4 Video Analysis and Summarization at Structural and Semantic Levels

Hari Sundaram and Shih-Fu Chang

Columbia University

4.1 Introduction

In this chapter, we discuss research issues and promising techniques related to three important aspects of audio-visual content analysis — (a) segmentation, (b) event analysis and (c) summarization. Each component plays an important role in the greater semantic understanding of the audio-visual data.

Segmenting audio-visual data into homogeneous segments in a manner suited for further processing (e.g. visual summaries) is an important first step. The segmentation of video data into manageable chunks is complicated by the presence of complex interactions between audio and video data (e.g. films). Domain dependent syntactical elements (e.g. dialogs) further complicate the segmentation task.

There has been prior work on video scene segmentation using image data alone [24][59]. In [59], the authors derive scene transition graphs to determine scene boundaries. However, cluster thresholds are difficult to set and often have to be manually tuned. In [24], the authors use an infinite, non-causal memory model to segment the video. We refine this idea of memory in our recent work [48], but in a finite, causal setting. Prior work [39][41][46] concerning the problem of audio segmentation dealt with very short-term (100 ms) changes in a few features (e.g. energy, cepstra). This was done to classify the audio data into several predefined classes such as speech, music ambient sounds etc. However, they do not explore the possibility of using the long-term consistency found in the audio data for segmentation.

Event analysis in video is important not only because we are often interested in detecting a specific phenomena, but also because it complements the segmentation procedure. There has been much work done in event analysis [1][3][11][15][21][23] [29][30][46][54][57][58][62], and this chapter we shall only be able to summarize some of the various methods and applications in event analysis. In particular we focus on the use of Hidden Markov Models (HMM's) [57] in soccer and an application for detecting important events in baseball [23].

Summarization is concerned with the problem of generating a drastically reduced representation of video data. This problem is important in many contexts, such as: (a) browsing digital libraries, (b) on demand summaries of the data stored in set-top

boxes (interactive TV), (c) personalized summaries for mobile devices and (d) for news channels (e.g. CNN) that receive a tremendous amount of raw footage.

There has been much research on generating image-based storyboards [42][52][55][59][60][61] and video skims [8][18][25][27][34]. Image based storyboards typically use time constrained clustering on the key-frames of the video shots to determine semantically representative images. However, since they are laid out in a static manner in an html page, they do not convey the underlying dynamism of the audio visual data. In the Informedia skimming project [8], important regions of the video were identified via a TF/IDF analysis of the transcript. They also used face detectors and performed motion analysis for additional cues. The MoCA project [25][34] worked on automatic generation of film trailers. They used heuristics on the trailers, along with a set of rules to detect certain objects (e.g. faces) or events (e.g. explosions). Work at Microsoft Research [18] dealt with informational videos; there, they looked at slide changes, user statistics and pitch activity to detect important segments. Recent work [27] has dealt with the problem of preview generation by generating "interesting" regions based on viewer activity in conjunction with topical phrase detecting. However, in order to generate the preview, some viewers need to have seen the video. Now, we begin by discussing scene analysis.

4.2 Scene analysis

In this section, we begin by giving a computation definition of a *scene* that is based on low-level features alone, than on semantics. Then, we shall discuss two methods for detecting scenes — scene transition graphs [59][60] and a memory model based approach [48][51]. Finally, we shall conclude the section by discussing applications in films, sports video and for news.

THE COMPUTATIONAL SCENE DEFINITION

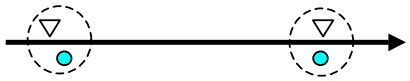
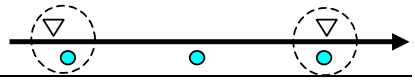
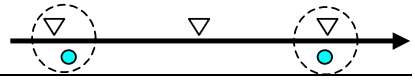
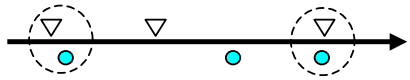
There are constraints on what we see and hear in films, due to *rules* governing camera placement, *continuity* in lighting as well as due to the *psychology* of audition. In this chapter, we develop notions of a video and audio computable scenes by making use of these constraints. We adopt the following definition of audio and video scenes. A video scene is a continuous segment of visual data that shows *long-term*[1] consistency with respect to two properties: (a) chromaticity and (b) lighting conditions, while an audio scene exhibits a long terms consistency with respect to ambient sound. We denote them to be *computable* since these properties can be reliably and automatically determined using low-level features present in the audio-visual data. Note that these are *not* semantic scenes. We believe that these c-scenes are the first step in greater semantic understanding of a scene [51].

The a-scene and the v-scenes represent elementary, homogeneous chunks of information. We define a computable scene (abbreviated as c-scene) in terms of the relationships between a-scene and v-scene boundaries. It is defined to be a segment

[1] Analysis of experimental data (one hour each, from five different films) indicates that for both the audio and the video scene, a minimum of 8 seconds is required to establish context. These scenes are usually in the same location (e.g. in a room, in the marketplace etc.) are and are typically 40~50 seconds long.

between two consecutive, synchronized[2] audio-visual scenes. This results in four cases of interest[3] (Table 4-1). We validated the computable scene definition, which appeared out of intuitive considerations, with actual film data. The data were from three one-hour segments from three English language films[4]. The definition for a scene works very well in many film segments. In most cases, the c-scenes are usually a collection of shots that are filmed in the same location and time and under similar lighting conditions (these are the P and the Ac-V scenes).

Table 4-1: The four types of c-scenes that exist between consecutive, synchronized audio-visual changes. solid circles: indicate audio scene boundaries, triangles indicate video scene boundaries

Type	Abbr.	Figure
Pure, no audio or visual change present.	P	
Audio changes consistent visual.	Ac-V	
Video changes but consistent audio.	A-Vc	
Mixed mode: contains unsynchronized audio and visual scene boundaries.	MM	

The A-Vc (consistent audio, visuals change) scenes seem to occur under two circumstances. In the first case, the camera placement rules are violated. These are montage[5] sequences and are characterized by widely different visuals (differences in location, time of creation as well as lighting conditions) which create a unity of theme by manner in which they have been juxtaposed. MTV videos are good examples of such scenes. The second case consists of a sequence of v-scenes that individually obey the camera placement rules (and hence each have consistent

[2] In films, audio and visual scene changes will *not* exactly occur at the same time, since this is disconcerting to the audience. They make the audio flow "over the cut" by a few seconds [37], [40].
[3] Note that the figures for Ac_v, A-Vc and MM, in Table 4-1 show only one audio/visual change. Clearly, multiple changes are possible. We show only one change for the sake of clarity.
[4] The English films: *Sense and Sensibility, Pulp Fiction, Four Weddings and a Funeral.*
[5] In classic Russian montage, the sequence of shots are constructed from placing shots together that have no immediate similarity in meaning. For example, a shot of a couple may be followed by shots of two parrots kissing each other etc. The meaning is derived from the way the sequence is arranged.

chromaticity and lighting). We refer to the second class as *transient* scenes. Typically, transient scenes can occur when the director wants to show the passage of time e.g. a scene showing a journey, characterized by consistent audio track.

Mixed mode (MM) scenes are far less frequent, and can for example occur, when the director continues an audio theme well into the next v-scene, in order to establish a particular semantic feeling (joy/sadness etc.). Table 4-2 shows the c-scene type break-up from the first hour of the film *Sense and Sensibility*. There were 642 shots detected in the video segment. The statistics from the other films are similar. Clearly, c-scenes provide a high degree of abstraction, that will be extremely useful in generating video summaries. Note that while this paper focuses on computability, there are some implicit semantics in our model: the P and the Ac-V scenes, that represent c-scenes with consistent chromaticity and lighting are almost certainly scenes shot in the same location.

Table 4-2: c-scene breakup from the film sense and sensibility.

C-scene breakup	Count	Fraction
Pure	33	65%
Ac-V	11	21%
A-Vc	5	10%
MM	2	4%
Total	51	100%

METHODS

In this section, we review two different approaches to determining computable scenes — scene transition graphs that only use visual features and a memory model

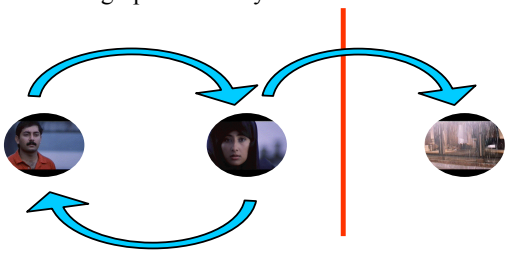


Figure 4-1: A Scene-Transition Graph is segmented into scenes by detecting cut-edges. These edges divide the graphs into disconnected sub-graphs.

based framework that performs multi-modal segmentation.

Scene Transition Graphs

A scene transition graph [59] [61] is a compact representation of a video that is a directed graph. Each node of the graph represents a cluster of similar shots (under a suitable similarity metric). Two nodes i, j are connected via an edge if there exists a

shot in cluster *i* that precedes a shot in cluster *j*. In [61], the authors perform time-constrained clustering on the shots using a time-window T to create the scene transition graph. Given two parameters δ and T, the maximum cluster diameter and the duration of the time window respectively, two shots belong to a cluster if temporally, they are within T sec. of each other and are within δ of each other with respect to the similarity metric [61].

In figure 4-1, we have three nodes, and each node represents a time-constrained cluster. The presence of the cycle between the first two clusters indicates that these two nodes belong to the same scene. A scene transition occurs at a *cut-edge* i.e. when there is forward transition from a sub-graph to another sub-graph with no backwards transition. Note however, that in [59][60][61], the authors attempt to segment the video at a *semantic* level. They do not have a computational model of a scene. An important concern in this work is the setting of the cluster threshold parameter δ and the time-window size T, both of which critically affect the segmentation result. Unfortunately, neither of these parameters can be set with taking the specific character of the data being analyzed.

Memory model

In order to segment data into computable scenes, we use a causal, first-in-first-out (FIFO) model of memory (figure 4-2). This model is derived in part from the idea of coherence [24]. In our model of a listener, two parameters are of interest: (a) memory: this is the net amount of information (T_m) with the viewer and (b) attention span: it is the most recent data (T_{as}) in the memory of the listener (typical values for the parameters are T_m=32 sec. and T_{as}=16 sec.). This data is used by the listener to compare against the contents of the memory in order to decide if a scene change has occurred.

The work in [24] dealt with a non-causal, infinite memory model based on psychophysical principles, for video scene change detection. We use the same psychophysical principles to come up with a causal and finite memory model. Intuitively, causality and a finite memory will more faithfully mimic the human memory-model than an infinite model. We shall use this model for *both* audio and video scene change detection.

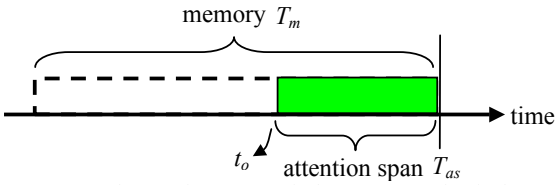


Figure 4-2: The attention span Tas is the most recent data in the memory. The memory (Tm) is the size of the entire buffer.

In order to segment the data into audio scenes, we compute correlations amongst the audio features in the attention-span with the data in the rest of the memory. The video data comprises shot key-frames. The key-frames in the attention span are compared to the rest of the data in the memory to determine a coherence value. This value is derived from a color-histogram dissimilarity. The comparison takes also into

account the relative shot length and the time separation between the two shots. We locate maxima and minima respectively, to determine scene change points.

We introduce a topological framework that examines the local metric relationships between images for structure detection. Since structures (e.g. dialogs) are independent of the duration of the shots, we can detect them independent of the v-scene detection framework. We exploit specific local structure to compute a function

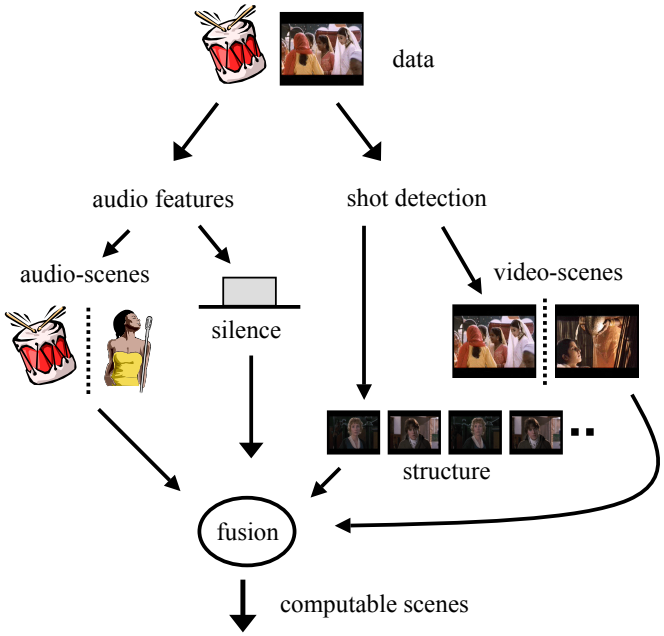


Figure 4-3: computable scene detection overview.

that we term the periodic analysis transform. We test for significant dialogs using the standard Students t-test. The silence is detected via a threshold on the average energy; we also impose minimum duration constraints on the detector.

A key feature of our work is the idea of imposing semantic constraints on our computable scene model. This involves fusing (see figure 4-3) results from silence and structure detection algorithms. The computational model cannot disambiguate between two cases involving two long and widely differing shots: (a) in a dialog sequence and (b) adjoining video scenes. However, human beings recognize the structure in the sequence and thus group the dialog shots together. Silence is useful in two contexts: (a) detecting the start of conversation by determining significant pauses [15] and (b) in English films, the transitions between computable scenes may involve silence. Our experiments [51] show that the c-scene change detector and the structure detection algorithm work well.

4.3 Event analysis

In this section we briefly review work done on event analysis. We begin by defining an event, then we discuss three methods for event detection — Hidden Markov Models, graphical models and Bayesian inference. Then, we discuss two specific applications in baseball detection and echocardiogram videos.

In this work, we define an event to be a change of state or property of an entity. MPEG-7 has a rich description schemes (DS's) to describe entities, entity attributes and relationships between entities [28]. While these descriptions may just be textual, the framework is powerful since it supports reasoning and inference. The Event DS describes an event, which is a semantic activity that takes place at a particular time or in a particular location. The Event DS can describe either a perceivable or an abstract event in a narrative world. A perceivable event is a dynamic relation involving one or more objects taking place in time and space in a narrative world (e.g., "Alex shaking hands with Ana"). An abstract event results from abstraction of a perceivable event (e.g., "A man shaking hands with a woman"). The Event DS includes elements that describe the composition of the event from sub-events, in addition to the location and time of the event.

METHODS

Hidden Markov Models

In this section we shall discuss how Hidden Markov Models (HMM's) a widely used statistical technique [36] can be used for event detection in soccer videos [57]. The problem is useful in automatic content filtering for soccer fans and professionals, and it is more interesting in the broader background of video structure analysis and content understanding. By structure, we are primarily concerned with the recurrent temporal sequence of high-level game states, namely *play* and *break*. The game is *in play* when the ball is in the field and the game is going on; *break,* or *out of play*, is the compliment set, i.e. whenever "the ball has completely crossed the goal line or touch line, whether on the ground or in the air" or "the game has been halted by the referee".

The states *play* (*P*) and *break* (*B*) consists of different sub-structures such as the switching of shots and the variation of motion. This is analogous to isolated word recognition [36] where models for each word are built and evaluated with the data likelihood. But as these domain-specific classes *P/B* in soccer are very diverse in themselves (typically ranging from 6 seconds up to 2 minutes in length), we use a set of models for each class to capture the structure variations.

We take a fixed-length sliding window (width 3 seconds, sliding by 1 second) and classify the feature vector into either one of the *P/B* classes. The feature stream is first smoothed by a temporal low-pass filter, normalized with regard to its mean and variance of the entire clip, then the segment of size 2xN in each time slice (2 is feature dimension, N is window length) is fed into the HMM-dynamic programming modules for classification. In our system, 6 HMM topologies are trained for *play* and for *break*, respectively. These include 1/2/3-state fully connected models, 2/3 state left-right models and a 2-state fully connected model with an entering and an exiting

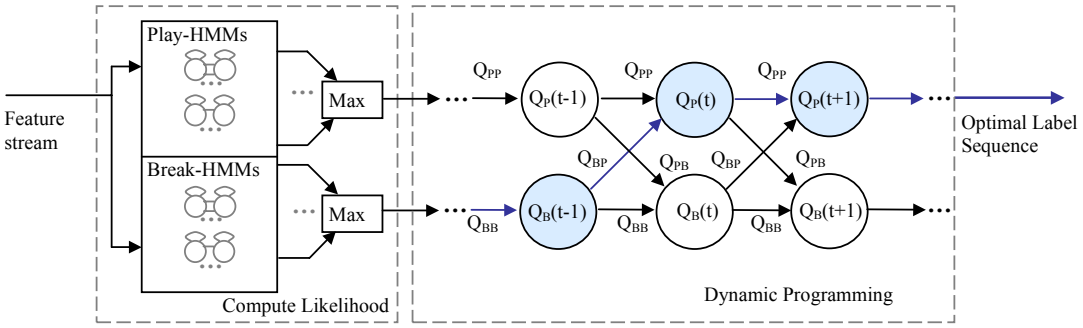


Figure 4-4. HMM-dynamic programming diagram. Only 2 out of 6 HMM topologies are shown; the *Q*s are HMM model likelihood or transition likelihood

state. The observations are modeled as mixture of Gaussians, and we have 2 mixtures per feature dimension per state in the experiments (figure 4-4).

The HMM model parameters are trained using the EM algorithm. Training data are manually chopped into homogeneous *play/break* chunks; EM for the *play*-models are conducted over every complete *play* chunks, and vice versa for *break*-models. HMM training is not conducted over 3-second windows because we hope that the HMM structures can take longer time correlation into account, and thus "tolerate" some less frequent events in a state, such as short *close-ups* within a *play*. Experiments show that the overall accuracy will be consistently 2~3% lower if models are trained on short segments, and the video tends to be severely over-segmented as some of the short close-ups and cutaways during a *play* will be misclassified as *break*.

Extensive statistical analyses show that classification accuracy is about 83.5% over diverse data sets, and most of the boundaries are detected within a 3-second ambiguity window. It is encouraging that high-level domain-dependent video structures can be computed with high accuracy using compressed-domain features and generic statistical tools. We believe that the performance is because the features used are tuned to the domain syntax, as well as due to the power of the statistical tools in capturing the temporal dynamics of the video.

Learning object/scene detectors

In Visual Apprentice [23], we addressed a different research issue – how to efficiently develop classifiers or detectors for detecting video objects or scenes that correspond to specific events in video (such as the pitching scene in baseball). A user defines visual object/scene models via a multiple-level definition hierarchy: a scene consists of objects, which consist of object-parts, which consist of perceptual-areas, which consist of regions. The user trains the system by providing example images/videos and labeling components according to the hierarchy she defines (e.g., image of two people shaking hands contains two faces and a handshake). As the user trains the system, visual features (e.g., color, texture, motion, etc.) are extracted from each example provided, for each node of the hierarchy (defined by the user). Various machine learning algorithms are then applied to the training data, at each node, to

learn classifiers. The best classifiers and features are then automatically selected for each node (using cross-validation on the training data). The process yields a visual object/scene detector (e.g., for a handshake), which consists of an hierarchy of classifiers as it was defined by the user. The visual detector classifies new images/videos by first automatically segmenting them, and applying the classifiers according to the hierarchy: regions are classified first, followed by the classification of perceptual-areas, object-parts, and objects. In [19] the concept of recurrent visual semantics is discussed, specifically on how it can be used to identify domains in which learning techniques such as the one presented in that work can be applied.

The technique proposed is specifically used to detect batting events in baseball video. This is done by building a hierarchy that represents the frontal view of the typical batting scene. The scene is represented by a hierarchy with the following elements: the scene contains a pitcher, a batter, and a field. The field contains two nodes, one for the sand and one for the grass. Each of the leaf nodes (pitcher, batter, sand-field, and grass-field) is represented by regions obtained from automatic segmentation. The approach in [23] can be used in video event detection by constructing Visual Object/Scene Detectors for specific domains. One possibility would be to construct several detectors, for example, and combine them to define higher-level semantic events. In baseball, a model for a batting scene from the VA, for example, could be used with a model for the typical camera motion that follows a homerun (extending the VA or using another approach). The detection of the homerun event, then, would occur when the batting scene and corresponding motion are found. The VA was also use to detect handshake "events" in news images.

4.4 Video Summarization

In this section, we review work done in summarizing video data. A video summary is a drastically reduced temporal representation that attempts to capture the semantics of the underlying audio-visual data. There are two forms of summaries that we shall discuss here — image storyboards and visual skims.

IMAGE STORYBOARDS

Image based storyboards offer a key-frame based non-linear navigation of the video data [42][52][59]. Briefly, all image based storyboard algorithms broadly share the following approach: (a) determine an appropriate feature space to represent the images (b) a clustering technique to cluster the image sequence in the feature space and (c) computing a measure of importance to determine the appropriate key-frame to represent the cluster.

Scene transition graphs

The scene transition graph (STG) offers a compact representation of the video. We can browse and navigate the video in a hierarchical, non-linear fashion. A STG shares similar characteristics to other image based storyboards in that it clusters frames in a feature space. However, by analyzing the shot transitions amongst the clusters, it provides for some of the temporal dynamics to be visualized in the storyboard. The analysis of the label transitions also allows the STG to detect elements of visual syntax, such as the dialog.

Enhancing image storyboards

Conventional image based storyboards do not capture the dynamism of the underlying audio-visual data. Hence there has been some effort to improve the interactivity of these schemes [53]. There, the image summary was enhanced with text (either from manual transcripts or OCR) and presented in a *manga*[6] like fashion. We outline four possible ways of enhancing current image summarization schemes.

1. **Text balloons:** If we have text aligned transcripts, then it may be possible to extract the important sentences corresponding to the cluster and then displaying them when the user moves the mouse over the relevant image.
2. **Audio segments:** If we perform an acoustic analysis of prosody [15], then we can identify important boundaries in the discourse (both for spontaneous and structured speech). Then, we could associate each key-frame with this audio segment. In the usage scenario, the user would click on the key-frame to hear the corresponding audio segment. In [50], we have automatically detected discourse boundaries for use in video skims.
3. **Animations:** At present image based storyboards are static i.e. the image representing each cluster does not change over time. An interesting variation would be to represent each image by an animated GIF, which cycles through other images in the cluster when the user moves the cursor over the storyboard key-frame. Another attempt at infusing dynamism is the *dynamic STG* [60] in which the shots comprising each cluster are rendered slowly over time.
4. **Structure and syntactical highlights:** Many domains possess very specific rules of syntax and characteristic structural elements that are meaningful in that domain. Examples of structure in films include the dialog and the regular anchor [51]. Specialized domains such as baseball, echocardiogram videos have a very specific syntactical description. These domains would greatly benefit from higher-order domain grouping rules that arranges the key-frames of the cluster in a manner highlighting the rules of syntax and the domain specific structures.

In the next section, we shall discuss the generation of audio-visual skims.

VISUAL SKIMS

A video skim is a short audio-visual clip that summarizes the original video data. The problem is important because unlike the static, image-based video summaries [53], video skims preserve the dynamism of the original audio-visual data. Applications of audio-visual skims include: (a) on demand summaries of the data stored in set-top boxes (interactive TV) (b) personalized summaries for mobile devices and (c) for news channels (e.g. CNN) that receive a tremendous amount of raw footage.

There has been prior research on generating video skims. In the Informedia skimming project [8], important regions of the video were identified via a TF/IDF

[6] Manga is the Japanese for a comic book.

analysis of the transcript. They also used face detectors and performed motion analysis for additional cues. The MoCA project [34] worked on automatic generation of film trailers. They used heuristics on the trailers, along with a set of rules to detect certain objects (e.g. faces) or events (e.g. explosions). Work at Microsoft Research [18] dealt with informational videos; there, they looked at slide changes, user statistics and pitch activity to detect important segments. Recent work [27] has dealt with the problem of preview generation by generating "interesting" regions based on viewer activity in conjunction with topical phrase detecting. However, in order to generate the preview, some viewers need to have seen the video.

The goal of this work is the automatic generation of audio-visual skims for *passive*[7] tasks, that summarize the video. We make the following assumptions:

1. We do *not* know the semantics of the original.
2. The data is not a raw stream (e.g. home videos), but is the result of an editing process (e.g. films, news).

Since we work on passive tasks, the information needs of the user are a priori unknown. A decision to detect certain set of predefined events will induce a bias in the skim, thereby conflicting with the assumption that the user needs are unknown.

We first begin by discussing syntax preserving visual skims, then we discuss techniques for auditory analysis so as to integrate audio in the skim, and then we conclude by presenting our optimization framework for skim generation.

Syntax preserving

There are four important challenges to be overcome in skim generation:

1. What is the relationship between the visual complexity of a shot and its comprehension time?
2. How does the syntactical structure of the video data affect its meaning?
3. How to select audio segments optimally? And how to ensure that the resulting skim is coherent?
4. Can we solve the skim generation problem in a general constrained utility maximization framework, so as to be able to easily add additional constraints easily?

First, we discuss the visual analysis that comprises two parts — visual complexity and analysis of visual syntax.

[7] A task is defined to be active when the user requires certain information to be present in the final summary (e.g. "find me all videos that contain Colin Powell."). In a passive task, the user does not have anything specific in mind, and is more interested in consuming the information e.g. set-top box previews.

Visual complexity

The intuition for relating image complexity and comprehension time comes from two sources: (a) empirical observations from film theory and (b) experimental evidence from psychology. Directors have long made use of the fact that the audience takes less time to comprehend close ups than long shots (as long shots usually have a lot of detail) [44] to modulate the duration of each shot. Recent results in experimental psychology [13] indicate the existence of an empirical law: the subjective difficulty in learning a concept is directly proportional to the logical incompressibility of the Boolean concept.

We define the visual complexity of an shot to be its Kolmogorov complexity [10]. In

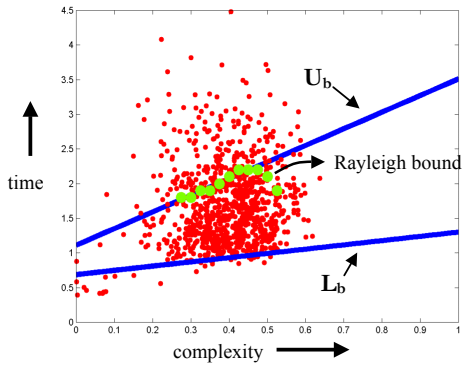


Figure 4-8: Avg. comprehension time (sec.) vs. normalized complexity (x-axis) showing comprehension (upper/lower) bounds. It also shows the Rayleigh (95th percentile) bounds.

[49], we showed that length of the Lempel-Ziv[8] codeword asymptotically converges to the Kolgomorov complexity of the shot. The complexity is estimated using a single key-frame. We conducted our experiments [49] over a corpus of over 3600 shots from six films. There, a shot was chosen at random (with replacement) and then its key-frame presented to the subject (the first author). Then, we measured the time to answer the following four questions (in randomized order), in an interactive session: (a) who (b) when (c) what and (d) where. The subject was expected to answer the questions in minimum time *and* get all four answers right.

From an analysis of the complexity–average comprehension time density (see figure 4-8), we obtain lower and upper bounds on the density. The equations for the lines are as follows:

$$U_b(c) = 2.40c + 1.11,$$
$$L_b(c) = 0.61c + 0.68, \qquad \text{<1>}$$

[8] Lempel-Ziv encoding is a form of universal data coding that doesn't depend on the probability distribution of the source [10].

where c is the normalized complexity and U_b and L_b are the upper and lower bounds respectively, in sec. The upper bound [49] implies that the for 95% of the shots, the average time for comprehension lies below this line; the second line just lower bounds the entire density. The lines were estimated for $c \in [0.25, 0.55]$ (since most of the data lies in this range) and then extrapolated. Hence, given a shot of duration t_o and normalized complexity c_S, we can condense it to at most $U_b(c_S)$ sec by removing the last t_o - $U_b(c_S)$ sec.

Analysis of film syntax

The phrase film syntax refers to the specific arrangement of shots so as to bring out their mutual relationship [49]. In practice, this takes on many forms (chapter 2, [44]) : (a) minimum number of shots in a sequence (b) varying the shot duration, to direct attention (c) changing the scale of the shot (there are "golden ratios" concerning the distribution of scale) (d) the specific ordering of the shots (this influences the meaning). These syntactical rules lack a formal basis, and have been arrived at by trial and error by film-makers. Hence, even though shots in a scene only show a small portion of the entire setting at any one time, the syntax allows the viewers to understand that these shots belong to the same scene. Visual syntax is important because film-makers do not think in terms of individual shots, but in phrases of shots. A shot can have a multitude of meanings, that gets clarified by its relationship to other shots. In [49], we used two elements of cinematic syntax for scene-level compression, and showed how one can exploit film-making rules in order to come up with a shot-dropping strategy.

Natural language integration

Audio skim generation aims at dramatic time reduction (up to 90%) while preserving perceptual coherence. There are some clear drawbacks to simple approaches to determining useful segments in the audio stream. Let us assume that we wish to compress an audio track that is 100 sec. long, by 90%. Then: (a) downsampling the audio by 90% will leave the audio to be severely degraded since the pitch of the speech segments will increase dramatically. (b) PR-SOLA [18] is a non-linear time compression technique that eliminates long pauses, and attempts to preserve the original pitch in the output. User studies indicate that users do not prefer to have the speech sped up beyond 1.6x (i.e. ~40% compression). (c) selecting only those segments that are synchronous with the pre-selected video shots makes the audio stream is choppy and difficult to comprehend [8].

Audio segment classification

We build a tree-structured classifier to classify each frame (100ms) into four generic classes: silence, clean speech, noisy speech and music / environmental sounds. We use 16 features in our approach. Silence frames are first separated from the rest of the audio stream using an adaptive threshold on the energy. Two SVM classifiers are then used in cascade: the remaining frames are separated into speech vs. non-speech (music or environmental sounds); and the speech class is further classified as clean and noisy speech. We then apply a modified Viterbi decoding algorithm [50] to smooth the sequence of frame labels. The decoder makes use of the class transition

probabilities, classifier error likelihood and a duration utility (a function of the prior duration distribution of each class) to find the maximum likelihood class path.

Detecting significant phrases

In this work we focus on detecting segment beginnings (SBEG's) in speech [20][19]. These are important as they serve as the introduction of new topic in the discourse. There has been much work in the computational linguistics community [19][20][43] to determine the acoustic correlates of the prosody in speech. Typically, SBEG's have a preceding pause that is significantly longer than for other phrases, higher initial pitch values (mean, variance), and smaller pauses that end the phrase than for other phrases [19][20]. In our algorithm, we extract the following features per phrase: pitch and energy values (min, max, mean, variance) for the (initial, last and complete) portions of the phrase, pause durations preceding and following the phrase. Once we've extracted the acoustic features per candidate phrase, the phrase is then classified using a SVM classifier [50].

Highlight generation

We use a constrained utility framework to create audio-visual skims [49]. There are three key components to our framework: (a) a utility model for video shots and audio segments (b) constraints stemming from audio-visual synchronization considerations and from minimum duration and (c) a constraint relaxation strategy to ensure a feasible solution. We shall only summarize our work here, the details can be found in [50].

Utility functions

In order to determine the skim duration, we need to measure the comprehensibility of a video shot and a audio segment as a function of its duration. The shot utility function, models the comprehensibility of a shot as a continuous function of its duration and its visual complexity. Note that the results dealing with visual complexity do not tell us how the comprehensibility of a shot *changes* when we decrease its duration. Hence the need for a shot utility function. While we do not have any experimental results indicating a similar complexity-time relationship for audio, it seems fairly reasonable to conjecture its existence. Hence, the form of our audio utility function will be similar to the utility function for video shots [50]. We model the utility of a video shot (audio segment) independently of other shots (segments).

Constraints

There are three principal constraints in our algorithm: (a) audio-visual synchronization requirements (b) minimum and maximum duration bounds on the video shots and the audio segments and (c) the visual syntactical constraints.

Audio-visual synchronization is achieved using the idea of tied multimedia segments. A multimedia segment is said to be fully *tied* if the corresponding audio and video segments begin and end synchronously, and in addition are *uncompressed.* We only associate those speech segments that contain significant phrases with tied multimedia segments

We focus on the generation of passive information centric summaries that have maximum coherence. Since we deem the speech segments to contain the maximum information, we shall seek to achieve this by biasing the audio utility functions in favor of the clean speech class. In order to ensure that the skim appears coherent, we do two things: (a) ensure that the principles of visual syntax are not violated and (b) have maximal number of tie constraints. These constraints ensure synchrony between the audio and the video segments. The target skim duration is met by successively relaxing the constraints. Relaxing the synchronization constraints has two effects: (a) the corresponding audio and video segments are no longer synchronized (b) they can be compressed and if necessary dropped. The details of the search strategy and the mathematical framework can be found in [50].

DOES THERE EXIST AN "OPTIMAL" SUMMARY?

In this section, we attempt to answer a fairly intuitive question — are audio-visual skims optimal in some sense? Clearly, they preserve the dynamism of the original video, while attempting to preserve the semantics. The answer lies in looking at the relationship of the summary to the device, and the user's information needs.

The device on which the summary is to be rendered affects the skim in at least two ways: the nature of the user interface and the device constraints. The user interface can be complex (e.g. the PC), medium (e.g. a palm pilot) and simple (e.g. a cell phone). The user interface affects the resolution of the visual skim as well as the size of the thumbnails of the image storyboard. This is an important consideration, because on very small screens (e.g. cell phone) it would be very difficult for the user to comprehend the tiny thumbnails shown on the screen. Note that the user interface also influences the kinds of tasks that the user has in mind (e.g. it is difficult to input a query on a cell phone).

The computational resources available on the device — cpu speed, memory, bandwidth, availability of an audio rendering device, all effect the form of the summary. For example, a palm-pilot or a cell phone may not have the computational resources to render a video skim. The specific resources present will affect the resolution of the skim, and the decision to include video (as opposed to still images) in the skim. Hence, only when we know the nature of the user interface and the device resource capabilities can we come to a conclusion on the form of the summary.

4.5 Summary

In this chapter we have discussed three important aspects of video analysis — (a) scene analysis, (b) analysis of events and (c) schemes to summarize video.

In scene analysis we reviewed work done with scene transition graphs and the memory model based segmentation framework. We described a computational scene model for films. We showed the existence of four different types of computable scenes, that arise due to different synchronizations between audio and video scene boundaries. The computational framework for audio and video scenes was derived from camera placement rules in film-making and from experimental observations on the psychology of audition. We believe that the computable scene formulation is the first step towards deciphering the semantics of a scene.

Scene transition graphs are constructed using time-constrained clustering of video frames along with cluster label transition analysis. This results in a directed graph, that is then analyzed via analysis of cut edges for scene changes. We then showed how a causal, finite memory model formed the basis of our audio and video scene segmentation algorithm. In order to determine audio scene segments we determine correlations of the feature data in the attention span, with the rest of the memory. An important aspect of this work is the incorporation of high-level semantic constraints for merging information from different modalities (audio, video, silence and structure) to ensure that the resulting segmentation is consistent with human perception.

In event analysis, we first defined an event to be a change in state or property of an entity. Then we briefly discussed MPEG-7 event description schemes. We discussed in some detail the use of hidden markov models for detecting states in soccer. We also gave an overview of an interactive framework, the Visual Apprentice, for learning video object/scene detectors and their applications for event detection in sports.

We discussed two summarization schemes — image based storyboards and video skims. Image based storyboards are typically constructed by clustering video shots in a feature space with an appropriate metric. The video shot key-frame closest to the cluster centroid is picked as the representative key-frame.

In this paper, we've presented a novel framework for condensing computable scenes. The solution has three parts: (a) analysis of visual complexity and film syntax, (b) robust audio segmentation and significant phrase detection via SVM's and (c) determining the duration of the video and audio segments via an constrained utility maximization. We defined a measure for visual complexity and then showed how we can map visual complexity of a shot to its comprehension time. After noting that the syntax of the shots influences the semantics of a scene, we devised algorithms based on simple rules governing the length of the progressive phrase and the dialog. We devised a robust audio segmentation algorithm using SVM classifiers in a tree structure, and imposed duration constraints on the segments using the a modified Viterbi algorithm. We also showed how we could analyze the prosody and detect significant phrases in the speech segments.

We have focused on generating information centric skims with maximum coherence. First, we developed utility functions for both audio and video segments, and we minimized an objective function that was based on the sequence utility. We introduced the idea of tied multimedia segments that imposes synchronization constraints on the skim. Additionally, the objective function is subject to video and audio penalty functions and minimum duration requirements on the audio and video segments.

4.6 Open Issues

We now discuss some of the interesting research issues connected to the ideas discussed here.

For video scene segmentation, there are several clear improvements possible to the work on memory model based scene analysis. The computational model for the detecting the video scene boundaries is limited, and needs to tightened in view of the model breakdowns discussed in [51]. One possible improvement is to do motion analysis on the video and prevent video scene breaks under smooth camera motion. Since shots misses can cause errors, we are also looking into using entropy-based irregular sampling of the video data in addition to the key-frames extracted from our shot-segmentation algorithm.

At a more conceptual level, regardless of the domain of analysis, we believe that general-purpose segmentation algorithms should be modified keeping the following aspects in mind: (a) How does the specific domain affect the data that we are trying to analyze? Are there explicit (or implicit) constraints? (b) what are the specific syntactical structures in the domain? How do the viewers in that domain group these elements and what meanings do they assign to these structures. (c) What is the sensitivity of the final task to segmentation? In many cases, it may be likely that the final task (e.g. classification) is not very sensitive to the error and performance improvements can be found by fine tuning other aspects of the overall algorithm.

In summarization algorithms, there are several interesting avenues of research. Work on enhancing image based storyboards with animations, sound and text captions is an exciting area of research. The work on audio visual skims can be improved in many ways: (a) the creation of summaries for active tasks, in a constrained environment (b) computing time complexity curves with both audio and video (c) how to construct skims for raw video streams by modifying the skim generation mechanism discussed here? (d) given a domain, a systematic methodology for the assignment of utilities to the various entities of interest in that domain, and how these utilities are modified via their inter-relationships (e) summarization by changing the modality — for example, it may be possible to summarize a video in shorter period of time, by overlaying text captions that summarize the audio visual data (for example, the text captions could come from the transcript).

4.7 Acknowledgements

We would like to sincerely appreciate the help and valuable input provided by Ana Benitez, Shahram Ebadollahi, Alejandro Jaimes, and Lexing Xie.

References

[1] B. Arons *Pitch-Based Emphasis Detection For Segmenting Speech Recordings,* Proc. ICSLP 1994, Sep. 1994, vol. 4, pp. 1931-1934, Yokohama, Japan, 1994.

[2] B. Adams et. al. *Automated Film Rhythm Extraction for Scene Analysis,* Proc. ICME 2001, Aug. 2001, Japan.

[3] A.B. Benitez, S.F. Chang, J.R. Smith *IMKA: A Multimedia Organization System Combining Perceptual and Semantic Knowledge,* Proc. ACM MM 2001, Nov. 2001, Ottawa Canada.

[4] A.S. Bregman Auditory Scene Analysis: The Perceptual Organization of Sound, MIT Press, 1990.

[5] B. Burke and F. Shook, "Sports photography and reporting", Chapter 12, in *Television field production and reporting*, 2nd Ed, Longman Publisher USA, 1996

[6] M. Burrows, D.J. Wheeler *A Block-sorting Lossless Data Compression Algorithm,* Digital Systems Research Center Research Report #124, 1994.

[7] C.-C. Chang, C.-J.Lin, *LIBSVM: a library for support vector machines,* http://www.csie.ntu.edu.tw/~cjlin/libsvm

[8] M.G. Christel et. al *Evolving Video Skims into Useful Multimedia Abstractions,* ACM CHI '98, pp. 171-78, Los Angeles, CA, Apr. 1998.

[9] N. Christianini, J. Shawe-Taylor *Support Vector Machines and other kernel-based learning methods,* 2000, Cambridge University Press, New York.

[10] T.M. Cover, J.A. Thomas *Elements of Information Theory,* 1991, John Wiley and Sons.

[11] S. Ebadollahi, S.F. Chang, H. Wu, *Echocardiogram Videos: Summarization, Temporal Segmentation And Browsing,* to appear in ICIP 2002, Sep. 2002, Rochester NY.

[12] D.P.W. Ellis *Prediction-Driven Computational Auditory Scene Analysis,* Ph.D. thesis, Dept. of EECS, MIT, 1996.

[13] J. Feldman *Minimization of Boolean complexity in human concept learning,* Nature, pp. 630-633, vol. 407, Oct. 2000.

[14] Bob Foss *Filmmaking: Narrative and Structural techniques* Silman James Press LA, 1992.

[15] Y. Gong; L.T. Sin; C. Chuan; H. Zhang; and M. Sakauchi, *Automatic parsing of TV soccer programs*, *Proc. ICMCS'95*, Washington D.C, May, 1995

[16] B. Grosz J. Hirshberg *Some Intonational Characteristics of Discourse Structure,* Proc. Int. Conf. on Spoken Lang. Processing, pp. 429-432, 1992.

[17] A. Hanjalic, R.L. Lagendijk, J. Biemond *Automated high-level movie segmentation for advanced video-retrieval systems,* IEEE Trans. on CSVT, Vol. 9 No. 4, pp. 580-88, Jun. 1999.

[18] L. He et. al. *Auto-Summarization of Audio-Video Presentations,* ACM MM '99, Orlando FL, Nov. 1999.

[19] J. Hirschberg, B. Groz *Some Intonational Characteristics of Discourse Structure,* Proc. ICSLP 1992.

[20] J. Hirschberg D. Litman *Empirical Studies on the Disambiguation of Cue Phrases,* Computational Linguistics, 1992.

[21] J. Huang; Z. Liu; Y. Wang, *Joint video scene segmentation and classification based on hidden Markov model*, *Proc. ICME 2000,* P 1551 -1554 vol.3, New York, NY, July 30-Aug3, 2000

[22] J. Huang; Z. Liu; Y. Wang, *Integration of Audio and Visual Information for Content-Based Video Segmentation,* Proc. ICIP 98. pp. 526-30, Chicago IL. Oct. 1998.

[23] A. Jaimes and S.F. Chang, *Concepts and Techniques for Indexing Visual Semantics,* book chapter in Image Databases, Search and Retrieval of Digital Imagery, edited by V. Castelli and L. Bergman. Wiley & Sons, New York, 2002

[24] J.R. Kender B.L. Yeo, *Video Scene Segmentation Via Continuous Video Coherence,* CVPR '98, Santa Barbara CA, Jun. 1998.

[25] R. Lienhart et. al. *Automatic Movie Abstracting,* Technical Report TR-97-003, Praktische Informatik IV, University of Mannheim, Jul. 1997.

[26] L. Lu et. al. *A robust audio classification and segmentation method*, ACM Multimedia 2001, pp. 203-211, Ottawa, Canada, Oct. 2001.

[27] T.S-Mahmood, D. Ponceleon, *Learning video browsing behavior and its application in the generation of video previews,* Proc. ACM Multimedia 2001, pp. 119 - 128, Ottawa, Canada, Oct. 2001.

[28] MPEG MDS Group, *Text of ISO/IEC 15938-5 FDIS Information Technology — Multimedia Content Decsription Interface — Part 5 Multimedia Description Schemes,* ISO/IEC JTC1/SC29/WG11 MPEG01/N4242, Sydney, July 2001.

[29] J. Nam, A.H. Tewfik *Combined audio and visual streams analysis for video sequence segmentation,* Proc. ICASSP 97, pp. 2665 –2668, Munich, Germany, Apr. 1997.

[30] M. Naphade et. al. *Probabilistic Multimedia Objects Multijects: A novel Approach to Indexing and Retrieval in Multimedia Systems*, Proc. I.E.E.E. International Conference on Image Processing, Volume 3, pages 536-540, Chicago, IL, Oct 1998.

[31] M. Naphade et. al *A Factor Graph Framework for Semantic Indexing and Retrieval in Video, Content-Based Access of Image and Video Library* 2000 June 12, 2000 held in conjunction with the IEEE Computer Vision and Pattern Recognition 2000.

[32] R. Patterson et. al. *Complex Sounds and Auditory Images, in Auditory Physiology and Perception* eds. Y Cazals et. al. pp. 429-46, Oxford, 1992.

[33] S. Paek and S.-F. Chang, *A Knowledge Engineering Approach for Image Classification Based on Probabilistic Reasoning Systems* , IEEE International Conference on Multimedia and Expo. (ICME-2000), New York City, NY, USA, Jul 30-Aug 2, 2000.

[34] S. Pfeiffer et. al. *Abstracting Digital Movies Automatically,* J. of Visual Communication and Image Representation, pp. 345-53, vol. 7, No. 4, Dec. 1996.

[35] W.H. Press et. al *Numerical recipes in C,* 2nd ed. Cambridge University Press, 1992.

[36] L. R. Rabiner B.H. Huang *Fundamentals of Speech Recognition,* Prentice-Hall 1993.

[37] K. Reisz, G. Millar, *The Technique of Film Editing,* 2nd ed. 1968, Focal Press.

[38] C. Saraceno, R. Leonardi *Identification of story units in audio-visual sequences by joint audio and video processing,* Proc. ICIP 98. pp. 363-67, Chicago IL. Oct. 1998.

[39] E. Scheirer M.Slaney *Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator* Proc. ICASSP '97, Munich, Germany Apr. 1997.

[40] S. Pfeiffer et. al. *Automatic Audio Content Analysis,* Proc. ACM Multimedia '96, pp. 21-30. Boston, MA, Nov. 1996,

[41] J.Saunders *Real Time Discrimination of Broadcast Speech/Music,* Proc. ICASSP '96, pp. 993-6, Atlanata GA May 1996.

[42] B. Shahraray, D.C. Gibbon *Automated Authoring of Hypermedia Documents of Video Programs,* in Proc. ACM MM 95, pp. 401-409, 1995.

[43] D. O'Shaughnessy *Recognition of Hesitations in Spontaneous Speech,* Proc. ICASSP, 1992.

[44] S. Sharff *The Elements of Cinema: Towards a Theory of Cinesthetic Impact,* 1982, Columbia University Press.

[45] L.J. Stifelman *The Audio Notebook: Pen and Paper Interaction with Structured Speech,* PhD Thesis, Program in Media Arts and Sciences, School of Architecture and Planning, MIT, Sep. 1997.

[46] S. Subramaniam et. al. *Towards Robust Features for Classifying Audio in the CueVideo System,* Proc. ACM Multimedia '99, pp. 393-400, Orlando FL, Nov. 1999.

[47] H. Sundaram S.F. Chang *Audio Scene Segmentation Using Multiple Features, Models And Time Scales,* ICASSP 2000, International Conference in Acoustics, Speech and Signal Processing, Istanbul Turkey, Jun. 2000.

[48] H. Sundaram, S.F. Chang *Determining Computable Scenes in Films and their Structures using Audio-Visual Memory Models,* Proc. Of ACM Multimedia 2000, pp. 95-104, Nov. 2000, Los Angeles, CA.

[49] H. Sundaram, S.F. Chang, *Condensing Computable Scenes using Visual Complexity and Film Syntax Analysis,* IEEE Proc. ICME 2001, Tokyo, Japan, Aug 22-25, 2001.

[50] H. Sundaram L. Xie Shih-Fu Chang *A framework work audio-visual skim generation.* Tech. Rep. # 2002-14, Columbia University, April 2002.

[51] H. Sundaram, S.F. Chang *Computable Scenes and structures in Films,* IEEE Trans. on Multimedia, Vol. 4, No. 2, June 2002.

[52] Y. Taniguchi et. al. *PanoramiaExcerpts: Extracting and Packing Panoramas for Video Browsing*, in Proc. ACM MM 97, pp. 427-436, Seattle WA, Nov. 1997.

[53] R. Tansley. *The Multimedia Thesaurus: Adding A Semantic Layer to Multimedia Information.* Ph.D. Thesis, Computer Science, University of Southampton, Southampton UK, August 2000.

[54] V. Tovinkere , R. J. Qian, *Detecting Semantic Events in Soccer Games: Towards A Complete Solution*, *Proc. ICME 2001*, Tokyo, Japan, Aug 22-25, 2001

[55] S. Uchihashi et. al. *Video Manga: Generating Semantically Meaningful Video Summaries* Proc. ACM Multimedia '99, pp. 383-92, Orlando FL, Nov. 1999.

[56] T. Verma *A Perceptually Based Audio Signal Model with application to Scalable Audio Compression,* PhD thesis, Dept. Of Electrical Eng. Stanford University, Oct. 1999.

[57] L. Xie et. al *Structure Analysis Of Soccer Video With Hidden Markov Models*, to appear in ICASSP 2002, Orlando, Fl, May 2002.

[58] P. Xu, L. Xie, S.F. Chang, A. Divakaran, A, Vetro, and H. Sun, *Algorithms and system for segmentation and structure analysis in soccer video*, *Proc. ICME 2001*, Tokyo, Japan, Aug 2001

[59] M. Yeung B.L. Yeo *Time-Constrained Clustering for Segmentation of Video into Story Units,* Proc. Int. Conf. on Pattern Recognition, ICPR '96, Vol. C pp. 375-380, Vienna Austria, Aug. 1996.

[60] B.L. Yeo, M. Yeung *Classification, Simplification and Dynamic Visualization of Scene Transition Graphs for Video Browsing,* Proc. SPIE '98, Storage and Retrieval of Image and Video Databases VI, San Jose CA, Feb. 1998.

[61] M. Yeung, B.L. Yeo and B. Liu, *Segmentation of Video by Clustering and Graph Analysis*, Computer Vision and Image Understanding, V. 71, No. 1, July 1998.

[62] D. Yow, B.L.Yeo, M. Yeung, and G. Liu, "Analysis and Presentation of Soccer Highlights from Digital Video" *Proc. ACCV, 1995,* Singapore, Dec. 1995

[63] T. Zhang C.C Jay Kuo *Heuristic Approach for Generic Audio Segmentation and Annotation,* Proc. ACM Multimedia '99, pp. 67-76, Orlando FL, Nov. 1999.

[64] D. Zhong and S.F. Chang, "Structure Analysis of Sports Video Using Domain Models", *Proc. ICME 2001*, Tokyo, Japan, Aug. 2001

[65] D. Zhong *Segmentation, Indexing and Summarization of Digital Video Content* PhD Thesis, Dept. Of Electrical Eng. Columbia University, NY, Jan. 2001.