Determining Computable Scenes in Films and their Structures using Audio-Visual Memory Models

Hari Sundaram
Dept. of Electrical Engineering,
Columbia University,
New York New York 10027.
sundaram@ctr.columbia.edu

Shih-Fu Chang
Dept. of Electrical Engineering,
Columbia University,
New York New York 10027.
sfchang@ctr.columbia.edu

ABSTRACT

In this paper we present novel algorithms for computing scenes and within-scene structures in films. We begin by mapping insights from film-making rules and experimental results from the psychology of audition into a computational scene model. We define a computable scene to be a chunk of audio-visual data that exhibits long-term consistency with regard to three properties: (a) chromaticity (b) lighting (c) ambient sound. Central to the computational model is the notion of a causal, finite-memory 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 respective scene boundaries are determined using local minima and aligned using a nearest neighbor algorithm. We introduce the idea of a discrete object series to automatically determine the structure within a scene. We then use statistical tests on the series to determine the presence of dialogue. The algorithms were tested on a difficult data set: five commercial films. We take the first hour of data from each of the five films. The best results: scene detection: 88% recall and 72% precision, dialogue detection: 91% recall and 100% precision.

Keywords

Computable scenes, scene detection, shot-level structure, films, discrete object series, memory models.

1. INTRODUCTION

This paper deals with the problem of computing scenes within films using audio and visual data. We also derive algorithms for shot-level structures that exist within each scene. The problem is important for several reasons: (a) automatic scene segmentation is the first step towards greater semantic understanding of the film (b) breaking up the film into scenes will help in creating film summaries, thus enabling a non-linear navigation of the film. (c) the determination of visual structure within each scene (e.g. dialogues), will help in the process of visualizing each scene in the film summary.

There has been prior work on video scene segmentation using image data alone [8], [19]. In [8], the authors derive scene transition graphs to determine scene boundaries. Their method assumes a presence of repetitive shot structure within a scene. While this structure is present in scenes such as interviews, it can be absent from many scenes in commercial films. This can happen, for example, when the director relies on fast succession of shots to heighten suspense or uses a series of shots to merely develop the plot of the film.

Prior work [14], [15], [20] 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 environmental sounds etc. They do not examine the possibility of using the long-term consistency found in the audio data for segmentation. Audio data has been used for identifying important regions [6] or detecting events such as explosions [9] in video skims. These skims do not segment the video data into scenes; the objective there is to obtain a compact representation.

There has been prior work on structure detection [19], [20]. There, 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 the clustering parameters that need to be manually tuned.

In this paper, we derive the notion of a computable scene using rules from film-making and from experimental observations in the psychology of audition. A computable scene exhibits long-term consistency with respect to three properties: (a) chromatic composition of the scene (b) lighting conditions and (c) ambient audio. We term such a scene as *computable*, since it 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 present algorithms for determining computable scenes and periodic structures that may exist within such scenes. We begin with a causal memory model based in part on the model in [8]. The model has two parameters: (a) an analysis window that stores the most recent data (the attention span) (b) the total amount of data (memory).

In order to segment the data into audio scenes, we compute correlations amongst the envelopes of the audio features in the attention-span with feature envelopes 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. In both cases, we use a local minima for detecting a scene change and the audio and video scene boundaries are aligned using a simple time-constrained nearest neighbor approach.

The visual structure within each computable scene is determined using a novel idea of a discrete object series. The series computes the degree of periodicity amongst a time-ordered sequence of images (key-frames of shots). We use the Student's t-test in conjunction with a simple rule on this series, to detect the presence of a dialogue. In contrast to [19], [20] which require manually tweaked cluster diameter threshold parameters, this algorithm is almost parameter free. Our experiments show that the scene change detector and the intra-scene structure detection algorithm show good results.

The rest of this paper is organized as follows. In the next section, we formalize the definition of a computable scene. In section 3, we present an algorithm for the detection of such computational scenes. In section 4, we derive algorithms for automatically determining periodic structures within a scene. In section 5, we present our experimental results. In section 6 we discuss shortcomings of our model and finally in the section 7, we present our conclusions.

2. 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.

2.1 Insights from Film Making Techniques

The line of interest is an imaginary line drawn by the director in the physical setting of a scene [4]. During the filming of the scene, all the cameras are placed on one side of this line. This is because we desire successive shots to maintain the spatial arrangements between the characters and other objects in the location. As a consequence there is no confusion in the mind of the viewer about the spatial arrangements of the objects in the scene. He (or she) can instead concentrate on the dramatic aspect of the scene. It is interesting to note that directors willingly violate this rule only in very rare circumstances[1].

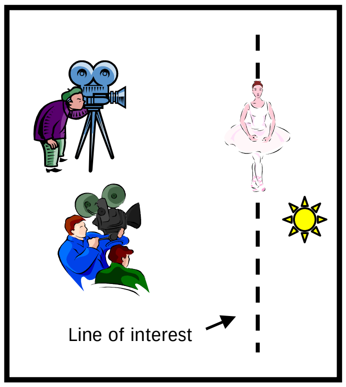

Figure 1: showing the line of interest in a scene.

The line-of-interest 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. This implies that there will be a consistency to the chromatic composition and the lighting in all the shots.

2.2 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 below:

- Unrelated sounds seldom begin and end at the same time.
- A sequence of sounds from the same source seem to change its properties smoothly and gradually over a period of time. The auditory system will treat the sudden change in properties as the onset of a new sound.
- 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. For example, if we are walking away from the sound of a bell being struck repeatedly, the amplitude of all the harmonics will diminish gradually. At the same time, the harmonic relationships and common onset[2] are unchanged.

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. However, the role played by higher forms of knowledge in grouping is yet to be ascertained.

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. Notably, Bregman's observations indicate that long-term grouping strategies are also used by human beings (e.g. it is easy for us to identify a series of footsteps as coming from one source) to group sound.

2.3 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 a scene: It is a continuous segment of audio-visual data that shows *long-term*[3] consistency with respect to three properties:

- Chromaticity
- Lighting conditions

[1] This is so infrequent that directors who transgress the rule are noted in the film theory community. e.g. Alfred Hitchcock willingly violates this rule in a scene in his film *North by Northwest* thus adding suspense to the scene [4].

[2] Different sounds emerging from a single source begin at the same time.

[3] Analysis of experimental data (one hour each, from five different films) indicates that the scenes in the same location (e.g. in a room, in the marketplace etc.) are typically 40~50 seconds long.

- Ambient sound

Table 1: Several scenarios are examined using our c-scene definition. These would normally be viewed as a single "normal" scene.

Scenario	Shot-sequence	C-Scenes	Explanation
Alice goes home to read a book.	(a) She is shown entering the room (b) Shown picks up the book (c) We see her reading silently (d) while she is reading we hear the sound of rain.	2	One consistent visual but two consistent chunks of audio.
Alice goes to sleep.	(a) She is shown reading on her bed (b) she switches off the light and room is dark.	2	Two consistent visuals but the audio is consistent over both video segments.
Bob goes for a walk	(a) He switches on his handy-cam inside the house and walks outside. Note, this is one single camera take.	2	There are two consistent visuals (inside/outside) as well as two consistent chunks of audio.

We denote this to be a *computable* scene since these properties can be reliably and automatically determined using low-level features present in the audio-visual data. We need to examine the relationship between a computable scene (abbreviated as c-scene) and normal notions of a shot and a scene. A shot is a segment of audio-visual data filmed in a single camera take. A scene is normally defined to be sequence of shots that share a common semantic thread. Table 1 examines the impact of the c-scene definition for several scenarios.

The semantics of a normal scene within a film, are often difficult to ascertain. While a collection of shots may have objects that are meaningful without context (e.g. a house, a man, a woman the colors of the dress etc.), the collection of shots are infused with meaning only with regard to the context.

The context is established due to two factors: the film-maker and the viewer. The film-maker infuses meaning to a collection of shots in three ways: (a) by *deciding* the action in the shots (b) the kind of shots that precede this scene and the shots that follow it (c) and finally by the manner in which he *visualizes*[4] the scene. All three methods affect the viewer, whose *interpretation* of the scene depends on his world-knowledge. Hence, if the meaning in a scene is based on factors that cannot be measured directly, it is imperative that we begin with a scene definition in terms of those attributes that are measurable and which lead to a consistent interpretation. We believe that such a strategy will greatly help in deciphering the semantics of the c-scene at a later stage.

[4] In order to show tension in a scene, one film-maker may have fast succession of close-ups of the characters in a scene. Others may indicate tension by showing both characters but changing the music.

2.4 The C-Scene Definition

We wished to validate the computational scene definition, which appeared out of intuitive considerations, with actual film data. The data was diverse with one hour segments from three English language films and two foreign films[5].

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. However, the definition does not work well for montage[6] sequences. In such sequences, we observed a long-term consistency of the ambient audio. We need to define a c-scene in order to accommodate different production styles. We now make two distinctions:

1. **N-type:** These scenes (or normal scenes) fit our original definition of a scene: they are characterized by a long-term consistency of chromatic composition, lighting conditions and sound.
2. **M-type:** These scenes (or montage/Mtv scenes) 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. However M-type scenes will be assumed to be characterized by a long-term consistency in the audio track.

In this paper, we narrow our focus to derive algorithms that detect two adjacent N-type scenes. We will not handle the two cases when we have either (a) two adjacent M-type scenes or (b) an N-type scene that borders an M-type scene. Analysis of the ground truth indicates that these two transitions constitute about 25% of all the transitions. Henceforth, for the sake of brevity, we shall use the term "scene" for our notion of a computable scene (c-scene).

3. DETECTING SCENES

We begin the process of scene detection by first detecting audio and video scene segments separately and then aligning the two by a simple nearest neighbor algorithm.

This section has four areas of focus: In section 3.1, we develop the idea of a memory model. In sections 3.2 and 3.3, we build upon some early techniques in [16], [17] for automatic audio and video scene detection. In section 3.4 we present a simple nearest-neighbor algorithm for aligning the two scene detector results.

3.1 A Memory Model

In order to segment data into scenes, we use a causal, first-in-first-out (FIFO) model of memory (figure 2). This model is derived in part from the idea of coherence [8].

[5] The English films: *Sense and Sensibility, Pulp Fiction, Four Weddings and a Funeral.* The foreign films: *Farewell my Concubine (Chinese), Bombay (Hindi).*

[6] 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.

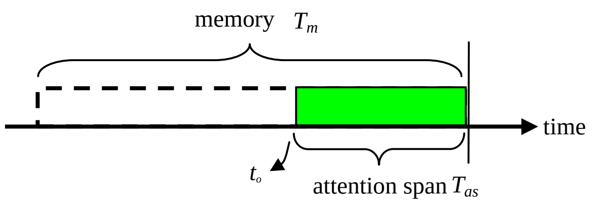

Figure 2: The attention span (T_{as}) is the most recent data in the buffer. The memory (T_m) is the size of the entire buffer. Clearly, $T_m \geq T_{as}$.

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. 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 [8] 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 finiteness of the 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.

3.2 Determining Audio Scenes

In this section we present our algorithm for audio-scene segmentation. We model the *audio-scene* as a collection of a few dominant sound sources. These sources are assumed to possess stationary properties that can be characterized using a few features. An audio-scene change is said to occur when the majority of the dominant sources in the sound change. A more detailed description of audio scene segmentation is to be found in [16].

3.2.1 Features and Envelope Models

We use ten different features [13], [14], [15], [16], [20] in our algorithm: (a) cepstral-flux (b) multi-channel cochlear decomposition (c) cepstral vectors (d) low energy fraction (e) zero crossing rate (f) spectral flux (g) energy (h) spectral roll off point. We also use the variance of the zero crossing rate and the variance of the energy as additional features. The cochlear decomposition was used because it was based on a psychophysical ear model. The cepstral features are known to be good discriminators [13]. All the other features were used for their ability to distinguish between speech and music [14], [15], [20]. Features are extracted per *frame* (100ms. duration) for the duration of the analysis window.

Given a particular feature *f* and a finite time-sequence of values, we wish to determine the behavior of the envelope of the feature. The feature envelopes are force-fit into signals of the following types: constant, linear, quadratic, exponential, hyperbolic and sum of exponentials. All the envelope (save for the sum of exponentials case) fits are obtained using a robust curve fitting procedure [5]. We pick the fit that minimizes the least median error. The envelope model analysis is only used for the scalar variables. The vector variables (cepstra and the cochlear output) and the aggregate variables (variance of the zero-crossing rate and the spectral roll off point) are used in the raw form.

3.2.2 Detecting a Scene Change

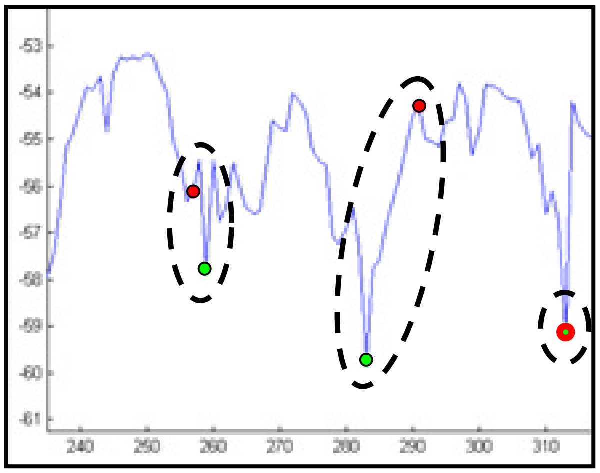

Figure 3: Audio detector results for a part of the audio track of the film *Sense and Sensibility*. The red dots show the ground truth label while the green dots show the detector result. In the first two cases the result is very close while in the third case there is an exact match. The x-axis shows the time in sec. while y-axis shows the detector magnitude.

Let us examine the case where a scene change occurs just to the left of the listeners attention span. First, for each feature, we do the following:

1. Place an analysis window of length T_{as} (the attention-span length) at t_o and generate a sequence by computing a feature value for each frame (100 ms duration) in the window.
2. Determine the optimal envelope fit for these feature values.
3. Shift the analysis window back by Δt and repeat steps 1. and 2. till we have covered all data in the memory.

We then define a local correlation function per feature, using the sequence of envelope fits. The correlation function C_f for each feature *f* is then defined as follows:

$$C_f(m\Delta t) = 1 - d(f(t_o, t_o - t_{as}), f(t_o + m\Delta t, t_o + m\Delta t - t_{as})) \quad (1)$$

where, $f(t_1, t_2)$ represents the envelope fit for feature *f* for the duration $[t_1,t_2]$. Clearly, $m \in [0..-N]$, where $N \equiv (T_m - T_{as})/\Delta t$. Δt is the duration by which the analysis window is shifted. and *d* is the Euclidean metric[7] on the envelopes For the vector and the aggregate data, we do not compute the distance between the windows using envelope fits but use a L^2 metric on the raw data. In our experiments we use $\Delta t = 1$ sec.

We model the correlation decay as a decaying exponential [16]: $C_i(t) = \exp(b_i t),\ t < 0$ where C_i is the correlation function for feature *i*, and b_i is the exponential decay parameter. The audio-scene decision function $D(t_o)$ at any instant t_o is defined as follows: $D(t_0) = \sum_i b_i$.

[7]This metric is intuitive: it is a point-by-point comparison of the two envelopes.

The audio-scene change is detected using the local minima of the decision function. In order to do so, we use a sliding window of length $2w_a+1$ sec. to slide across the data. We then determine if the minima in the window coincides with the center of the window. If it does, the location is labeled as an audio scene change location. The result for a single film is shown figure 3. The figure shows that the results agree within an ambiguity window of w_a sec.

3.3 Determining Video Scenes

In this section, we shall describe the algorithm for video-scene segmentation. The algorithm is based on notions of *recall* and *coherence*. We model the *video-scene* as a contiguous segment of visual data that is chromatically coherent and also possesses similar lighting conditions. A video-scene is said to occur when there is a change in the long-term chromaticity and lighting properties in the video.

In an ideal case, we would like to work with raw frames. However, this would lead to an enormous increase in the computational complexity of the algorithm. Hence, the video stream is converted into a sequence of shots using a simple color and motion based shot boundary detection algorithm [10]. A frame at a fixed time after the location of the cut is extracted and denoted to be the key-frame.

3.3.1 Recall

In our visual memory model, the data is in the form of key-frames of shots (figure 4) 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 [8] the shots prior to that point. The idea of recall between two shots *a* and *b* is formalized as follows:

$$R(a,b) = (1-d(a,b)) \bullet f_a \bullet f_b \bullet (1-\Delta t/T_m), \qquad (2)$$

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

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.

In order to model the continuous nature of the film, we need to introduce the notion of a "shot-let". A shot-let is a fraction of a shot, typically δ sec. in length but could be smaller due to shot boundary conditions. Shot-lets are obtained by breaking individual shots into δ sec. long chunks. 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 δ = 1 sec. works well. Figure 4 shows how shot-lets are constructed. The formula for recall for shot-lets is identical to that for shots. The idea of shot-lets can be shown to significantly improve the detection rate results in [17], [8].

3.3.2 Computing Coherence

Coherence is easily defined using the definition of recall:

$$C(t_o) = \left(\sum_{\forall a \in T_{as},\, b \in \{T_m \setminus T_{as}\}} R(a,b) \right) \Big/ C_{\max}(t_o) \qquad (3)$$

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. This normalization compensates for the different number of shots in the buffer at different instants of time.

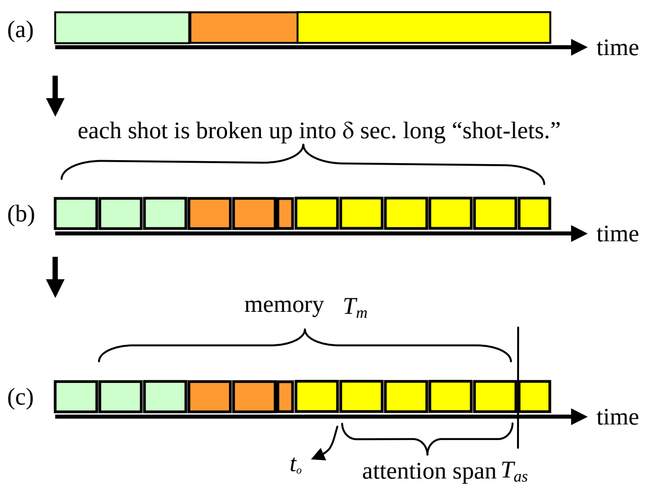

Figure 4: (a) Each solid colored block represents a single shot. (b) each shot is broken up into "shot-lets" each at most δ sec. long. (c) the bracketed shots are present in the memory and the attention span. Note that sometimes, only fractions of shots are present in the memory.

We compute coherence at the boundary between every adjacent pair of shot-lets. Then, similar to the procedure for audio scene detection, we determine the local minima. This we do by using a sliding window of length $2w_v+1$ sec. and determine if the minima in the window coincides with the center of the window. If it does, the location is labeled as a video scene change location.

Shot-lets become necessary in films since films can contain c-scenes with shots that have a long duration. In such cases, if in equation (3), we use shots instead of shot-lets, we will be unable to determine correct minima locations as we will have too few data points. A thought will indicate that interpolating the coherence values will not be of much use. Shot-lets have a smaller time granularity and hence provide us with more reliable minima location estimates.

3.4 Aligning the Detector Results

We generate correspondences between the audio and the video scene boundaries using a simple time-constrained nearest-neighbor algorithm. Let the list of video scene boundaries be V_i $i \in \{1..N_v\}$. Let the list of audio scene boundaries be A_i $i \in \{1..N_a\}$. The ambiguity window around each video scene is w_v sec. long. The ambiguity window width around each audio scene boundary is w_a sec long. Note that these sizes are the same size of the windows used for local minima location. For each video scene boundary, do the following:

- Determine a list of audio scene boundaries whose ambiguity windows intersect the ambiguity window of the current video scene boundary.
- If the intersection is non-null, pick the audio scene boundary closest to the current video scene boundary. Remove this audio scene boundary from the list containing audio scene boundaries.
- If the intersection is null, add the current video scene boundary to the list of singleton (i.e. non-alignable) video scene changes.

At the end of this procedure, if there are audio scene boundaries left, collect them and add them to the list of singleton audio scene changes. Figure 5 illustrates this scenario.

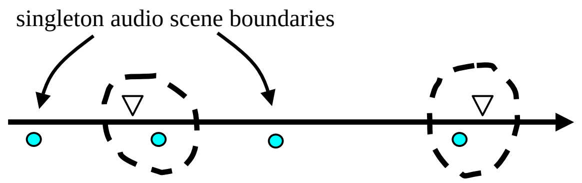

Figure 5: The figure shows video (triangles) and audio (solid circles) scene change locations. The dashed circles show audio/video scene boundaries which align.

Films exhibit interesting interactions between audio and video scene changes. Singleton audio and video scene boundaries can be caused due to the following reasons:

1. **Audio scene change but no video scene change:** this can happen for example : the director wants to indicate a change the mood of the scene, by using a sad / joyous sounding audio track. (b) In *Sense and Sensibility,* one character was shown singing and once she was finished, we had conversation amongst the characters *in the same location.*
2. **Video scene changes within an audio scene:** This happens when a sequence of video scenes have the same underlying semantic . For example, we can have a series of video scenes showing a journey and these scenes will be accompanied by the same audio track.

Now that we have determined the scene boundaries, we will now present algorithms that determine structure within a scene.

4. THE SCENE LEVEL STRUCTURE

In this section we shall discuss possible structures that could exist within a scene and technique to detect and classify such structures. The detection of these structures will help in summarizing the scene. Here, we focus on detection of visual structures.

4.1 Postulating Scene-level Structures

We postulate the existence of two broad category of scenes: N-type (based on the initial definition) and the M-type scene. The N-type scenes are further subdivided into three types: (a) pure dialogue (b) progressive and (c) hybrid. We use an abstract graph representation for representing the shot structure within a scene. Each node in the graph represents one cluster of shots. Figure 6 shows a hybrid scene containing an embedded dialogue.

4.1.1 N-type Scenes

An N-type scene has unity of location, time and sound. We now look at three sub-categories:

Dialogue: A simple repetitive visual structure (amongst shots) can be present if the action in the scene is a dialogue. Note that sometimes, directors will *not* use an alternating sequence to represent a dialogue between two characters. He (or she) may use a single shot of long duration that shows both the characters talking. A repetitive structure is also present when the film-maker shuttles back and forth between two shots (e.g. man watching television). We denote this as a thematic dialogue.

Progressive: There can be a linear progression of visuals without any repetitive structure (the first part of figure 6 is progressive). For example, consider the following scene: Alice enters the room looking for a book. We see the following shots (a) she enters the room (b) she examines her book-shelf (c) looks under the bed (d) locates the book and the camera follows her as she sits on the sofa to read.

Hybrid: This is the most common case, when we have a dialogue embedded in an otherwise progressive scene. For example, in the scene mentioned above, assume Bob enters the room while Alice is searching for the book. They are shown having a brief dialogue that is visualized using an alternating sequence. Then Bob leaves the room and Mary continues her search.

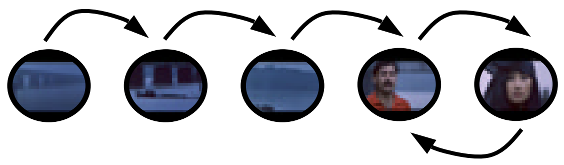

Figure 6: A hybrid scene with an embedded dialogue sequence.

4.1.2 M-type Scenes

In M-type scenes (in classic montage, commercials and MTV videos) we assume there to be no unity of visuals either in terms of location, time or lighting conditions[8]. However, we expect that the audio track will be consistent over the scene. This condition can be converted into a detection rule: A sequence of highly dissimilar shots with unity of sound will be labeled as a M-type scene.

4.2 Determining the Structure

In this section we shall describe techniques to identify structures within N-type scenes. We begin by first describing the discrete object series. Then we show how to use this series in conjunction with statistical tests to determine the presence of a dialogue. Finally we show a simple algorithm that determines the exact location of the dialogue.

[8] There will be a unity of theme which shall be brought about by how the director assembles the component shots of the scene.

4.2.1 *The Discrete Object Series*

The discrete object series (DOS) is a transform that helps us estimate the periodicity in an time-ordered sequence of N objects. It is defined as follows:

$$\Delta(n) \triangleq 1 - \frac{1}{N}\sum_{i=0}^{N-1} d(o_i, o_{\mathrm{mod}(i+n,N)}) , \qquad (4)$$

where, d is the distance function between the objects, mod is the usual modulus function and o_i represents the ith object in the sequence. The modulus function simply creates a periodic extension of the original input sequence. In our case, each object is an image: a key-frame of a shot. Distance between two images is computed using a L^1 color-histogram based distance function.

4.2.2 *Statistical Tests*

We shall use two statistical tests: the students t-test for the means and the F-test for the variances [12]. The F-test is used to determine the appropriate[9] students t-test. These tests are used to compare two series of numbers and determine if the two means and the variance differ significantly.

4.2.3 *Detecting Dialogues*

We can easily detect dialogues using the discrete object series. In a dialogue, every 2nd frame will be very similar while adjacent frames will differ. This is also to observed in figure 7. 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 $\Delta(n)$.
2. Check the DOS to see if $\Delta(2) > \Delta(1)$ and $\Delta(2) > \Delta(3)$.
3. A dialogue is postulated to exist if at least one of two conditions in step 2 is significant at $\alpha = 0.05$ and at most one is significant at $\alpha = 0.1$[10]. Note that $\Delta(n)$ for each n is the mean of N numbers. We use the Student's t-test to determine whether the two means are different in a statistically significant sense.

We use a simple technique to make a distinction between thematic and actual spoken dialogue. From test data we observe that the average shot length for a thematic dialogue is much shorter than for a spoken dialogue. The reason is that there is a minimum time required to utter a meaningful phrase. In [7], the authors assume that phrases last between 5~15 sec. An analysis of hand-labeled data reveals that dialogues with average shot length of less than 4 sec. are thematic.

4.2.4 *The Sliding Window Algorithm*

We use a sliding window algorithm to detect the presence of a dialogue (thematic or true dialogue) within a sequence of frames. Assuming that the total number of frames in the scene to be N, we set the size of the initial window to be *k* frames. Starting with the leftmost 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 immediate right until either a dialogue has been detected or we have reached the end of the scene.
3. If a dialogue has been detected, keep growing the window by adding the key-frame to the immediate right of the current window until either the end of the scene has been reached or the sequence of key-frames in the window is no longer a statistically significant dialogue. The dialogue is the longest statistically significant sequence.
4. Move the start of the new window to the immediate right of the detected dialogue. Go to step 1.

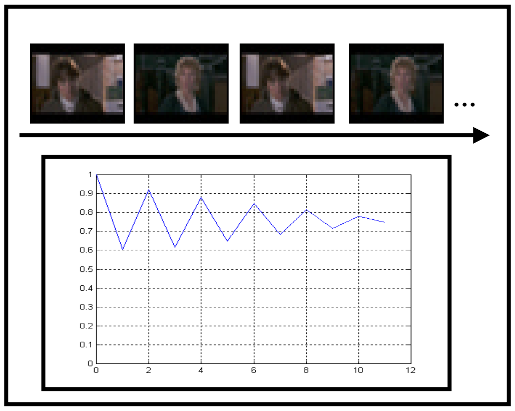

Figure 7: A dialogue scene and its corresponding discrete object series. Note the distinct peaks at n=2, 4 . . .

The initial window size is linearly dependent on the average shot length of the scene; it increases with an increase of the average shot length. This is intuitive since a larger average shot length indicates that we have fewer but longer shots.

The DOS could also be used for creating tests for more elaborate structures (e.g. peaks at n = 2 and n = 5) but it is unclear to us whether these structures exist.

5. 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 five one hour segments from five diverse films. There are three English films: (a) *Sense and Sensibility* (b) *Pulp Fiction* and (c) *Four Weddings and a Funeral*. We also have two foreign language films: (a) *Bombay* (Hindi) and (b) *Farewell my Concubine* (Chinese).

This section is organized as follows. We begin with a section that explains how the labeling of the ground truth data was done. The section following that section deal with the experimental results of the two detectors.

5.1 Labeling the Ground Truth

The audio and the video data were labeled separately. This was because when we use *both* the audio and the video (i.e.

[9] There are two t-tests depending upon whether the variances differ significantly.

[10] We are rejecting the null hypothesis that the two means are equal. We reject the hypothesis if we believe that the observed difference between the means occurred by chance with a probability less than α.

normal viewing of the film) we tend to label scene boundaries based on the semantics of the scene.

The video scene changes were labeled as follows. While watching the video if at any time there was a distinct change in lighting or color, this was labeled as a video scene change. This usually meant a change in location, but walking from a lit room to a dark one was also labeled as a scene change. Scene boundaries for M-type scenes (transient, montage) were very troublesome and difficult to locate.

Table 2: The ground truth data derived from labeling the audio and video data of each film separately. Columns two and three show the number of audio and video scene changes. The last column shows the number of audio/video scene change locations that align.

Film	Audio	Video	Synchronized A/V Changes
Bombay	77	46	33
Farewell my Concubine	91	58	44
Four Weddings and a Funeral	76	57	37
Pulp Fiction	45	39	31
Sense and Sensibility	52	65	41

For audio, we adopted the following policy: label a scene change if it was felt that the ambient audio properties had changed. For example, if we heard the sounds of a marketplace immediately followed a conversation, this was labeled as a scene change. Correctly labeling the audio scene boundaries is challenging since we don't see the associated video. Often, with the beginning and the end of dialogues since there is silence, it becomes very hard to place the boundary accurately[11]. This became particularly arduous when labeling the Chinese film. Since the labeler (the first author) had no semantic understanding of the film, it became hard to determine if a conversation had ended if there was a pause after the last sentence or if the speakers had changed (if one dialogue sequence followed the other).

There are a few interesting observations that one can make from the data in Table 2. The audio and video scene changes in a film, are not random events; i.e. there is a high degree of synergy between the two thus lending support to our computable scene model. In fact, for the first four films the a/v scene changes (last column) are highly correlated with the video scene changes (65%~80%) while for the last film, there is an 80% correlation with the audio scene change locations.

5.2 Scene Change Detector Results

There are three parameters of interest in each scene change algorithm (i.e. audio and video). They are: (a) memory (T_m) (b) attention-span (T_{as}) (c) ambiguity-window size. For both audio and video scene change algorithms, the attention-span and the memory parameters follow intuition: results improve with a large attention-span and a large memory. For both scene change algorithms, large windows have the property of smoothing the audio decision function and the video coherence function. Larger windows decrease the number of false alarms but also increase the number of misses.

Table 3: The table shows c-scene change detector results for the five films. We only deal with adjacent N-type scenes. The columns are: Hits, Misses, False Alarms, Recall and Precision.

Film	H	M	FA	Recall	Precision
Bombay	24	3	9	0.88	0.72
Farewell my Concubine	28	9	10	0.75	0.73
Four Weddings and a Funeral	17	11	4	0.60	0.80
Pulp Fiction	19	9	11	0.67	0.63
Sense and Sensibility	27	7	7	0.79	0.79

The audio and video ambiguity parameters[12] are used in the location of local minima in both scene change algorithms. The same parameters are used as time-constraints when aligning the two scene boundaries. The memory buffer parameters for the entire data set was fixed as follows: audio: T_m=31sec. T_{as}=16sec., video: T_m=16sec., T_{as}=8 sec.

We now present results for the five films in Table 3. These results are for two adjacent N-type transitions only since our algorithms cannot handle N-type $\rightarrow$ M-type or M-type $\rightarrow$ M-type transitions. Note that: recall = hits/(hits + misses) while precision = hits/(hits + false alarms).

The results show that our detector works well, achieving a best result of recall of 0.88 and precision of 0.72 for the film *Bombay*. There are two types of errors that decrease our algorithm performance: uncertainty in the location of the audio labels due to human uncertainty and (b) misses in the video shot boundary detection algorithm. Shot misses cause the shot preceding the missed shot to be deemed longer than its actual length. This affects our coherence formulation as it takes into account the length of shots in the buffer. This, in turn causes video-scene change detection to place the minima at the wrong location.

Prior work done in video scene segmentation used visual features alone [19], [8]. There, the authors focus on detecting scene boundaries for sitcoms (and other TV shows) and do not consider films. However, since we expect the c-scenes in sitcoms to be mostly long, coherent, N-type scenes, we expect our combined audio visual detector to perform very well.

[11] The accuracy in labeling that we refer to is with a comparison to the where the label would have been had we labeled the film with both audio and the video.

[12] This is half the size of the windows used for location of the audio and video minima (w_a and w_v sec. respectively).

5.3 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 ($\alpha = 0.05$). We do need to set two parameters: The initial sliding window size T_w (8 frames) and the threshold for the thematic dialogue test (4 sec.).

The results of the dialog detector show that its performs very well. The best result is precision: 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 unordered. The thematic/true dialog detector's performance is mixed: with a best detection result (precision) of 0.84 for the Indian film *Bombay* and a worst result of 0.50 for *Pulp Fiction*. Thematic dialogues seem to vary significantly with the film genre and the director style; hence a simple time threshold does not seem to suffice.

Table 4:The table shows the dialogue detector results for the five films. The columns are: Hits, Misses, False Alarms, Recall and Precision.

Film	H	M	FA	Recall	Precision
Bombay	10	2	0	0.83	1.00
Farewell my Concubine	10	2	1	0.83	0.90
Four Weddings and a Funeral	16	4	1	0.80	0.94
Pulp Fiction	11	2	2	0.84	0.84
Sense and Sensibility	28	3	0	0.91	1.00

6. DISCUSSING MODEL BREAKDOWNS

In this section we shall discuss three situations that arise in different film-making situations. In each instance, the line-of-interest rule is adhered to and yet our computational scene model breaks down.

1. **Change of scale:** Rapid changes of scale cannot be accounted for in simple model as they show up as change in the chrominance of the shot. For example, the director might show two characters talking in a medium-shot[13]. Then he cuts to a close up. This causes a change in the dominant color of the shots.

2. **Widely differing backgrounds:** This results from the 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.

3. **Background changes with time:** The background can change with time. This can happen for example if the film shows several characters talking in a party (or in a crowd) . Then the stream of people who are moving behind the characters of interest can cause the dominant chrominance/lighting of the shot to change.

These situations cause errors in the video scene segmentation algorithm; either resulting in misses or an incorrect placement of the scene boundary. We have similar problems when M-type scenes abut N-type scenes. Clearly, our computational model makes simplifying assumptions on the possible scenarios when film-makers adhere to the line-of-interest rule.

One possible strategy to deal with these situations is to compute the self-coherence[14] of the shot-lets in the attention span and the self-coherence of the shot-lets in the rest of the buffer. If either self-coherence is low, then we would decide to put an "uncertain" label there and continue ahead. Then, we would group all shot-lets between two "uncertain" labels using the audio-segmentation results.

7. CONCLUSIONS

In this paper we have presented a novel paradigm for film segmentation using audio and video data and visual structure detection within scenes. We developed the notion of computational scenes. The computational model for the c-scenes was derived from film-making rules and experimental observations on the psychology of audition. These scenes exhibit long-term consistency with regard to (a) lighting conditions (b) chromaticity of the scene (c) ambient audio. We believe that the c-scene formulation is the first step towards deciphering the semantics of a scene.

We showed how a causal, finite memory model formed the basis of our scene segmentation algorithm. In order to determine audio scene segments we first determine the correlation amongst the envelope fits for each feature extracted in the memory buffer. We then determine the correlation amongst the envelope fits. The video segmentation algorithm determines the coherence amongst the shot-lets in the memory. The coherence between shot-lets is proportional to the length of the each of the two shot-let as well as incorporates the time difference between the two shot-lets. A local minima criterion determines the scene change points and a nearest neighbor algorithm aligns the scenes.

We introduced the formulation of the discrete object series to determine the periodic structure within a scene. We showed how one can design statistical tests using the Student's t-test to detect the presence of dialogues.

The scene segmentation algorithms were tested on a difficult test data set: five hours from commercial films. They work well, giving a best scene detection result of 88% recall and 72% precision. 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: (a) the data set is complex (b) the audio ground truth labeling was difficult and introduced errors (c) the shot cut detection algorithm had misses that introduced additional error.

There are some clear improvements possible to this work. (a) The computational model for the c-scene is limited, and needs to

[13] 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.

[14] Self-coherence is of a c-scene is determined by computing the coherence of the scene with itself.

tightened in view of the model breakdowns pointed out in section 6. (b) we need to come up with a technique that handles N-type scenes that abut M-type scenes and also the case when M-type scenes are in succession. A possible solution is to introduce a short-term self-coherence function followed by audio-scene based grouping.

8. ACKNOWLEDGEMENTS

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

9. REFERENCES

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

[2] M. Christel et. al. *Evolving Video Skims into Useful Multimedia Abstractions* Proc. of the Conference on Human Factors in Computing System, CHI'98, pp 171-178, Los Angeles, CA, Apr. 1998.

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

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

[5] F. R. Hampel et. al. *Robust Statistics: The Approach Based on Influence Functions,* John Wiley and Sons, 1986.

[6] A. Hauptmann M. Witbrock *Story Segmentation and Detection of Commercials in Broadcast News Video* Advances in Digital Libraries Conference, ADL-98, Santa Barbara, CA., Apr. 22-24, 1998.

[7] Liwei He et. al. *Auto-Summarization of Audio-Video Presentations,* ACM MM ′99, Orlando FL, Nov. 1999.

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

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

[10] J. Meng S.F. Chang, *CVEPS: A Compressed Video Editing and Parsing System*, Proc. ACM Multimedia 1996, Boston, MA, Nov. 1996

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

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

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

[14] Eric Scheirer Malcom Slaney *Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator* Proc. ICASSP ′97, Munich, Germany Apr. 1997.

[15] 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.

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

[17] H. Sundaram S.F Chang *Video Scene Segmentation Using Audio and Video Features,* to appear in IEEE International Conference on Multimedia and Expo, New York, NY, Aug. 2000.

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

[19] 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.

[20] M. Yeung B.L. Yeo *Video Content Characterization and Compaction for Digital Library Applications,* Proc. SPIE ’97, Storage and Retrieval of Image and Video Databases V, San Jose CA, Feb. 1997.

[21] 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.