Computable Scenes and Structures in Films

Hari Sundaram and Shih-Fu Chang

***Abstract*—In this paper, we present a computational scene model and also derive novel algorithms for computing audio and visual scenes and within-scene structures in films. We use constraints derived from film-making rules and from experimental results in the psychology of audition, in our computational scene model. Central to the computational model is the notion of a causal, finite-memory viewer model. We segment the audio and video data separately. In each case, we determine the degree of correlation of the most recent data in the memory with the past. The audio and video scene boundaries are determined using local maxima and minima, respectively. We derive four types of computable scenes that arise due to different kinds of audio and video scene boundary synchronizations. We show how to exploit the local topology of an image sequence in conjunction with statistical tests, to determine dialogs. We also derive a simple algorithm to detect silences in audio.**

An important feature of our work is to introduce semantic constraints based on structure and silence in our computational model. This results in computable scenes that are more consistent with human observations. The algorithms were tested on a difficult data set: three commercial films. We take the first hour of data from each of the three films. The best results: computational scene detection: 94%; dialogue detection: 91%; and recall 100% precision.

***Index Terms*—Computable scenes, film-making production rules, joint audio-visual segmentation, structure discovery.**

I. Introduction

THIS paper deals with the problem of computing scenes within films by fusing information from audio and visual boundary detectors and visual structure. We also derive algorithms for detecting visual structures in the film. The problem is important for several reasons.

1) Automatic scene segmentation is the first step toward greater semantic understanding of the film.
2) Breaking up the film into scenes will help in creating film summaries, thus enabling a nonlinear navigation of the film.
3) In recent work, we have used these computable scenes in conjunction with the idea of visual complexity for generating visual skims [19].

There has been prior work on video scene segmentation using image data alone [7], [22]. In [22], the authors derive scene transition graphs to determine scene boundaries. However, cluster thresholds are difficult to set and must be manually tuned. In [7], the authors use a infinite, noncausal memory model to segment the video. We refine this idea of memory in our current work, but in a finite, causal setting. Prior work [11], [13], [16] 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. They do not examine the possibility of using the long-term consistency found in the audio data for segmentation. We shall discuss [5] and [6] in our section on experimental results.

There has been prior work on structure detection [22], [23]. Here, the authors begin with time-constrained clusters of shots and assign labels to each shot. Then, by analyzing the label sequence, they determine the presence of dialogue. This method critically depends upon cluster threshold parameters that need to be manually tuned.

In this paper, we develop notions of video and audio computable scenes (v-scenes and a-scenes) by making use of constraints stemming from rules governing camera placement, lighting continuity, as well as due to the psychology of audition. First, we refine the memory model found in [7], to cover both audio and video data. Second, we make our memory model causal and finite. 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 detect local maxima and minima, respectively, to determine scene change points.

We derive four types of computable scenes (c-scenes) that arise from different forms of synchronizations between a-scene and v-scene boundaries. We term these scenes *computable,* since they can be reliably computed using low-level features alone. In this paper, we do not deal with the *semantics* of a scene. Instead, we focus on the idea of determining a computable scene, which we believe is the first step in deciphering the semantics of a scene.

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. A key feature of our work is the idea of imposing semantic constraints based structural grouping and silence, on our computable scene model. This makes the segmentation result more consistent with human perception. Finally, we merge the results from segmentation, structure analysis and silence to come up with a list of c-scenes. Our results indicate that our approach performs well.

Manuscript received March 8, 2002. This work was supported in part by ONY Young Investigator Award N00014-98-1-0488. The associate editor coordinating the review of this paper and approving it for publication was Prof. Alberto Del Bimbo.

The authors are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: sundaram@ee.columbia.edu; sfchang@ee.columbia.edu).

Digital Object Identifier 10.1109/TMM.2002.802017

1520-9210/02$17.00 © 2002 IEEE

The rest of this paper is organized as follows. In Section II, we formalize the definition of a computable scene. In Section III, we present our memory model. In Section IV, we discuss techniques to determine video scene boundaries. In Section V, we discuss our topological framework for determining visual structure, while in Section VI, we discuss audio scene boundary detection. In Section VII, we discuss our technique to merge information from audio, video scene boundaries, structure detection and silences. In Section VIII and IX we present our experimental results and a discussion on possible model breakdowns. Finally, in Section X, we summarize our contribution and present our conclusions.

II. What is a Computable Scene?

In this section, we shall define the notion of a computable scene. We begin with a few insights obtained from understanding the process of film-making and from the psychology of audition. We shall use these insights in creating our computational model of the scene.

A. Insights From Film Making Techniques

The line of interest is an imaginary line drawn by the director in the physical setting of a scene [14]. During the filming of the scene, all the cameras are placed on one side of this line (also referred to as the 180° rule). This is because we desire successive shots to maintain the spatial arrangements between the characters and other objects in the location. The 180° rule has interesting implications on the computational model of the scene. Since all the cameras in the scene remain on the same side of the line in all the shots, there is an overlap in the field of view of the cameras (see Fig. 1). This implies that there will be a consistency to the chromatic composition and the lighting in all the shots. Film-makers also seek to maintain continuity in lighting amongst shots within the same physical location. This is done even when the shots are filmed over several days. This is because viewers perceive the change in lighting as indicative of the passage of time. For example, if two characters are shown talking in one shot, in daylight, the next shot cannot show them talking at the same location, at night.

B. The Psychology of Audition

The term *auditory scene analysis* was coined by Bregman in his seminal work on auditory organization [1]. In his psychological experiments on the process of audition, Bregman made many interesting observations, a few of which are reproduced as follows.

1) Related sounds seldom begin and end at the same time.
2) A sequence of sounds from the same source seem to change its properties smoothly and gradually over a period of time.
3) Changes that take place in an acoustic event will affect all components of the resulting sound in the same way and at the same time.

Bregman also noted that different auditory cues (i.e., harmonicity, common-onset, etc.) compete for the user's attention and depending upon the context and the knowledge of the user, will result in different perceptions. Different computational models (e.g., [3]) have emerged in response to those experimental observations. While these models differ in their implementations and differ considerably in the physiological cues used, they focus on short-term grouping strategies of sound.

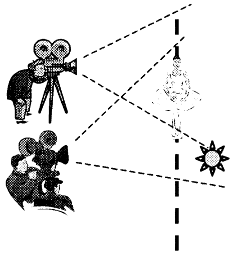


Fig. 1. Showing the line of interest (thick dashed line) in a scene. We also see the fields-of-view of the two cameras intersecting.

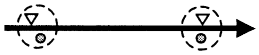
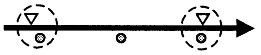
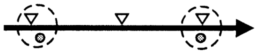
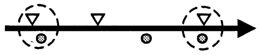
C. The Computable Scene Model

The constraints imposed by production rules in film and the psychological process of hearing lead us to the following definition of audio and video scenes. A video scene is a continuous segment of visual data that shows *long-term* consistency with respect to two properties: 1) chromaticity and 2) 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. 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 between two consecutive, synchronized audio visual scenes. This results in four cases of interest (see Table I). In this table, solid circles indicate audio scene boundaries, while triangles indicate video scene boundaries.

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. 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 scenes and Ac-V scenes).

The A-Vc (consistent audio, visuals change) scenes seem to occur under two circumstances. In the first case, the camera placement rules discussed in Section II-A are violated. These are montage 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 (e.g., Mtv videos). The second case consists of a sequence of v-scenes that individually obey the camera placement rules (and hence each have consistent 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 tim, e.g., a scene showing a journey, characterized by consistent audio track.

TABLE I
FOUR TYPES OF C-SCENES THAT EXIST BETWEEN CONSECUTIVE, SYNCHRONIZED AUDIO-VISUAL CHANGES

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	

Mixed mode (MM) scenes 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.). A c-scene type break-up of the first hour of the film (there were 642 shots) *Sense and Sensibility* reveals the following statistics—Pure: 65%, Ac-V: 21%, A-Vc:10%, and MM: 4%. The statistics from the other films are similar.

III. THE MEMORY MODEL

In order to segment data into scenes, we use a causal, first-in-first-out (FIFO) model of memory (see Fig. 2). This model is derived in part from the idea of coherence [7]. In our model of a listener, two parameters are of interest: 1) memory (this is the net amount of information (T_m) with the viewer) and 2) attention span (this is the most recent data (T_{as}) in the memory of the listener [typical values for the parameters are $T_m = 32$ s and $T_{as} = 16$ s]). 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 [7] dealt with a noncausal, 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 that will more faithfully mimic the human memory-model. This is done for *both* audio and video scene change detection.

IV. DETERMINING VIDEO SCENE BOUNDARIES

In this section, we shall describe the algorithm for v-scene boundary detection. The algorithm is based on notions of *recall* and *coherence*. We model the v-scene as a contiguous segment of visual data that is chromatically coherent and also possesses similar lighting conditions. A v-scene boundary is said to occur when there is a change in the long-term chromaticity and lighting properties in the video. This stems from the film-making constraints discussed in Section II-A. The video stream is converted into a sequence of shots using a sophisticated color and motion based shot boundary detection algorithm [9], that produces segments that have predictable motion and consistent chromaticity. A frame at a fixed time after the shot boundary is denoted to be the key-frame.

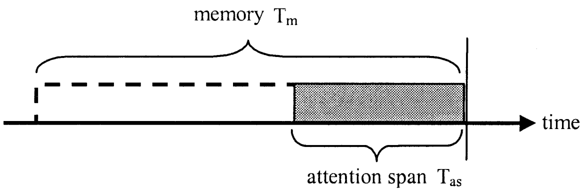


Fig. 2. Attention span T_{as} is the most recent data in the memory. The memory (T_m) is the size of the entire buffer.

A. Recall

In our visual memory model, the data is in the form of key-frames of shots and each shot occupies a definite span of time. The model also allows for the most recent and the oldest shots to be partially present in the buffer. A point in time (t_o) is defined to be a scene transition boundary if the shots that come after that point in time, do not recall [7] the shots prior to that point. The idea of recall between two shots a and b is formalized as follows:

$$R(a,b) = (1 - d(a,b)) \bullet f_a \bullet f_b \bullet \left(1 - \frac{\Delta t}{T_m}\right) \qquad (1)$$

where

$R(a,b)$ recall between the two shots a, b;
$d(a,b)$ L^1 color-histogram based distance between the key-frames corresponding to the two shots;
f_i ratio of the length of shot i to the memory size (T_m);
Δt time difference between the two shots.

The distance function $d(a,b)$, is computed as follows. First, we use 232 bin histogram in the HSV space (h:18, s:4, v:3, and 16 gray levels). The use of the HSV space accomplishes two things.

1) Perceptually close colors are close in this space.
2) The lighting changes are now easily detected (via changes in value).

The metric is the normalized color histogram difference.

The formula for recall indicates that recall is proportional to the length of each of the shots. This is intuitive since if a shot is in memory for a long period of time it will be recalled more easily. Again, the recall between the two shots should decrease if they are further apart in time. Note that the term *recall* is different from the one used in information retrieval (precision/recall).

We need to introduce the notion of a "shot-let." A shot-let is a fraction of a shot, obtained by breaking individual shots into δ s long chunks but could be smaller due to shot boundary conditions. Each shot-let is associated with a single shot and its representative frame is the key-frame corresponding to the shot. In our experiments, we find that $\delta = 1$ s works well. The formula for recall for shot-lets is identical to that for shots.

B. Computing Coherence

Coherence is easily defined using the definition of recall

$$C(t_o) = \frac{\left(\sum\limits_{a\in T_{as}} \sum\limits_{b\in\{T_m\setminus T_{as}\}} R(a,b)\right)}{C_{\max}(t_o)} \qquad (2)$$

where $C(t_o)$ is the coherence across the boundary at t_o and is just the sum of recall values between all pairs of shot-lets across the boundary at t_o. $C_{\max}(t_o)$ is obtained by setting $d(a,b)=0$ in the formula for recall (1) and re-evaluating the numerator of (2). This normalization compensates for the different number of shots in the buffer at different instants of time. Note that shot-lets essentially fine-sample the coherence function while preserving shot boundaries.

C. Detecting Coherence Minima

We detect the local coherence minima to determine if a v-scene boundary exists. To this end, we need to define two windows W_0 and W_1. W_0 is a window of size $2k+1$ points and W_1 is a smaller window centered in W_0 of size $k+1$. A typical value of k is 4. To determine if a minima exists, we first check if a minimum exists within W_1. If it does, we then need to impose conditions on this minima with respect to the coherence values in the larger window W_0 before we deem it to be a v-scene boundary.

First, we need to define three parameters relating to coherence values in W_0. α, β: they are, respectively, the difference between the maxima in the left and right half coherence windows and the minima value. γ : this is the difference between the minima and the global minima in W_0. Then, on the basis of these three values, we classify the minima into three categories (see Fig. 3):

Strong: $S \equiv \min(\alpha,\beta) > 0.3 \vee (\min(\alpha,\beta) > 0.1 \wedge \max(\alpha,\beta) > 0.4)$;

Normal: $N \equiv (\max(\alpha,\beta) > 0.1) \wedge (\min(\alpha,\beta) > 0.05) \wedge (\gamma < 0.1) \wedge (\neg S)$;

Weak: $W \equiv \max(\alpha,\beta) > 0.1 \wedge (\neg S) \wedge (\neg N)$.

The values above were determined using a 500 s training set obtained from the film *Sense and Sensibility*. The two window technique helps us detect the weak minima cases. The strong case is good indicator a v-scene boundary between two highly chromatically dissimilar scenes. The weak case becomes important when we have a transition from a chromatically consistent scene to a scene which is not as consistent. These are the $P \rightarrow A - Vc$, or $Ac - V \rightarrow A - Vc$ (and vice-versa) type scene transitions (see Table I).

D. Comparing Shot and v-Scene Detection

Now, we briefly comment on the differences between the shot detection algorithm and the v-scene detection algorithm. The shot detection algorithm [9] operates on the MPEG-1 compressed stream. It uses the following features — average color and variance, motion statistics (ratio of intra coded blocks to motion predicted blocks, ratio of number of forward to backward motion vectors). The detection is done over two short windows (0.2 s and 2 s) with a decision tree to come up with a robust algorithm. The performance is excellent over a wide range of datasets (precision 91% and recall 95%). We now highlight the key differences.

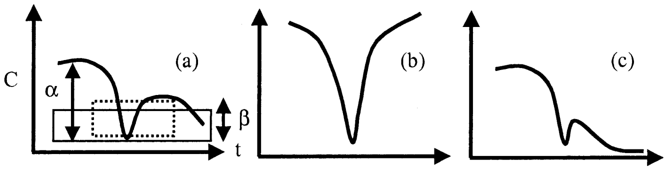


Fig. 3. Showing the (a) normal, (b) strong, and (c) weak coherence minima cases. The x-axis show time, while the y-axis indicates coherence.

The shot detection algorithm compares *two* frames and picks the local minima over a small window (typically $0.2 \sim 2$ s) to detect shots. However, the consistency of scene is a long-term ($\sim$ 30 s) *group* property, and is better determined by using the mutual information between two video segments (approximated by two groups of key-frames of shots).

The distance function for the v-scene detection takes into account the distance between the color-histogram two shots, the duration of each shot, and their temporal separation. There is no temporal weighting in the shot detection algorithm. In the next section, we discuss techniques for structure discovery.

V. Computing Visual Structure

In this section, we shall give an overview of some of the possible structures that exist in video sequences, an abstract representational technique and an algorithm for computing dialogs. The analysis that follows assumes that the video data has been segmented into shots and that each shot is represented by a single key-frame.

Structures (e.g., dialogs) contain important semantic information, and also provide complimentary information necessary to resolve v-scene boundaries. For example, in a dialog that contains very long shots (e.g., 25 s each) showing very different backgrounds, the algorithm in Section IV-B will generate v-scene boundaries after each shot. Computationally, this situation is no different from two long shots from completely chromatically different (but adjacent) v-scenes. Human beings easily resolve this problem by not only inferring the semantics from the dialogue, but also by recognizing the dialog structure and grouping the shots contained in it into one semantic unit.

A. The Topology of Shots

Structures in video shot sequences, have an important property that the structure is independent of the individual shot lengths. It is the topology (i.e., the metric relationships between shots, independent of the duration of the shots) of the shots that uniquely characterizes the structure.

B. The Topological Graph

Let $\mathbf{S} = \{\mathbf{I},\mathbf{d}\}$ be the metric space induced by the set of all images $\mathbf{I}$ in the video sequence by the distance function $\mathbf{d}$. The topological graph $T_G = \{V, E\}$ of a sequence of $\mathbf{k}$ images, is a fully connected graph, with the images at the vertices and where the edges specify the metric relationship between the images. The graph has associated with it, the topological matrix T_{MAT}, which is the k by k matrix where $T_{\mathrm{MAT}}(i,j)$ contains

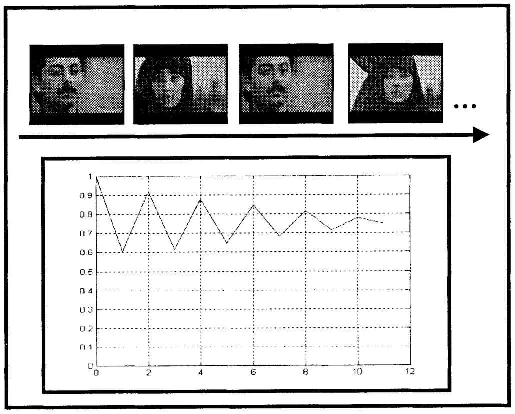


Fig. 4. Dialogue scene and its corresponding periodic analysis transform. Note the distinct peaks at $n = 2, 4 \ldots$.

the value of the edge connecting node i to node j in the graph. The idea of the topological graph is distinct from the scene transition graph [22], [23]. There, the authors cluster shots, and examine relationships between these clusters to determine scene change points as well as dialogs. Here, we are strictly interested in the topological property of a sequence of images and not in determining scene transitions.

C. Detecting Dialogs

A six image length dialog A-B-A-B-A-B, is completely specified with the following idealized topological relationship: $d(A, B) = 1$. A dialog has a specific local topological property: every second frame is alike while adjacent frames differ (Fig. 4). In the idealized topological matrix for the dialog, this appears as the 1st off-diagonal being all ones, the second off-diagonal being all zeros and the third off-diagonal being all ones. Hence we need to define a periodic analysis transform to estimate existence of this pattern in a sequence of N shot key-frames. Let o_i where $i \in \{0, N-1\}$ be a time-ordered sequence of images. Then

$$\Delta(n) \triangleq 1 - \frac{1}{N} \sum_{i=0}^{N-1} d(o_i, o_{\mathrm{mod}(i+n,N)}) \tag{3}$$

where

$\Delta(n)$ the transform;
d L^1 color-histogram based distance function;
mod usual modulus function.

The modulus function simply creates a periodic extension of the original input sequence. We shall use two statistical tests: the students t-test for the means and the F-test for the variances. These tests are used to compare two series of numbers and determine if the two means and the variance differ significantly.

Detecting Dialogues: We can easily detect dialogues using the periodic analysis transform. Let us assume that we have a time-ordered sequence of N key-frames representing different shots in a scene. Then, we do the following.

1) Compute the series $\Delta(n)$ (see Fig. 4).
2) Check if $\Delta(2) > \Delta(1)$ and $\Delta(2) > \Delta(3)$.
3) A dialogue is postulated to exist if one of two conditions in Step 2 is at least significant at $\alpha = 0.05$ and the other one is at least significant at $\alpha = 0.1$. Note that $\Delta(n)$ for each n is the mean of N numbers. We use the student's t-test to reject the null hypothesis that the two means are equal.

The Sliding Window Algorithm: We use a sliding window algorithm to detect the presence of a dialogs in the entire shot sequence for the video. Dialogs in films have an interesting rule associated with them: showing a meaningful conversation between m people requires at least $3m$ shots [12]. Hence in a dialogue that shows two participants, this implies that we must have a minimum of six shots. As a consequence, we analyze six frames at a time starting with the first shot key-frame. The algorithm is as follows.

```
1. Run the dialogue detector on the current window.
2. If no dialogue is detected, keep shifting the
   window to the right by one key-frame to the im-
   mediate right until either a dialogue has been
   detected or we have reached the end of the video
   sequence.
3. If a dialogue has been detected, keep shifting
   the starting point of the window by two key-frames,
   until we no longer have a statistically signifi-
   cant dialog or if we reached the end of the video
   sequence.
4. Merge all the overlapping dialog sequences just
   detected.
5. Move the starting point of the window to be the
   first frame after the last frame of the last suc-
   cessful dialog.
```

The sliding window algorithm can sometimes "overshoot" and "undershoot." i.e., it can include a frame before (or after) as being part of the dialog. These errors are eliminated by simply checking if the local dialog topological property holds at the boundaries. If not, we simply drop those frames. This results in an algorithm that generates statistically significant dialogs, with precise begin and end locations.

VI. Determining Audio Scenes

In this section, we present a brief description (due to space constraints) of our computable audio scene boundary detection framework. Earlier work is to be found in [17] and [18], with a detailed analysis and new results in [20]. We model the scene as a collection of sound sources. We further assume that the scene is *dominated* by a few of these sources. These dominant sources are assumed to possess stationary properties that can be characterized using a few features. A scene change is said to occur when the majority of the dominant sources in the sound change.

We model the audio data using three types of features: 1) scalar sequences; 2) vector sequences; and 3) single points. Features ([11], [15], [17], [23]) are extracted per section of the memory, and each section is T_{as} s long (the length of the attention span). We use six scalar features: 1) zero-crossing rate; 2) spectral flux; 3) cepstral flux; 4) energy; 5) energy variance;

and 6) the low-energy fraction. We determine three vector features: 1) cepstral vectors; 2) multichannel cochlear decomposition; and 3) mel-frequency cepstral coefficients. We also compute two point features: 1) spectral roll off point and 2) variance of the zero crossing rate. The point features are called so because just one value is obtained for the duration of the entire attention span. All other features (except for the low-energy fraction and the energy variance, which are computed per second), are obtained per 100 ms *frame* of the attention span. The cochlear decomposition was used because it was based on a psychophysical ear model. The cepstral features are known to be good discriminators. All the other features were used for their ability to distinguish between speech and music [11], [15], [23]. The scalar sequence of feature values are modeled to consist of three parts: 1) a trend (in order to incorporate Bregman's constraints); 2) a set of periodic components; and 3) noise.

A. Determining Correlations

We determine correlations of the feature values stored in the attention, with the data in the rest of the memory, to determine if a scene change point has occurred at t_o. At the end of this procedure, we have a sequence of distance values for each feature, at discrete time intervals of $\delta\, t$ i.e., at $t \in \{t_o + p\delta t\}$, where p is an integer. If a scene change was located at t_o, to the immediate left of the attention span we would intuitively expect the distance values to increase rapidly as the data ought to be dissimilar across scenes. We then compute β_{inc}, the rate of *increase* of the distance at time $t = 0$. The local maxima of the distance increase rate estimate β_{inc}, represents the scene change location point as estimated by that feature. Finally, we use a voting procedure amongst the features, to determine scene change location points.

B. Determining Silences

Silences become particularly useful in detecting c-scene boundaries where v-scene boundary occurs in a relatively silent section. There are two forms of silence in speech [15]: 1) within phrase silences and 2) between phrase silences. The within phrase silences are due to weak unvoiced sounds like /f/ or /th/ or weak voiced sounds such as /v/ or /m/. However, such silences are short usually $20 \sim 150$ ms long. In [15], the author uses a two class classifier using Gaussian models for each pause class, to come up with a threshold of 165 ms. However, others have used a threshold of 647 ms [4], for distinguishing significant pauses. In our experiments, we detect silences greater than 500 ms duration [20].

C. Determining Weak a-Scene Boundaries

We now define the notion of a weak a-scene boundary. This is useful when determining the rules for c-scene detection. A weak a-scene boundary has a significant amount of silence at the boundary. We make further distinctions based on the amount of silence present.

Compute the fraction of silence in a symmetric window ($2W_{AS}$ s long) around the a-scene boundary. Let L_{AS} and R_{AS} be the left and right silence fractions. i.e., the amount of silence in the left and right windows. Using the computed values of L_{AS} and R_{AS}, we make the following distinctions:

Pure Silence : $F_a \equiv \min(L_{AS,}R_{AS}) = 1;$
Silent : $S \equiv \max(L_{AS}, R_{AS}) = 1;$
Conversation : $C_p \equiv \min(L_{AS}, R_{AS}) = 0 \wedge \max(L_{AS}, R_{AS}) > 0.125 \wedge (\neg S).$

This is just a test of significant silence in speech (see Section VI-B).

VII. Integrating Audio, Silence, Video, and Structure

In this section, we discuss our algorithm to integrate information from the a-scene, v-scene boundary detection algorithms with the results of the structure and silence detection algorithms. The computational scene model in Table I, can generate c-scenes that run counter to grouping rules that human beings routinely use. Hence, the use of silence and structure detection imposes additional semantic constraints on the c-scene boundary detection algorithm.

A. Detecting c-Scene Boundaries

There are three principal rules for detecting c-scenes.

1) We detect a c-scene boundary (c-scene-b) whenever we can associate a v-scene boundary (v-scene-b) with an a-scene boundary (a-scene-b) that lies within a window of W_C s
2) We declare a c-scene-b to be present when normal v-scene-b's (see Section IV-C) intersect silent regions.
3) We always associate a c-scene boundary with strong v-scene boundary locations.

The first rule is the synchronization rule for detecting c-scenes. The window W_C is necessary as film directors deliberately do not exactly align a-scene and v-scene boundaries; at a perceptual level, this causes a smoother transition between scenes. There are some exceptions to this rule, which we discuss later in the section. The second rule is important as many transitions between c-scenes are silent (e.g., the first scene ends in silence and then the second scene shows conversation, which also begins with silence). In such cases, audio scene boundaries may not exist within W_C s of the v-scene.

The third rule becomes necessary when there is no detectable a-scene boundary within W_C s of a strong v-scene boundary. Strong v-scene boundaries occur as transitions between two v-scenes that are long in duration, and which differ greatly in chromatic composition. The notation used in the figures in this section: silence: gray box, structure: patterned box, solid dot: a-scene boundary, equilateral triangle: v-scene, solid right-angled triangle: weak v-scene. Now, we detail the steps in the algorithm.

```
Step 1: Remove v-scene or a-scene changes or silence
 within structured sequences (i.e., within dialogs
 and regular anchors) (Fig. 5). This is intuitive
 since human beings recognize and group structured
 sequences into one semantic unit.
Step 2: Place c-scene boundaries at strong (see Sec-
 tion IV-C) v-scene boundaries. Remove all strong
 v-scenes from the list of v-scenes.
```

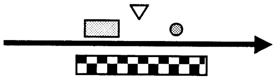


Fig. 5. Remove a-scenes, v-scenes, and detected silence, when present in structured sequences.

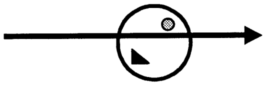


Fig. 6. Tight synchronization between weak v-scenes and nonweak a-scenes.

Step 3: If an a-scene lies within W_C s of a v-scene-b, place a c-scene boundary at the v-scene-b location. However, there are three exceptions:

1. Do not associate a weak v-scene-b with a weak a-scene-b.
2. If the v-scene-b is weak, it must synchronize with a nonweak a-scene-b that is within $W_C/2$ s i.e., we have tighter synchronization requirements for weak v-scene-b's (see **Fig. 6**).
3. Do not associate a normal v-scene-b with a weak a-scene-b marked as silent (see **Section VI-C**).

Step 4: Nonweak (see **Section IV-C**) v-scene boundaries (i.e., normal boundaries. Note that strong boundaries would have already been handled in Step 2) that intersect silent regions are labeled as c-scene boundaries. To determine whether a v-scene boundary intersects silence, we do the following

- Compute the fraction of silence in a symmetric window ($2W_{\mathrm{VS}}$ s long) around the v-scene boundary. Let L_{VS} and R_{VS} be the left and right silence fractions. i.e., the amount of data in the left and right windows that constitute silence.
- Then, declare a c-scene boundary if: $L_{\mathrm{VS}} > 0.8 \quad \vee \quad R_{\mathrm{VS}} > 0.8$.

Now, we have a list of c-scenes, as well as lists of singleton video and audio scene boundaries. The c-scenes are then post-processed to check if additional structure is present.

B. Post-Processing c-Scenes

Once we have detected all the c-scenes, we use a conservative post-processing rule to eliminate false alarms. An irregular anchor shot in a semantic scene is a shot that the director comes back to repeatedly, but not in a regular pattern, within the duration of the semantic scene. This is known in film-making, as the "familiar-image" [12]. We check if an anchor is present across adjacent scenes and merge them, if present. We make this rule transitive: i.e., if we have three c-scenes A, B, C, in succession, and if A and B have share a regular anchor and B and C share a (possibly different) irregular anchor, then c-scenes A, B and C are merged into one c-scene.

VIII. Experimental Results

In this section, we shall discuss the experimental results of our algorithms. The data used to test our algorithms is complex: we have three 1-h segments from three diverse films in English: 1) *Sense and Sensibility*; 2) *Pulp Fiction*; and 3) *Four Weddings and a Funeral*. In the tables that follow, we have abbreviated the films as *sense*, *pulp*, and *four*, respectively. We begin with a section that explains how the labeling of the ground truth data was done (see Table II). It is followed by sections on c-scene boundary detection and structure detection.

TABLE II
Ground Truth Data

Film	A-s	V-s	C-Scenes				
			P	Ac-V	A-Vc	MM	Total
Sense	69	57	33	11	5	2	51
Pulp	45	40	23	8	5	0	36
Four	72	61	31	8	8	6	53

TABLE III
C-Scene Detector Results

Film	H	M	FA	CR	Shots	NCL	P	R
Sense	48	3	21	570	642	591	0.70	0.94
Pulp	29	7	19	423	478	442	0.60	0.81
Four	41	12	18	665	736	683	0.70	0.77

A. Labeling the Ground Truth

The audio and the video data were labeled separately (i.e., label audio without watching the video and label video without hearing the audio). This was because when we use *both* the audio and the video (i.e., normal viewing of the film) we tend to label scene boundaries based on the semantics of the scene. Only one person (the first author) labeled the data. Due to space constraints, we summarize our labeling procedure.

We attempt to label the audio and video data into coherent segments. From empirical observations of film data, it became apparent that for a group of shots to establish an independent context, it must last at least 8 s Hence, all the v-scenes that we label must last more than 8 s We also set the minimum duration of an a-scene to be 8 s Then, the labeling criteria were as follows:

a) Do not mark v-scene boundaries in the middle of dialogs or regular anchors, instead mark structure detection points at the beginning and end of the dialogs/regular anchors.

b) When encountering montage sequences (see Section II-C), only label the beginning and end of the montage sequence.

c) When encountering silences greater than 8 s label the beginning and ends of the silence.

d) When encountering speech in the presence of music, label the beginning and the end of the music segment.

e) Do not mark speaker changes.

Table III shows the data obtained from labeling the audio and the video data separately — audio scenes (A-s) and video scenes (V-s). The c-scenes are broken up into constituent units: P-pure, Ac-V (audio changes, visuals consistent), A-Vc: audio consistent, visuals change, and MM: mixed mode. Note that an a-scene and v-scene are denoted to be synchronous if they less than 5 s apart.

TABLE IV
V-SCENE DETECTOR RESULTS FOR THE THREE FILMS

Film	H	M	FA	CR	Shots	NCL	P	R
Sense	52	5	22	563	642	585	0.70	0.91
Pulp	34	6	21	417	478	438	0.62	0.85
Four	49	12	41	634	736	675	0.55	0.80

TABLE V
THE DIALOGUE DETECTOR RESULTS

Film	H	M	FA	Precision	Recall
Sense	28	3	0	1.00	0.91
Pulp	11	2	2	0.84	0.84
Four	16	4	1	0.94	0.80

B. Scene Change Detector Results

In this section, we discuss the scene change detector results. First, we discuss the parameters that we need to set. The memory and attention span sizes for the audio and video scene detection algorithm, and the synchronization parameter W_C, which we set to 5 sec (i.e., c-scene boundary is marked when the audio and video scenes are within 5 s of each other). For detecting video coherence, video we set the attention span to be 8 s (in accordance with our labeling rule) and the size of the memory is set to 24 s In general, increasing the memory size reduces false alarms, but increases misses.

In evaluating our results, we shall compare against c-scenes against the total number of shots in the film, since they are all candidate c-scene change points. Second, it is important to note that we are dealing with an asymmetric two-class problem (scene change vs. nonscene change) where the number of ground truth scene change locations is typically less than 10% of the total number of shots.

We now present results for c-scene and v-scene detection in Tables III and IV. These results are for the entire duration of the film (each film is one hour long) and for all types of transitions. We us the following notation: H: hits; M: misses; FA: false alarms; CR: correct rejection; shots: the number of shots detected by the shot detection algorithm; NCL: non-scene change locations (this is just the number of shots less the number of ground truth scene change locations); P: precision (i.e., hits/(hits + falsealarms); and R: recall (i.e., hits/(hits + misses).

The result shows that the c-scene and the v-scene detectors work well. The recall for c-scene detectors varies between 77 $\sim$ 94% while the precision varies between $60 \sim 70\%$. The recall for the v-scene detector varies between $80 \sim 91\%$ while the precision varies between $55 \sim 70\%$. Note that the correct rejection is excellent — around 95% across all cases. We now discuss two aspects relevant to our results —sources of error, and the relevance of the low precision.

Shot Detector Errors: Misses in the video shot boundary detection algorithm cause the wrong key-frame to be present in the buffer, thus causing an error in the minima location

Labeling Uncertainty: Labeling the audio data is time consuming and often there is genuine uncertainty about the a-scene change location. This can happen for example, when we have a long sequence of low amplitude sounds (e.g., background sounds, soft footsteps) that changes into silence. Thus, this can translate to c-scene misses. This uncertainty may be mitigated to a certain extent by using additional labelers, but is difficult to eliminate altogether.

Low Precision: It is clear that our algorithm apparently over-segments the data. However, a detailed look at the false alarms indicates that these scenes are correct from a computational standpoint (i.e., satisfied the requirements for a change), but were wrong semantically (e.g., a conversation shot against a wall, that continues against a backdrop of a large window that has sun shining through). This seems to imply that even though we had signal-level guidelines for labeling the ground truth, the labeler ended up labeling the data on a semantic level. One of the goals of our work is to generate video summaries by condensing computable scenes [19]. There we compress each computable scene via an analysis of the visual complexity of the shots and by using film syntax. In such tasks, "over-segmentation" resulting in c-scenes that are reasonable from a computational standpoint do not affect the results; there, misses in c-scenes are more problematic.

C. Structure Detection Results

In this section, we present our structure detection results. The statistical tests that are central to the dialogue detection algorithm make it almost parameter free. These test are used at the standard levels of significance ($\alpha = 0.05$). The sliding window size T_w (six frames). The results of the dialog detector (Table V) show that it performs very well. The best result is a precision of 1.00 and recall of 0.91 for the film *Sense and Sensibility*. The misses are primarily due to misses by the shot-detection algorithm. Missed key-frames will cause a periodic sequence to appear less structured.

D. Comparison With Related Work

We now briefly compare our results with prior work. Note that the these algorithms use different datasets, and also have different objectives in mind. Hence direct performance comparisons are difficult.

Scene Detection: There has been some prior work on joint audio-visual segmentation [6]. In [6], the authors, denote a scene change point to occur at a frame, which exhibits:1) a shot cut; 2) an audio change; and 3) a high-motion change. However, these are short term phenomena, and the they do not investigate long term correlations in either audio or video data, or the relationship of these detectors to the presence of structure (e.g., dialogs). Also, by focusing on synchronous audio visual events, they overlook the possibility of having single, unsynchronized, but semantically important audio or visual events.

There has been some prior work that analyzed film data [5], [10]. In [5], the authors use visual features alone to determine a logical story unit (LSU): a collection of temporally interrelated events. The LSU is detected using a single link clustering algorithm (via subblock matching across key-frames) with cluster thresholds that change with the content of the cluster. This is done on the shots while ignoring the duration. However, importantly, the duration of a scene can vary greatly with directorial style and semantics. In [10], the authors aim at automating the process of creating (using visual features) video abstracts, given a time budget, not segmentation. Neither of these works attempts to incorporate film-making constraints on the minimum number of shots in a scene, dialogs, or examines the inter-relationships between audio and video scene boundaries.

Dialog Detection: There has been prior work [22], [23] to determine dialogs in video sequences. The results there are also good, however, they need to set cluster threshold parameters. In contrast, our algorithm is almost parameter free. The main contribution of our work is to present an abstract conceptual framework in terms of the topological graph — this framework is easily extended for systematically detecting arbitrary structures using robust statistical methods.

IX. Discussing c-Scene Detector Breakdowns

In this section, we shall discuss three situations that arise in different film-making situations. In each instance, the 180° rule is adhered to and yet our assumption of chromatic consistency across shots is no longer valid.

Sudden Change of Scale: A sudden change of scale accompanied by a change in audio cannot be accounted for in our algorithm. This can happen in the following case: a long shot[1] shows two people with low amplitude ambient sound; then, there is a sudden close up of one person as he starts to speak. Detecting these breaks, requires understanding the semantics of the scene.

Widely Differing Backgrounds: This can happen in two circumstances: 1) a right-angled camera pan and 2) a set up involving two cameras. In the first case, the coherence model will show a false alarm for v-scene, and if accompanied by an a-scene change, this will be labeled as a c-scene break. In the second case, we have two opposing cameras having no overlap in their field-of-view causing an apparent change in the background. This can happen for example, when the film shows one character inside the house, talking through a widow to another character who is standing outside.

Change of Axis: The axis of action (i.e., the line of interest) can change in several ways. Let us assume that we have a scene which shows a couple engaged in conversation. The director can change the axis of action within a scene [1] by 1) moving the one of two people across the room or 2) by using a circular tracking shot around the couple, thereby establishing a new axis in both cases. The motion continuity alerts the viewer about this change.

These situations are problematic (incorrect boundary placement) only when they take place over long time scales (i.e., camera pans and stays there); Short term changes will be handled by our algorithm. Also, if these changes exhibit structure, (i.e., in a dialog or in a regular anchor), these false alarms will be eliminated. One way to overcome the slow-pan situation is to incorporate motion information into our decision framework. Motion continuity will be of help in detecting the change of axis scenario. Clearly, our computational model makes simplifying assumptions concerning the chromatic consistency of a v-scene, even when film-makers adhere to the 180° rule.

X. Conclusion

We now summarize the work presented in this paper. We have presented a computational scene model for films. We show 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. A v-scene exhibits long-term consistency with regard to lighting conditions and chromaticity of the scene. The a-scene shows long term consistency with respect to the ambient audio. We believe that the computable scene formulation is the first step toward deciphering the semantics of a semantic scene.

We 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. The maxima of the rate of increase of the correlation is used to determine scene change points. We use ideas of recall and coherence in our video segmentation algorithm. The algorithm works by determining the coherence amongst the shot-lets in the memory. A local minima criterion determines the scene change points.

We derived a periodic analysis transform based on the topological properties of the dialog to determine the periodic structure within a scene. We showed how one can use the student's t-test to detect the presence of statistically significant dialogues. We also showed how to determine silences in audio.

We derived semantic constraints on the computable scene model, and showed how to use the silence and structure information along with audio and video scene boundaries to resolve certain ambiguities. These ambiguities cannot be determined with using just the a-scene and the v-scene detection models.

The scene segmentation algorithms were tested on a difficult test data set: 3 h from commercial films. They work well, giving a best c-scene detection result of 94%. The structure detection algorithm was tested on the same data set giving excellent results: 91% recall and 100% precision. We believe that the results are very good when we keep the following considerations in mind: 1) the data set is complex and 2) the shot cut detection algorithm had misses that introduced additional error.

Contributions:

1) A computational scene model that incorporates the synergy between audio, video, and structure in the data.
2) A finite, causal memory model framework for segmenting both audio and visual data.
3) An abstract topological framework for arbitrary structures video in a robust manner. We show an algorithm for detecting dialogs in this framework.
4) The features and the models used in segmentation incorporate production rules from film-making as well as due to the psychology of audition.
5) The idea that we need top-down structural grouping rules to improve segmentation results.
6) Constraints for merging information from different modalities (audio, video, silence and structure) to ensure that the resulting segmentation is consistent with human perception.

Future Work: There are several clear improvements possible to this work.

1) The computational model for the detecting the video scene boundaries is limited, and needs to tightened in

[1] The size (long/medium/close-up/extreme close-up) refers to the size of the objects in the scene relative to the size of the image.

view of the model breakdowns discussed. One possible improvement is to do motion analysis on the video and prevent video scene breaks under smooth camera motion.

2) The v-scene detection algorithm should dynamically adapt to the low-contrast scenarios to improve performance.
3) 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.

Current work includes generating video skims using these computable scenes [19].

ACKNOWLEDGMENT

The authors would like to thank D. Zhong for help with the shot boundary detection algorithm.

REFERENCES

[1] D. Bordwell and K. Thompson, *Film Art: An Introduction*, 6th ed. New York: McGraw-Hill, 2000.

[2] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press, 1990.

[3] D. P. W. Ellis, "Prediction-driven computational auditory scene analysis," Ph.D. dissertation, The Department of Electrical Engineering and Computer Science, Massuchusetts Institute of Technology, Cambridge, MA, 1996.

[4] B. Grosz and J. Hirshberg, "Some intonational characteristics of discourse structure," in *Proc. Int. Conf. Spoken Lang. Processing*, 1992, pp. 429–432.

[5] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 580–88, June 1999.

[6] J. Huang, Z. Liu, and Y. Wang, "Integration of audio and visual information for content-based video segmentation," in *Proc. ICIP*, Chicago, IL, Oct. 1998, pp. 526–30.

[7] J. R. Kender and B. L. Yeo, "Video Scene Segmentation Via Continuous Video Coherence," in *Proc. CVPR*, Santa Barbara, CA, Jun. 1998.

[8] R. Lienhart *et al.*, "Automatic movie abstracting," Praktische Informatik IV, Univ. of Mannheim, Mannheim, Germany, TR-97–003, 1997.

[9] D. Zhong, "Segmentation, indexing and summarization of digital video content," Ph.D. dissertation, Dept. Elect. Eng., Columbia Universty, New York, 2001.

[10] S. Pfeiffer *et al.*, "Abstracting digital movies automatically," *J. Vis. Commun. Image Represent.*, vol. 7, pp. 345–53, Dec. 1996.

[11] E. S. M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. ICASSP*, Munich, Germany, Apr. 1997.

[12] S. Pfeiffer *et al.*, "Automatic audio content analysis," in *Proc. ACM Multimedia*, Boston, MA, Nov. 1996, pp. 21–30.

[13] J. Saunders, "Real time discrimination of broadcast speech/music," in *Proc. ICASSP*, Atlanata, GA, May 1996, pp. 993–6.

[14] S. Sharff, *The Elements of Cinema: Toward a Theory of Cinesthetic Impact*. New York: Columbia Univ. Press, 1982.

[15] L. J. Stifelman, "The audio notebook: pen and paper interaction with structured speech," Ph.D. dissertation, Program in Media Arts and Sciences, Sch. Architecture and Planning, Mass. Inst. Technol., Cambridge, 1997.

[16] S. Subramaniam *et al.*, "Toward robust features for classifying audio in the cuevideo system," in *Proc. ACM Multimedia*, Orlando, FL, Nov. 1999, pp. 393–400.

[17] H. Sundaram and S. F. Chang, "Audio Scene Segmentation Using Multiple Features, Models And Time Scales," in *Proc. ICASSP*, Istanbul, Turkey, Jun. 2000.

[18] ——, "Determining computable scenes in films and their structures using audio-visual memory models," in *Proc. ACM Multimedia*, Los Angeles, CA, Nov. 2000, pp. 95–104.

[19] ——, "Constrained utility maximization for generating visual skims," in *IEEE Workshop CBAIVL in conjunction with IEEE CVPR*, Kauai, Hawaii, Dec. 2001.

[20] ——, "Computable audio scene boundary detection using a listener memory model," Department of Electrical Engineering, Columbia University, New York, ADVENT, 2001.

[21] S. Uchihashi *et al.*, "Video Manga: Generating Semantically Meaningful Video Summaries ," in *Proc. ACM Multimedia*, Orlando, FL, Nov. 1999, pp. 383–92.

[22] M. Yeung and B. L. Yeo, "Time-constrained clustering for segmentation of video into story units," in *Proc. ICPR*, vol. C, Vienna, Austria, Aug. 1996, pp. 375–380.

[23] ——, "Video content characterization and compaction for digital library applications," in *Proc. SPIE '97, Storage and Retrieval of Image and Video Databases V*, San Jose, CA, Feb. 1997.

[24] T. Zhang and C. C Jay Kuo, "Heuristic approach for generic audio segmentation and annotation," in *Proc. ACM Multimedia*, Orlando, FL, Nov. 1999, pp. 67–76.

Hari Sundaram received the M.S. degree in electrical engineering from the State University of New York at Stony Brook in 1995 and the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, in 1993. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, Columbia University, New York. He is interested in issues relating to signal representation and its attendant applications to multimedia analysis and retrieval, signal processing, and computer vision. Other interests include approximate algorithms, processing with dynamic computational constraints. His Ph.D. work under Prof. Shih-Fu Chang focuses on video summarization.

Mr. Sundaram received the Best Paper Award in the area of video retrieval.

Shih-Fu Chang received the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1993.

He is with the Department of Electrical Engineering at Columbia University, where he currently leads Columbia's Digital Video/Multimedia Laboratory and ADVENT industry-university research consortium, which focuses on representation, manipulation, retrieval, and transmission of multimedia content. He leads digital video research within several cross-disciplinary projects at Columbia, including its Health Care Digital Library Project supported by NSF's DLI Phase II initiative.

Dr. Chang received a Navy ONR Young Investigator Award in 1998, a Faculty Development Award from IBM in 1995, a CAREER Award from the National Science Foundation in 1995, and three Best Paper Awards in the areas of video representation and searching. He actively participates in international conferences and standardization efforts, such as MPEG-7. He has been a General Co-chair of ACM Multimedia Conference 2000, an Associate Editor for several journals, and a Consultant in several new media technology companies. He is currently a Distinguished Lecturer of IEEE Circuits and Systems Society in the area of multimedia technologies and applications.