction”, 5th ed. New York: McGraw-Hill, 1997.

[BOR 96] Boresczky S., Rowe, L.A., “A comparison of video shot boundary detection techniques”, *Proc. of the SPIE Conference on Storage & Retrieval for Image and Video Databases IV*, pp.170-179, 1996.

[BOR 98] Boreczky J., Wilcox L., “A Hidden Markov Model Framework for Video Segmentation Using Audio and Image Features”, Proc. IEEE Conf. on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3741-3744, 1998.

[BUI 01] Bui H., Venkatesh S., West G., “Tracking and Surveillance in Wide-Area Spatial Environments Using the Abstract Hidden Markov Model”, *Int. J. of Pattern Recognition and AI*, 2001.

[CAO 03] Cao Y., Tavanapong W., Kim K., Oh J., “Audio Assisted Scene Segmentation for Story Browsing”, *Proc. of International Conference on Image and Video Retrieval*, pp. 446-455, 2003.

[CHA 01] Chang S.-F., Zhong D, Kumar R, “Real-Time Content-Based Adaptive Streaming of Sports Videos”, *Proc. IEEE CBAIVL*, Hawaii, pp.139-146, December 2001.

[CHA 02] Chang P., Han M., Gong Y., “Extract highlights from baseball game video with hidden Markov models”, *Proc. IEEE ICIP*, 2002.

[CHE 04] Chen Jianyun, Li Yunhao, Lao Songyang, Wu Lingda, “A Unified Framework for Semantic Content Analysis in Sports Video”, *Proc. of ICITA*, pp.149-153, 2004.

[CHE 98] Chen S.S., Gopalakrishnan P.S., “Speaker environment and channel change detection and clustering via the Bayesian Information Criterion”, *DARPA Speech Recognition Workshop*, 1998.

[CHE 02] Chen S.C., Shyu M.L., Liao W., Zhang C., “Scene Change Detection by Audio and Video Clues”, *Proc. of IEEE ICME*, pp. 365-368, 2002.

[COV 03] Cover T., Thomas J., “Elements of Information Theory, Wiley Series in Telecommunications”, *John Wiley and Sons*, 2003.

[CRY 88] Crystal T.H., House A.S., “Segmental Durations in Connected Speech Signals: Current Results”, *Journal of Acoustic Society of America*, Vol. 83, No.4, pp. 1553-1573, 1988.

[DIM 00] Dimitrova N., Agnihotri L., Wei G., “Video Classification Based on HMM Using Text and Faces”, *European Signal Processing Conference*, Tampere, Finland, 2000.

[DIZ 01] Di Zhong, Shih-Fu Chang, ”Structure Analysis of Sports Video Using Domain Models”, *Proc. IEEE ICME’01*, Japan, pp.182-185, August 2001.

[DON 01] Dong Zhang, Wei Qi, Hong Jiang Zhang, “A New Shot Boundary Detection Algorithm”, *IEEE Pacific Rim Conference on Multimedia*, pp.63-70, 2001.

[DUD 73] Duda R.O., Hart P.E., “Pattern Classification and Scene Analysis”, Wiley, New York, 1973.

[EIC 99] Eickeler S., Muller S., “Content-Based Video Indexing of TV Broadcast News Using Hidden Markov Models”, *IEEE ICASSP*, USA, pp. 2997-3000, 1999.

[EKI 03] Ekin A., Tekalp A.M., Mehrotra R., “Automatic Soccer Video Analysis and Summarization”, *IEEE Trans. on Image Processing,* Vol. 12, pp. 796-807, July 2003.

[GUI 04] “Guidelines for the TRECVID 2004 Evaluations”, in http://www-nlpir.nist.gov/projects/tv2004/tv2004.html, 2004.

[HAM 03] Hammami M., Tsishkou D., Chen L., “Data-mining based Skin-color Modeling and Applications”, *Third International Workshop on Content-Based Multimedia Indexing*, Ed. SuviSoft Oy Ltd, ISBN 2-7261-1254-4, Rennes, France, Septembre 22-24, pp.157-162, 2003.

[HAR 03a] Harb H., Chen L., “A Query by Example Music Retrieval Algorithm”, *4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS03)*, April 9-11, Queen Mary, University of London, UK, pp. 122-128, 2003.

[HAR 03b] Harb H., Chen L., “Robust Speech/Music Discrimination Using Spectrum’s first Order Statistics and Neural Networks”, *Proc. Of the IEEE Int. Symposium on Signal Processing and its Applications ISSPA2003*, July 1-4, Paris – France, 2003.

[HAR 03c] Harb H., "Classification Sémantique du Signal Sonore en Vue d'une Indexation par le Contenu des Documents Multimédias", Ecole Centrale de Lyon, 11 décembre 2003.

[HAR 06] Harb H., Chen L., “Audio-based visualizing and structuring of videos”, *International Journal on Digital Libraries, Special issue on Multimedia Contents and Management in Digital Libraries*, Vol. 6(1), Springer-Verlag, pp.70-81, 2006.

[HOE 01] Hoey J., “Hierarchical Unsupervised Learning of Facial Expression Categories”, *ICCV Workshop on Detection and Recognition of Events in Video*, 2001.

[HOU 59] Hough P.V.C., “Machine Analysis of Bubble Chamber Pictures”, *Int. Conference on High Energy Accelerators and Instrumentation*, CERN, pp. 554-556, 1959.

[HSU 03] Hsu W., Chang S.-F., “A Statistical Framework for Fusing Mid-Level Perceptual Features in News Story Segmentation”, *IEEE Int. Conference ICME*, 2003.

[HSU 04] Hsu W., Kennedy L., Huang C.-W., Chang S.-F., Lin C.-Y., Iyengar G., “News Video Story Segmentation Using Fusion of Multi-Level Multi-Modal Features in TRECVID 2003”, *IEEE Int. Conference ICASSP*, 2004.

[HUA 99] Huang J., Liu Z., Wang Y., Chen Y., Wong E.K., “Integration of Multi-modal Features for Video Scene Classification Based on HMM”, *IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, 1999.

[IBM] IBM. http://www.research.ibm.com/MediaStar/VideoSystem.html.

[JIA 00] Jiang H., Zhang H., Lin T., “Video Segmentation with hte Support of Audio Segmentation and Classification”, IEEE Int. Conference on Multimedia and Expo (ICME’2000), USA, July 30 – August 2, 2000.

[JUA 85] Juang B.H., Rabiner L.R., “Mixture Autoregressive Hidden Markov Models for Speech Signals”, *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-33, No. 6, pp. 1404-1413, 1985.

[KEN 98] Kender J.R., Yeo B.L., “Video scene segmentation via continuous video coherence”, *Proc. of IEEE CVPR*, pp. 367-373, 1998.

[KIJ 03a] Kijak E., Gravier G., Gros P., Oisel L., Bimbot F., “HMM based structuring of tennis videos using visual and audio clues”, *Proc. IEEE ICME*, 2003.

[KIJ 03b] Kijak E., Gravier G., Oisel L., Gros P., “Audiovisual integration for tennis broadcast structuring”, *International Workshop on CBMI*, pp. 421-428, 2003.

[KOL 04] Kolonias I., Christmas W., Kittler J., “Tracking the Evolution of a Tennis Match Using Hidden Markov Models”, *SSPR/SPR*, pp. 1078-1086, 2004.

[LEK 02] Lekha Chaisorn, Tat-Seng Chua, Chin-Hui Lee, “The Segmentation of News Video into Story Units”, *Proc. IEEE ICME*, 2002.

[LEW 91] Lewis D.D., “Evaluating Text Categorization”, *Proc. of the Speech and Natural Language Workshop*, pp.312-318, 1991.

[LIE 01] Lienhart R., “Reliable transition detection in videos: A survey and practitioner’s guide”, *Int. Journal of Image and Graphics*, Vol.1, No. 3, pp. 469-286, 2001.

[LIE 97] Lienhart R., Pfeiffer S., Effelsberg W., “Video abstracting”, *Communications of the ACM*, 1997.

[LIE 99] Lienhart R., “Comparison of automatic shot boundary detection algorithms”, *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, pp. 290-301, Jan. 1999.

[MAH 00] W.Mahdi, M.Ardebilian, L.Chen, “Automatic Video Scene Segmentation based on Spatial-temporal Clues and Rhythm”, *Networking and Information Systems Journal*, Vol. 3, No. 5, 2000.

[MAH 01] Mahdi W., Ardabilian M., Chen L., “Exterior and interior images classification”, *Patent number 001500933 BF*, Ecole Centrale of Lyon, 2001.

[MAH 02] Mahdi W., “Macro-Segmentation Sémantique des Documents Audiovisuels à l’Aide des Indices Spatio-Temporel”, *Thesis of Doctorale*, Ecole Central of Lyon, 2002.

[MIT 96] Mitchell T.M., “Machine Learning”, McGraw-Hill Series in Computer Science, 1996.

[MIY 00] Miyamori H., Iisaku S-I., “Video Annotation for Content-based Retrieval using Human Behavior Analysis and Domain Knowledge”, *IEEE AFGR*, pp.320-325, 2000.

[MOO 01] Moore D., Essa I., “Recognizing Multitasked Activities Using Stochastic Context-Free Grammars”, *CVPR Workshop on Models vs Exemplars in Computer Vision*, 2001.

[MUR 01] Murphy K.P., Paskin M.A., “Linear Time Inference in Hierarchical HMMs”, Proc. of Neural Information Processing Systems, Vancouver, Canada, 2001.

[MUR 03] Murat Tekalp A., “Sports Video Processing for Event Detection and Summarization”, *3rd Int. Workshop on Content-Based Multimedia Indexing (CBMI 03)*, p.325, 2003.

[NAM 98] Nam J., Enis Cetin A., Tewfik A.H., “Audio-Visual Content-Based Violent Scene Characterization”, *IEEE Int. Conference on Image Processing*, Vol. 1, pp. 353-357, Chicago, USA, 1998.

[NIT 02] Nitta N., Babaguchi N., “Automatic Story Segmentation of Closed-Caption Text for Semantic Content Analysis of Broadcasted Sports Video”, *Multimedia Information Systems*, 2002.

[NGU 05] Nguen N., Venkatesh S., “Discovery of Activity Structures Using the Hierarchical Hidden Markov Model”, *16th British Machine Vision Conference*, Oxford, UK, 2005.

[PAR 04] Parshin v., Chen V., “Video summarization based on user-defined constraints and preferences”, *Proc. of RIAO*, France, pp. 18-24, 2004.

[PAR 05a] V.Parshin, L.Chen, “Event Driven Content Structure Analysis of Tennis Video”, *Proc. of Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Switzerland, April 13-15, 2005.

[PAR 05b] Parshin V., Paradzinets A., Chen L., “Multimodal Data Fusion for Video Scene Segmentation”, *8th Int. Conference on Visual Information Systems (VIS2005)*, Amsterdam, the Netherlands, July 5, 2005.

[PAR 06] Parshin V., Chen L., “Statistical Audio-Visual Data Fusion for Video Scene Segmentation”, *Semantic-Based Visual Information Retrieval*, Idea Group Inc., accepted for publication in March 2006.

[PEL 00] Peleg S., Rousso B., Rav-Acha A., Zomet A., “Mosaicing on Adaptive Manifolds”, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, pp. 1144-1154, vol. 22, No 10, October 2000.

[PHU 05] Phung D.Q., Duong T.V., Venkatesh S., Bui H.H., “Topic Transition Detection Using Hierarchical Hidden Markov and Semi-Markov Models”, Proc. of ACM Multimedia, Singapore, pp. 11-20, 2005.

[RAS 03] Rasheed Z., Shah M., “A Graph Theoretic Approach for Scene Detection in Produced Videos”, *Multimedia Information Retrieval Workshop*, Aug 1, Toronto, Canada, August 2003.

[RAB 89] Rabiner L.R., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”. *Proc. of the IEEE*, 77, no. 2, pp. 257-286, Feb. 1989.

[RIJ 79] Rijsbergen C.J., “Information Retrieval”, Butterworths, 1979.

[ROZ 98] Rozenn Dahyot, Anil Kokaram, Niall Rea and Hugh Denman, “Joint audio visual retrieval for tennis broadcast”, *Proc. IEEE ICASSP*, Hong Kong, April 2003.

[RUI 99] Rui Y., Huang T.S., Mehrotra S., “Constructing table-of-content for videos”, ACM Multimedia Syst., Vol. 7, No. 5, pp. 359-368, September 1999.

[RUS 85] Russel M.J., Moore R.K., “Explicit modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition”, *Proc. IEEE Int. Conference on Acoustic, Speech and Signal Processing*, pp. 5-8, 1985.

[RUS 87] Russel M.J., Cook A.E., “Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition”, *Proc. IEEE Int. Conference on Acoustic, Speech and Signal Processing*, Dallas, pp. 2376-2379, 1987.

[SAR 98] Saraceno C., Leonardi R., “Identification of Story Units in Audio-Visual Sequences by Joint Audio and Video Processing”, IEEE Int. Conference on Image Processing, Chicago, USA, 1998.

[SHA 98] Shai Fine, Yoram Singer, Naftali Tishbi, “The Hierarchical Hidden Markov Model: Analysis and Applications”, *Machine Learning*, Vol. 32, pp. 41-62, 1998.

[SMI 98] Smith M.A., Kanade T., “Video skimming and characterization through the combination of image and language understanding”, *IEEE International Workshop on Content-Based Access of Image and Video Database*, pp. 61-70, 1998.

[SOC] “Soccer Terminology”, in http://www.decatursports.com/soccerterms.htm.

[SUN 00] Sundaram, H., Chang, S.F., “Determining computable scenes in films and their structures using audio-visual memory models”, *Proc. of ACM Multimedia*, USA, pp. 95-104.

[SUN 02] Sundaram H., “Segmentation, Structure Detection and Summarization of Multimedia Sequences”, *PhD thesis work*, Columbia University, 2002.

[THE 01] Theocharous G., Rohanimanesh K., Mahadevan S., “Learning Hierarchical Partially Observed Markov Decision Process Models for Robot Navigation”, IEEE *ICRA*, Seoul, Korea, 2001.

[TZA 01] Tzanetakis G., Essl G., Cook P., “Audio Analysis using the Discrete Wavelet Transform”, *Proc. WSES Int. Conf. Acoustics and Music: Theory and Applications (AMTA 2001)*, Skiathos, Greece, 2001.

[UCH 99] Uchihashi S., Foote J., Girhensohn A., Boreczky J., “Video Manga: Generating Semantically Meaningful Video Summaries”, *Proc. ACM Multimedia 99*, pp. 383-392, 1999.

[VAS 97] Vasconcelos N., Lippman A., “A Bayesian Video Modeling Framework for Shot Segmentation and Content Characterization”, *Proc. of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[VEN 02] Vendrig J., Worring M., “Evaluation measurement for logical story unit segmentation in video sequences”, *IEEE Transactions on Multimedia,* Vol. 4, No. 4, pp. 492-499, December 2002.

[VEN 00] Veneau E., Ronfard R., and Bouthemy P., ”From video shot clustering to sequence segmentation”, *Fifteenth International Conference on Pattern Recognition (ICPR'2000)*, Barcelona, Spain, 2000.

[VIT 67] Viterbi A.J., “Error bounds for convolutional codes and an asymptotically optimal decoding algorithm”, *IEEE Trans. Informat. Theory*, Vol. IT-13, pp. 260-269, Apr. 1967.

[WAL 04] Wallapak Tavanapong, Junyu Zhou, “Shot Clustering Techniques for Story Browsing”, *IEEE Trans. on Multimedia*, 6, No 4, pp. 517-527, 2004.

[XIE 02] Xie L., Chang S-F., Divakaran A., Sun H., “Structure analysis of soccer video with hidden Markov models”, *Proc. IEEE ICASSP*, 2002.

[XIN 03] Xingquan Zhu, Jianping Fan, Ahmed K. Elmagarmid, Xindong Wu, “Hierarchical Video Content Description and Summarization Using Unified Semantic and Visual Similarity”, *http://www.cs.uvm.edu/tr/CS-03-01.shtml*, 2003.

[XUP 01] Xu P., Xie L., Chang S-F., Divakaran A., Vetro A., Sun H., “Algorithms ans system for segmentation and structure analysis in soccer video”, *Proc. IEEE ICME*, 2001.

[YEU 96] Yeung M.M., Yeo B.L., “Time-constrained clustering for segmentation of video into story units”, *International Conference on Pattern Recognition*, vol. C, pp. 375-380, 1996.

[YIH 00] Yihong Gong, Xin Liu, “Generating optimal video summaries”, *Proc. ICME*, 2000.

[ZHE 04] Zhen Ye, Cheng-Chang Lu, “A Wavelet Domain Hierarchical Hidden Markov Model”, *Proc. IEEE ICIP’04*, pp.3491-3494, 2004.

[ZIV 01] Zivkovic Z., F.van der Heijden, Petkovic M., Jonker W., "Image processing and feature extraction for recognizing strokes in tennis game videos", *Proc. of 7th Annual Conference of the Advanced School for Computing and Imaging,* the Netherlands, June 2001, pp.512-516.

List of Figures

List of Tables