RC22569 (W0209-092) September 24, 2002
Computer Science

IBM Research Report

Automatic Scene Extraction in Motion Pictures

Ba Tu Truong[*], Svetha Venkatesh[*], Chitra Dorai[**]

[**]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

[*]Department of Computer Science
Curtin University of Technology
GPO Box U1987, Perth 6845, W. Australia

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Automatic Scene Extraction in Motion Pictures

Ba Tu Truong[†], Svetha Venkatesh[†], Chitra Dorai[‡]

Department of Computer Science[†]
Curtin University of Technology
GPO Box U1987, Perth, 6845, W. Australia
{truongbt,svetha}@cs.curtin.edu.au

IBM T. J. Watson Research Center[‡]
P.O. Box 704, Yorktown Heights
New York 10598, USA
dorai@watson.ibm.com

Abstract— **This paper addresses the challenge of bridging the *semantic gap* between the rich meaning users desire when they query to locate and browse media and the shallowness of media descriptions that can be computed in today's content management systems. To facilitate high-level semantics-based content annotation and interpretation, we tackle the problem of automatic decomposition of motion pictures into meaningful story units, namely scenes. Since a scene is a complicated and subjective concept, we first propose guidelines from film production to determine when a scene change occurs. We then investigate different rules and conventions followed as part of Film Grammar that would guide and shape an algorithmic solution for determining a scene. Two different techniques using intershot analysis are proposed as solutions in this paper. Further, we present different refinement mechanisms such as film punctuation detection founded on film grammar to further improve the results. In addition, the refinement techniques demonstrate significant improvements in overall performance.**

I. Introduction

POWERFUL computing infrastructure and high capacity storage devices are spurring digital media archival for reuse, which in turn has stimulated the need for effective content management for instant access, search, and browsing. Content management requires developing tools that enable annotating, indexing, cataloguing, and ultimately understanding unstructured video data. The core task is to structure streams of data and determine their associatedconceptual descriptions in an effective and meaningful manner. The first problem is therefore the identification of the required structure in a media stream and enablement of its automated extraction.

Motion pictures, like other video genres are constructed from shots, which are contiguous sequences of frames taken by a single camera. A large body of literature has been devoted to the problem of shot boundary detection using color, edges, object correlation and motion cues, singly or in combination. While fundamental and essential, shots in film are analogous to letters or words in written language (depending on their length and complexity), they convey little semantic information in isolation. Most meaningful information is embedded at a higher level of film structure, known as scenes or story units, which reflect the aggregation of multiple shots revolving around a single dramatic person, incidence or location. Numerical statistics may be useful here. On an average, a film contains around 2000 shots and 50 scenes, a complexity reduction at a factor of 40, a number that humans can deal without strain. The high-level scene structure reflects the dramatic and narrative structure of the film, as it reflects many film making decisions that are made such as character appearance, background setting, camera placement, sound, music, pace, mood. and so on. Although, the recovery of this level of structure is a challenging problem, it is the first step toward greater semantic understanding of the film, and a crucial endeavor in bridging the semantic gap [1] that beleaguers much of today's content management systems. In addition, segmenting the film into scenes will help create compact film summaries and enable meaningful nonlinear navigation of the film.

Content annotation at a scene level will not only provide the connections between shots that compose the scenes in terms of concepts that appear to unite the shots, but also highlight the progression of changing content in the scene in terms of dynamical aspects of concepts that differ from shot to shot. The MPEG-7 standard [2] provisions for both such higher level descriptions, where, for example, a Video Segment DS can be used to describe a scene.

For the purposes of this paper, the problem of extracting scenes becomes the problem of locating boundaries between scenes. The latter implies that we determine whether a scene boundary occurs across two groups of shots (a group may contain just a single shot) and this definition is followed throughout this paper. Since scenes are composed of many shots, the detection of scene boundaries requires high-level concepts connecting sequences of shots and between shot analysis.

We propose two approaches to extracting scene boundaries using only visual content. The first one is an novel approach based on multi-resolution edge detection on the $\mathcal{H}$, $\mathcal{L}$, and $\mathcal{S}$ shot signals inspired by [3]. The second technique is based on a neighborhood visual coherence measure at each shot boundary. We also propose a new shot similarity measure that takes similarity of colours into account in the formulation of color coherence. Lastly, we exploit Film Grammar to devise ways to further improve the results in four different ways.

The novelty of our work lies in its use of film grammar as the foundational underpinning. First, to establish where scene changes occur, we use the guidelines from film grammar for scene composition. This enables us to set up ground truth that is verifiable and easily replicable. Second, to define and refine the algorithms, we draw insights from rules and techniques that film makers use for scene construction. Lastly, we analyse the errors in the context of film production and this gives us useful insights into the limitations of colour based methods.

The outline of this paper is as follows. In Section 2, we describe previous work. Section 3 discusses how the scene definitions in film literature can be adapted to set up a consistent ground truth. Section 4 explains the concept of Film Grammar for scene composition. Section 5 describes features used in our techniques. Our first edge based approach is described in Section 6. Section 7 presents shot neighborhood coherence method. Section 8 discusses different techniques for further improvement of the results. Experimental results and their analysis are presented in Section 8. Section 9 concludes this paper.

II. Previous work

There are two major trends in using visual information for scene boundary extraction. The first trend groups shots into different clusters and interprets the temporal ordering of elements from each cluster [4], [5], [6]. Techniques from the second trend [7], [6] develop a memory viewer based model which is either causal, finite or non causal infinite to form a shot coherence measure based on shot recall ability and scenes are detected by searching for local minima on this coherence curve.

The problem with techniques from the first trend is that they critically depend upon clustering parameters (threshold, number of clusters), as clustering is the first step in this technique. Clustering also inhibits the ability to visualize the progression across shots which would be useful in locating scene boundaries. Techniques from the second trend requires accurate local minima detection from a rather noisy signal (due to many parameters involved in computation, shot length, spacing, etc) which would be difficult. Further, this model rather draws from the guidance on the subjective perception of viewers, as opposed to being based on how scenes are constructed using cinematic devices by directors.

Observing that a change in audio signal is closely associated with the boundaries of story units, several research groups have recently developed algorithms for detecting scene boundaries that incorporate both audio and visual information [8], [9], [6]. Generally, audio is segmented and often classified into music, speech, silence and environmental sounds. The position and type of audio segments are interpreted in relation to visual changes, to decide whether it strengthens or reduces the possibility of the current shot being at a scene boundary.

Film Grammar has not been systematically exploited in previous work in high level segmentation of video streams. [10] suggests the use of different editing cues to generate meaningful semantic decompositions of streams. [11] proposes a multimodal rule based approach. They first identify local rules about shot transition, shot repetition, editing rhythm; and then construct scenes, or 'macro-segments', by combining rules.

Many high level video segmentation techniques exploit specific characteristics embedded in highly structured video domains such as sports and news casts. [12] and [13] have proposed a multimodal mechanism for segmenting news program into reported news stories that correlates various video, audio, and close-captioned cues to detect when a story segment occurs. Similar work is reported in the Informedia Project [14]. These techniques are too domain oriented to be carried over to films.

In our survey of existing work, we discovered that most of the reported algorithms are not tested on a significant and comprehensive test set. Some are tested using TV programs in which the general definition of film scene would no longer be applicable. Although in this work we only focus on contemporary mainstream color motion pictures, we aim to address the issue by constructing a large data set and show comparative performance results from the two techniques proposed. This is an essential stage before any significant conclusion based on the performance statistics can be inferred. Furthermore, we analyze the errors and examine them in the context of film production techniques, which gives useful insights into the limitation of any color based method.

III. On scene definition

The term 'scene' is borrowed from the French classical theater that had a precise beginning and ending corresponding to the arrival and departure of characters [15]. As nonlinear film editing and complex camera movement opens up opportunities for better modeling of causal and parallel nature of film events, the term, while still being used widely in the film literature, has lost the precision of its meaning. One definition of film scene is given as, "In the strictest sense, a scene is defined as a section of a motion picture which is unified to time and space. It is made up from a series of shots from varying angles and is usually filmed in one session. As a unit of language, the scene is intermediate between a shot and a sequence, being a larger unit than the first and smaller unit than the latter" (The Film Encyclopedia) [16].

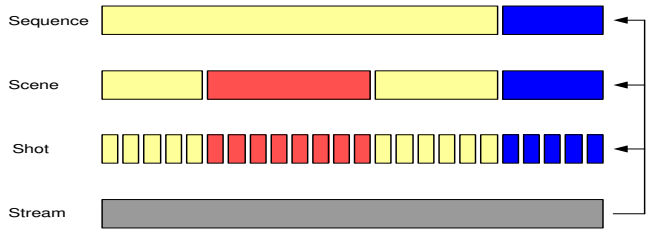

Fig. 1. Syntactic structure of films

As we seek a computational approach to detecting boundaries between scenes, there should be a mechanism to set up ground truth such that it is verifiable, replicable and widely accepted. This issue has not been addressed in the previous work except in [6]. [6] defines computable scenes as containing "long term consistency of chromacity, lighting condition and ambient sound". In our view, it is more appropriate to define scenes from the film maker's viewpoint and study cinematic devices to design an algorithmic solution. We use the following rules as guidelines to set up the ground truth and define scenes in our work:

1. When there are no two interwoven parallel actions, a change in location or time or both defines a scene change (e.g., a mountain shot sequence to a underwater sequence).
2. An establishing shot (e.g., a shot of New York city preceding a dialogue), although maybe different in location to its corresponding scene, is considered part of that scene, as they are unified by dramatic incidence.
3. When parallel actions are present and interleaved, and there is a switch between one action to another, a scene boundary is marked if and only if the duration of the action is shown for at least 30 sec. The reasoning is intuitive. When an action is briefly shown, it serves more as a reminder than representation of any significant event. This also means that the supporting action may never make a scene, while the dominant action would be broken into smaller scene units. For example, during the training scene in The Matrix, a few short shots are inserted to show group members watching through a computer (i.e., a different locale), and these shots should not be considered making up a new scene.
4. A montage sequence involves "dynamic cutting", where shots may containing different spatial properties are rapidly joined to convey a single dramatic event, forming a single scene. For example, in order to convey the desperate effort of Carolyn Burnham to sell a house in The American Beauty, the film maker

joins many different shots of the character showing different customers, different parts of the house. This is also common in dream/flashback sequences (12 Monkeys, Sleepy Hollow)

IV. Film grammar for scene composition

Film Grammar is defined in [17, p.2] as being comprised of a body of 'rules' and conventions that are "the product of experimentation, an accumulation of solutions found by everyday practice of the craft", and results from the fact that films are composed, shaped, and built to convey a certain story. Therefore, understanding Film Grammar is essential to understanding the film data under investigation.

The fundamental assumption in all visual aspect based scene detection algorithms is that shots within a scene exhibit higher visual similarity than those across scenes. This is inherent in the scene definition which generally requires shots in a scene to be homogeneous in space. However, the film maker can easily control the colors to be included/excluded in the scene by changing the camera angle. Similarly, she can minimize or emphasize a particular color in the scene by moving the camera backward for a long shot or forward for a close-up. Therefore, by controlling the the camera alone, the film maker could easily falsify the above assumption about color homogeneity. Fortunately, there are certain rules, or "structural elements" as termed in [18], that film makers use to craft a series of shots into a unified scene, that would ensure that the visual similarity assumption is still valid. The two most common structural elements are *Separation* and *Familar Image* as described by Sharff in his investigation of cinesthetic impact [18]. *Separation* is the arrangement of shots showing subjects individually and interleaving, creating the shot pattern **...ABABAB...**. This contains but is not limited to shot-reverse-shot dialogue scenes. Separation is used widely, since it is capable of creating intimate relationships between characters. In addition, in separation, any action involving motion should take place while the characters are involved in the screen, minimizing abrupt visual changes. For example, a character cannot be in a different corner of a room the next time we see him, even though he had enough time to walk to new position while another character is being shown. The *familiar image* is an element of structure in which a picture reappears in a film with rhythmic frequency with approximately the same composition and framing [18]. It can be distributed throughout the film to create the effect of recall. Within a scene, it functions as a pivot image. Separation and familiar image are among the factors motivating the computation of shot neighborhood coherence discussed later in this paper.

Further, it is well understood that color plays a important role in denoting/connoting a certain character, time, setting or mood [19]. Therefore, transitions between scenes, essentially involving changes in time and space, are often associated with changes in characters and mood, resulting in great visual changes. In addition, to reflect certain moods or themes, film makers may design the color for the scene using a mono-chromatic design, or a limited palette, which emphasizes a single color, varying only in hue or saturation [20]. This technique will essentially reduce the color differences between shots within a scene. However, a note of caution should be made here that when this technique is used in a larger scale or throughout the entire movie, it would reduce the scene boundary detection rate, as the visual distinctions across scenes is reduced.

In addition to reinforcing the visual and dramatic unification of all shots within a scene, film devices in entering/exiting scenes are deliberately placed by the film maker in order to deliver extra meaning to the viewer. These devices are termed as '*film punctuation*' [17]. According to Arijon, separation between sequences, pauses in narration, and stress of a passage in films are achieved by editing, camera movement and object movement, either alone or used in combination [17]. In this section, we only discuss devices that are computable. The popular device for denoting a time transition between scenes is fade out/fade in. A fade is a gradual shot transition in which the image gradually darkens and is replaced with another image which fades in or begins abruptly. White out and color fades are occasionally used as an alternative to ordinary fades. Other gradual transitions such wipes and dissolves can also be used to to connect two sequences isolated in time and/or space. Dark areas can also be used as an alternative to gradual shot transition devices. In this technique, the camera can pan or track from behind a dark area that fills screen and then cuts to the similar opening device. According to Arijon, if all scenes begin abruptly, undue emphasis would be placed on them. It is better to began neutrally and then move on to the main event, which can be achieved by moving the character or the camera. The actor body may block the camera lens and then start to move away disclosing the scene. Alternatively, the camera frames an object in silhouette that blackens the screen completely and tracks to the side to reveal the new scene behind. Since these punctuations denote scene changes, they can detected and merged with the list of scene indices detected by other methods.

We employ these guidelines in devising mechanisms that improve the results of the proposed automatic scene extraction techniques.

V. Feature extraction

This section describes color features used in our work and how these features are efficiently computed for each shot and segments of shots from individual keyframes.

A. HLS Color Space

For this work, HLS color model, rather than RGB, becomes a natural choice, since it better models human perception and is commonly used in art and psychology literature. HLS color model comprises of Hue ($\mathcal{H}$), Lightness ($\mathcal{L}$) and Saturation ($\mathcal{S}$). According to [21], in our objective perception of color we can detect three basic color sensations, which are also referred to as color attributes. These are: (1) hue; (2) saturation, or chroma; and (3) lightness, or value. Hue describes the color itself such as blue, red, green or yellow. Saturation or chroma describes the color richness, the color strength. Since white, gray and black have no chroma, they are called achromatic. Lightness indicates how light or dark the color would appear.

We quantize the HLS space into 12 bins of hue, 5 bins of lightness, and 4 bins of saturation. All colors with the first and last bin of lightness are combined as *black*, and *white* respectively, while all colors with first bin of saturation form 3 different gray levels, depending on their lightness. We have a total of

$113 = 1 + 1 + (5-2) + 12(4-1)(5-2)$ in our final quantized color palette.

B. Temporal normalization scheme

Shots have been long recognized as the fundamental unit for video sequence analysis. In our work, we use algorithms discussed in [22] without dissolve detection, as the number of dissolves detected is small compared to the number of false positives produced by high motion. Feature extraction at the shot level is common in video indexing and retrieval literature. In this section, we describe a simple, yet effective method, for computing color features for shots or sections of many shots. We can model the content of a shot by taking the average of all frames within the shot. However, this method is computationally expensive considering the average duration of a film is around 200000 frames (at 25fps). Alternatively, we can compute the color content of a shot based on its representative frames ($\mathcal{R}$-frame). The simple technique of selecting the first, middle or last frame of a shot as an $\mathcal{R}$-frame may not effectively approximate the content of a shot due to object and camera movement. It is desirable to extract $\mathcal{R}$-frames in a manner such that the number of extracted frames is proportional to the degree of visual change within the shot. The following technique can meet this requirement: Assume $\mathbf{F}_m$, $\mathbf{F}_{m+1}$,...$\mathbf{F}_n$ are $n-m+1$ frames making up a shot $\mathbf{S}$, $\mathcal{R}$-frames $\mathbf{F}_{k_1}$, $\mathbf{F}_{k_2}$, ...,$\mathbf{F}_{k_t}$ are selected as: $k_1 = m$, and for all $1 \leq i < t-1$, $\mathbf{F}_{k_{i+1}}$ is the first frame after $\mathbf{F}_{k_i}$ such that $\mathbb{S}(\mathbf{F}_{k_{i+1}}, \mathbf{F}_{k_i}) > \mathbf{T}$ with $\mathbb{S}(\mathbf{F}_i, \mathbf{F}_j)$ being the histogram difference measure between frame $\mathbf{F}_i$ and $\mathbf{F}_j$. If the shot is static, we only need one $\mathcal{R}$-frame, the first frame of the shot, whilst we require some $\mathcal{R}$-frames in the middle if the shot exhibits a significant level of visual change. Let $\mathbf{S}^l$ denote the length of shot $\mathbf{S}$ and $k_{t+1} = n$. The color feature χ of this shot can be normalized from these $\mathcal{R}$-frames as follows:

$$\mathbf{S}^{\chi} = \frac{\sum_{i=1}^{t} \mathbf{F}_{k_i}^{\chi} . (k_{i+1} - k_i)}{\mathbf{S}^l}$$

where $\mathbf{F}_{k_i}^{\chi}$ is the measurement of frame k_i. In this scheme, each $\mathcal{R}$-frame accounts for the visual content of the shot from its position to the position of the next $\mathcal{R}$-frame. Similarly, the feature χ of a segment $\mathbf{A}$ comprising of consecutive shots $\mathbf{S}_{h+1}$, $\mathbf{S}_{h+2}$, ..., $\mathbf{S}_{h+t}$ can be computed as:

$$\mathbf{A}^{\chi} = \frac{\sum_{i=1}^{t} \mathbf{S}_{h+i}^{\chi} . \mathbf{S}_{h+i}^{l}}{\sum_{i=1}^{t} \mathbf{S}_{h+i}^{l}}$$

Essentially, this scheme weights the color content of a frame or shot by the duration it remains in effect.

The averaging of $\mathcal{H}$ when computed for a shot needs to take into account the circular arrangement of $\mathcal{H}$ values. The vector addition operation appears to be an adequate approximation for this computation. This operation projects $\mathcal{H}$ values on both both the x- and y-axes, computes the averages with respect to these axes individually and finds the resulting average value of $\mathcal{H}$ using the $\arg\tan$ function:

$$\mathbf{F}^{\mathcal{H}} = \frac{360}{2\pi} \arg\tan \left(\frac{\sum \mathbb{H}_i \sin(\mathcal{C}_i^{\mathcal{H}})}{\sum \mathbb{H}_i \cos(\mathcal{C}_i^{\mathcal{H}})} \right)$$

where $\mathcal{C}_i$ is the ith color in our quantized color palette. Since $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$ values are different in domain ranges, they need to be normalized before they can be combined together effectively. We normalize the signals using first and second order moments. For example, the value of $\mathcal{H}$ for a shot i is normalized as.

$$\bar{\mathbf{F}}_i^{\mathcal{H}} = \frac{\mathbf{F}_i^{\mathcal{H}} - \mu^{\mathcal{H}}}{\sigma^{\mathcal{H}}},$$

where $\mu^{\mathcal{H}}$ and $\sigma^{\mathcal{H}}$ are the first and the second moments of $\mathbf{F}^{\mathcal{H}}$ computed for every shot in the movie.

VI. Edge detection based approach to scene extraction

Changes in the overall color atmosphere in a film will result in changes, together or individually, in the $\mathcal{H}$, $\mathcal{L}$, $\mathcal{S}$ histograms of associated shot sequences. These color transitions can be detected using an edge detector such as the one based on Deriche's recursive filtering algorithm using Gaussian kernels [3]. This is a multi-scale edge detector that is parameterized by σ, which determines the slope of target edges in the signal. Larger σ detects smaller slopes of the target edges and vice versa. Two thresholds (τ, and -τ), one negative and one positive, can be applied to the output from this algorithm, known as the *edge signal*, to locate true edges: the higher the threshold, the larger the edges detected, and vice versa. The section above/below these thresholds means a negative/positive change in the original signal. Since there are three different edge signals in our feature set, one for each $\mathcal{H}$, $\mathcal{L}$, and $\mathcal{S}$ signal, they need to be combined. There are two possible ways to do this. In the first method, edges are detected from each of these signals, they are individually thresholded and the output edges are then combined. The second way is to combine edge signals from the three features first before applying thresholds to extract edges. We chose the latter because it allows accumulation of smaller scale changes in $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$. For each σ and shot index i, accumulated edge signal $\mathbb{E}_i^{\mathcal{HLS}}$ is computed from individual edge signals $\mathbb{E}_i^{\mathcal{H}}$, $\mathbb{E}_i^{\mathcal{L}}$ and $\mathbb{E}_i^{\mathcal{S}}$ as follows:

$$\mathbb{E}_i^{\mathcal{HLS}}(\sigma) = \alpha |\mathbb{E}_{\mathbf{i}}^{\mathcal{H}}(\sigma)| + \beta |\mathbb{E}_{\mathbf{i}}^{\mathcal{L}}(\sigma)| + \gamma |\mathbb{E}_{\mathbf{i}}^{\mathcal{S}}(\sigma)|$$

The weighting reflects different degrees of contribution of changes in $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$ to a scene change. For example, if $\mathcal{L}$ changes contribute less to the scene change compared to $\mathcal{H}$, β should receive a smaller value than α. Currently, with the absence of any specific statistical information about their proportional contribution, we set all weights to 1. Then, $\mathbb{E}_i^{\mathcal{HLS}}(\sigma)$ is thresholded to extract edges.

Fig. 2 shows the detected edges on $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$ signals for a segment of movie, Twelve Monkeys. Each bar indicates the temporal extension of the detected edges.

Visually, a shot should be more similar to other shots in the same scene than to shots in a different scene. Therefore, the exact location of a scene transition should maximize average inter-differences between every shot of the previous scene and every shot of the next scene. For a detected edge extending from shot $\mathbf{S}_m$ to $\mathbf{S}_n$, the exact scene transition position $\mathbb{T}(m,n)$, (i.e., the index of the first shot of the next scene) in terms of shot

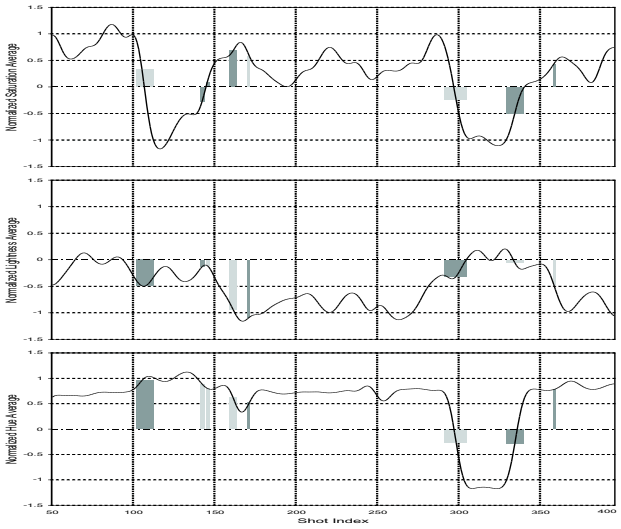

Fig. 2. Detected edges indicating scene tranistions on average $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$.

indices can be determined as:

$$\mathbb{T}(m,n) = \{t \mid \min_{m<t<n} \frac{\sum_{i=m}^{t-1}\sum_{j=t}^{n}\mathbb{S}(\mathbf{S}_i,\mathbf{S}_j)}{(t-m)(n-t+1)}\},$$

where $\mathbb{S}$ is the similarity measure between two shots as defined in the next section.

VII. Shot neighborhood coherence approach

Measuring the visual similarity between shots is the basis of most scene boundary detection techniques that use visual content. However, these techniques have failed to handle at least one of the following issues adequately, and in particular, the first one:

1. The similarity between two colors should be modeled as a continuous, rather than a discrete value. Shots within a scene may exhibit different lighting or shading characteristics, due to different camera shooting angles and overall motion.
2. In fast motion scenes, or slow disclosure shots, only a part of a shot is similar to another shot of the same scene. Therefore, we should use as many frames as needed to evaluate the similarity between shots. This is previously suggested in [23] in terms of the reduced frame set.

We address the first issue by moving away from traditional image comparison using bin-wise histogram comparison, which does not take similarities between colors into consideration. A technique for comparing histograms incorporating the similarity information between colors presented in a quadratic form is proposed in [24]. We propose a new metric, which mimics the process of measuring the similarity between two images by gradually excluding regions with highest similarity. We first form the color similarity matrix $\mathbf{I}$ based on the Euclidean distance between colors in $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$ space. Moreover, we set the distance to **INF** when the $\mathcal{H}$ component of two colors are more than two hue levels apart. This is to confirm the fact that it is neither possible to obtain both blue and red colors for the same object or background region by changing the camera angles nor to have both red/blue shading of the same color. Other values are normalized to [0-1]. Let $\mathbf{F}_i[u]$ denote the bin u of frame $\mathbf{F}_i$, and $\mathbf{I_{ut}}$, the similarity between colors u and t. We define component similarity between two frames $\mathbf{F}_i$,$\mathbf{F}_j$ and two bins u, t as:

$$\mathbb{P}(\mathbf{F}_i,\mathbf{F}_j,u,t) = \mathbf{I_{ut}}\min(\mathbf{F_i}[\mathbf{u}],\mathbf{F_j}[\mathbf{t}])$$

The overall similarity of two frames can be calculated as the recursive sum of individual component similarities:

$$\mathbb{S}(\mathbf{F}_i,\mathbf{F}_j) = \begin{cases} \mathbb{S}(\mathbf{F}_i^*,\mathbf{F}_j^*) + \mathbb{P}(\mathbf{F}_i,\mathbf{F}_j,u_0,t_0) & \exists(u_0,t_0) \\ 0 & \text{otherwise.} \end{cases}$$

with

$$(u_0,t_0) = \{(u,t) \mid (\max_{1\le u,t\le N}\mathbf{I_{ut}}), \mathbb{P}(\mathbf{F_i},\mathbf{F_j},\mathbf{u},\mathbf{t}) > \mathbf{0}\}$$

$$\mathbf{F}_i^*[u] = \begin{cases} \mathbf{F}_i[u] - \min(\mathbf{F}_i[u_0],\mathbf{F}_j[t_0]) & \text{if } u = u_0 \\ \mathbf{F}_i[u] & \text{otherwise.} \end{cases}$$

$$\mathbf{F}_j^*[t] = \begin{cases} \mathbf{F}_j[t] - \min(\mathbf{F}_i[u_0],\mathbf{F}_j[t_0]) & \text{if } t = t_0 \\ \mathbf{F}_j[t] & \text{otherwise.} \end{cases}$$

This means that after taking the component similarity of the two most similar colors, the component similarities of the remaining part of the two histograms are recursively extracted until there does not exist any pair of colors from each histogram that has component similarity greater than zero, i.e., two colors are not similar at all or the bin size of at least one of the colors is zero. The value of $\mathbb{S}$ is then normalized by the total number of pixels of a frame. It should be noted that this measure is essentially the conventional bin-wise intersection metric when $\mathbf{I}$ is the identity matrix, that is:

$$\mathbb{S}(\mathbf{F}_i,\mathbf{F}_j) = \sum_u \min(\mathbf{F}_i[u],\mathbf{F}_j[u]),$$

In order to handle the second issue, we formulate similarity between two shots as the maximum similarity between any pair of $\mathcal{R}$-frames of these shots. In our $\mathcal{R}$-frame representation, the number of frames needed for each shot is proportional to the magnitude of visual changes within the shot. If two shots are static, only their first frames are used. On the other hand, during a chase scene, many frames are used for each shot to adequately cover the changing scene. Thus, $\mathbb{S}$ is given as:

$$\mathbb{S}(\mathbf{S}_i,\mathbf{S}_j) = \max_{\mathbf{F}_{k_{mi}}\in\mathbf{S}_i^{\mathcal{R}},\mathbf{F}_{k_{nj}}\in\mathbf{S}_j^{\mathcal{R}}} \mathbb{S}(\mathbf{F}_{k_{mi}},\mathbf{F}_{k_{nj}})$$

We then define the *shot neighborhood coherence* measure at shot $\mathbf{S}_m$ as the maximum similarity $\mathbb{S}$ between neighborhood shots preceding and succeeding this shot and within a window τ. The low visual variance within a scene as well as the shot repetition (separate, dialogue, familiar images) means that the shots within a scene are highly coherent. On the other hand, we are unlikely to find two similar shots between two scenes, where the coherence between shots across the scene boundaries is typically low. In our implementation, we use both the number of frames and number of shots in defining the temporal window, as it would cater to both long and short shots. The neighborhood coherence at shot m is measured as:

$$\mathbb{C}(\mathbf{S}_m) = \max_{i<m\le j, j<i+\tau} \mathbb{S}(\mathbf{S}_i,\mathbf{S}_j)$$

Missed shot indices present a major problem with this coherence measure. This is illustrated in Fig. 3. Assume a gradual transition is missed between real shot **x** and **y** which results in the shot transition detector claiming shot **b** as the sequence containing these two. Since both scene and shot changes happen across **x** and **y**, these shots are likely to be visually distinctive causing a $\mathcal{R}$-frame of shot **b** fall into the **y** part. This $\mathcal{R}$-frame shares the same color characteristics with the following shots **c** and **d**; therefore, the coherence level at **b-c** is high and no scene change is detected. Similarly, no scene changes are detected between **a-b** either as the first $\mathcal{R}$-frame of shot **b** shares the same visual characteristics with shot **a**. The issue would be resolved by ignoring the immediate preceding shot in computing the shot coherence, that is shot **b** not playing a part in measuring coherence across shots **b** and **c**. A scene boundary, therefore, maybe claimed between shot **b** and **c**. Doing this would increase the noise level of the coherence measure and so we apply a larger threshold. This is reasonable as both dissolves and wipes represent high time and space transitions in scale, and therefore, connected scenes are usually visually distinctive.

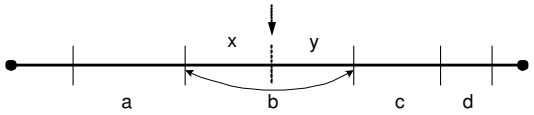

Fig. 3. Missed shot indices.

VIII. Refinement techniques

The identification of edges/thresholding shot coherence signal would return the majority of scene indices. They, however, do not specifically cater for punctuation devices such as those described in Section IV. They also contain many false positives due to fast visual changes across shots due to high camera movement, scene activity or complex film editing. The following subsections will discuss how these false alarms can be eliminated by extending the temporal window, and by using tempo and further color analysis. The data for learning parameters in this section are generated from ground truth of 7 movies (although 2 are not used in the final performance analysis).

A. Detecting film punctuations

We use a technique proposed in [22] to detect fades. This method first detects monochrome frames as significant fade cues. It then examines the luminance means and variances in the neighborhood frames to determine if a real fade is present. We observe that whilst the algorithm described in [22] can detect dissolves adequately in television programs, it is sensitive to high tempo segments in motion pictures and therefore would cause many false positives when compared to the number of correctly detected dissolves. Therefore, the dissolve detection process is omitted. We have not evaluated any wipe detection algorithm, so the wipe detection process is omitted as well.

The detection of other devices such as dark areas is done by searching for all shots that have the mean and variance of the lightness of the first or last frame below a threshold. Apparently, dark scenes contain low lightness as well. Fortunately, dark night scenes are often presented as a mixture of black and vibrant, highly saturated dark colours. For example, deep blues or greens are often used to convey the exterior night atmospheres. Therefore, after discarding the black component, night shots generally have higher average saturation than dark frames inserted for film punctuation purposes. As one of the primary uses of this technique is to create suspense (the viewer is left waiting for something to happen but does not know when [17]), we can set a lower bound on the duration of such shot. If the first or the last frame of a shot is detected as a punctuation frame, the current or the next shot respectively is considered as the start of a new scene.

As evident from the results reported in section IX, these punctuation frames can be reliably detected and are highly indicative of scene changes. The punctuations detected are merged with the list of indices output from the earlier step of the algorithm.

B. Relaxing the temporal window

This technique is inspired from the work described in [23]. In computing shot neighbourhood coherence, the temporal window is kept small, so that all shorter scenes can be detected, especially when they are succeeded and preceded by scenes with similar visual characteristics. This means some actual scenes are broken down into many units resulting in an over-segmentation of the film. For many applications, it is more beneficial to have several units representing an actual scene, and to present all these units, rather than to have one unit representing several scenes, as these scenes cannot be recovered in subsequent analysis [23]. Many falsely detected boundaries using shot neighbourhood coherence can be eliminated by extending the temporal window, thus countering the fixed-window-size limitation of the approach. Shot neighbourhood coherence is computed for each shot at each scene boundary using the extended temporal window. The new temporal window extends to either the next detected scene boundary or 10 shots (depending on which is lesser) on either side of the current shot. If the coherence level is greater than a threshold, that scene boundary is deemed as a false positive. If all scene boundaries are detected in the first step, the relaxed window for each shot at a boundary can extend from the first shot of the last scene to the last shot of the succeeding scene. However, if a scene boundary is missed, it would result in the window being extended too much either in the forward or backward directions. This problem is tackled by imposing a rather large threshold on the maximum length of the extended window. This also reduces the amount of computation required.

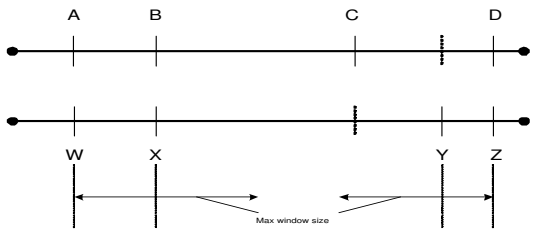

Fig. 4. Temporal window extension

An illustration of how the window extension mechanism is

used is shown in Fig. 4, assuming **A**, **B**, **C** and **D** are real scene boundaries; **W**, **X**, **Y**, and **Z** are boundaries claimed by the proposed algorithm. **W**, **X** and **Z** are correctly detected as **A**, **B** and **D**, while **C** is missed and **Y** is a false alarm. Extended temporal windows are denoted by the sequence at the bottom. When we extend the temporal window of shots around **Y**, it is more likely to find similar shots and **Y** would be eliminated. Note that without applying a large threshold on maximum extension, the window around **X** can be extended to **D** and we may find a shot within **CD** similar to a shot within **AB** and **X** would be incorrectly removed.

C. Tempo analysis

As mentioned earlier, formulations for a pace/tempo function based on shot length and motion are proposed in [3]. The underlying basis for a tempo function is that a film sequence with fast editing and/or high motion tends to be perceived as being of high tempo and vice versa. Fast tempo sequences often involve dynamic visual contents causing low shot neighborhood coherence. Therefore, more false alarms occur during high tempo segments of the film. In addition, it is unlikely that the film maker juxtaposes two scenes of high tempo in succession, especially when their visual contents are not highly distinctive, as the narration may confuse the audience. Therefore, a detected scene boundary would be deemed as a false alarm if sequences preceding and succeeding it have both high tempo and notable visual similarity. This argument is supported by Fig. 5 plotting the tempo similarity vs histogram 'similarity' across sample scene indices. The tempo similarity here is measured as the minimum (intersection) of two tempo values. When color similarity is greater than 0.6, few two successive scenes have tempo similarity greater than 0.1. This information can be used to eliminate these false positives.

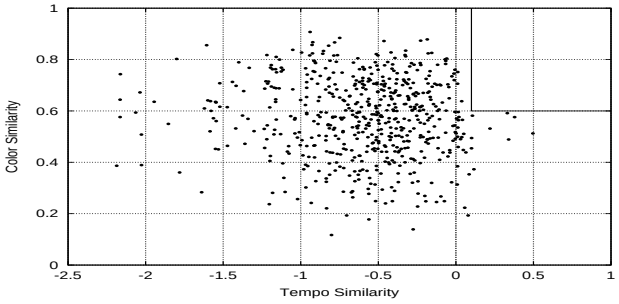

Fig. 5. Tempo Similarity and color similarity across sample scene indices

The initial tempo function proposed in [3] is as follows:

$$\mathbb{P}(n) = \alpha \frac{\mu_l - \mathbf{S}_n^l}{\sigma_l} + \beta \frac{\mathbf{S}_n^m - \mu_m}{\sigma_m},$$

where l refers to shot length in frames, m to motion magnitude and n to shot number. The motion magnitude is calculated for each shot as the aggregation of the absolute value of the sum of the pan and tilt value for consecutive frame pairs of that shot. The 1st and 2nd statistical moments (mean, μ and standard deviation σ) of shot length and motion magnitude are calculated for entire film. We use a simpler version in our implementation, in which the shot length is normalized using the median and the motion was calculated from intensity differences across frames.

D. Analysis of high "impact" colors

Colors can function as indexical signs to denote or connote the character, place and time [19]. It can also be used to add excitement and drama [21]. Scenes such as blue flashing lights of police car and colorful uniforms of a marching band illustrate how colors excite us and/or dramatically intensify an event. Colors can also be used as principal events such as projecting red flashes over entire screen to express a character's intense moments of rage or love. For maximum impact and comprehension, these colors tend to have high 'aesthetic impact', and is sometimes denoted as color energy [21]. They are unlikely to be associated with normal objects or backgrounds that are common from scene to scene. Two segments containing these colors are often semantically associated, and if consecutive, they are often part of the same scene. In fact a *prior* continuous value can be assigned to each color to denote its possibility as a dramatic element in the scene. Fig. 6 shows the unequal distribution of colors across scenes. Each color in our pallete is represented by a circle with the radius representing the average amount of that color across scene indices. Five circles lying on the *x*-axis represent five achromatic colors: black, white and 3 gray scales. The *y*-coordinate indicates the Hue of each color from Red (1) to Purple-Red (12). The *x*-axis indicates three lightness levels correspond to 3 segments (1-3), (4-6) and (7-9). In each segment, saturation increases from left to right. As can be seen from the figure, the most common colors across scenes are black, grey and other earthy tones which are common in our living environment. Blue/Green is also common as it represents the night color. It rarely occurs that the same highly saturated colors are contained in adjacent scenes.

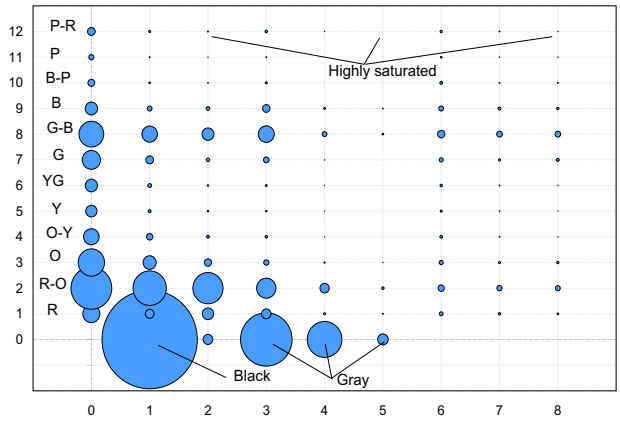

Fig. 6. Average across scene color intersection

The colors mapping to circles with radius smaller than a threshold are filtered to form the list of high 'impact' colors. Let $\tau_{\mathcal{C}}$ denote the maximum occurrence of color $\mathcal{C}$ across a set of scene indices used as training data. Successive scenes are merged if the amount of color $\mathcal{C}$ in both are more than $\alpha\tau_{\mathcal{C}}$. α should be greater than 1 to avoid overfitting the training data. This approach would be further improved if we would develop

a valid probability model of color occurrence across scene indices.

An important note is that this technique needs to be used selectively, since somearthouse movies, animated films may possesses a great deal of highly saturated colors.

IX. Experimental results

As stated previously, we limit our research to contemporary mainstream, color films. This means B&W, early colored and arthouse films are not included in the data set. However, the styles and characteristics of a film are influenced, although not determined, significantly by its genre. The wide selection of movies of different genres would ensure the overall measures of the performance of the algorithm are not biased toward a specific movie kind. Therefore, we set up a data set consisting of 10 full-length movies of all major genres including action (Act), horror (Hrr), science fiction (Scifi), adventure (Adv), thriller (Thrl), fantasy (Fts), family (Fml), drama (Drm), comedy (Cmd) and mystery (Mys). Basic information about each movie is represented in Table I. The genre classification is taken from The Internet Movie Database Web site (IMDB) [1].

TABLE I
Experimental data and ground truth

Movie	Dur(min)	Scenes	Genre
Star Wars I	134	97	Scifi/Adv/Act
The 13th Floor	100	65	Mys/Scifi/Thrl
The Matrix	136	60	Act/Thrl/Scifi
Sleepy Hollow	103	61	Fts/Hrr/Mys
Tall Tale	98	51	Adv/Family
Chameleon	120	46	Scifi/Thrl
12 Monkeys	130	68	Drm/Thrl/Scifi
The Mummy	124	52	Adv/Act/Hrr
American Beauty	121	74	Drm/Cmd
The Siege	115	72	Act/Thrl/Drm
Overall	1181	646	All genres

A. Performance statistics

Table II shows the final results of the edge based approach in terms of number of boundaries correctly detected (Det), number of missed boundaries (Miss), number of false alarms (False), recall rate (Rec) and precision rate (Prec). The corresponding results for shot neighbourhood coherence approach is presented in Table III. A scene boundary index is considered correctly detected if its extension was covered by one of the indices claimed by the algorithm. Recall and precision (measured in percentage) are calculated as 100*Det/(Det+Miss) and 100*Det/(Det+False) respectively.

Overall, the shot neighborhood coherence approach yields better precision and recall (86.3%, 79.2%) compared to the edge based approach (78.7%, 73.9%). This also persists across individual movies except for the recall values in The Matrix, Sleepy Hollow and The Mummy and the precision values of Chameleon, Tall Tale and American Beauty. The reason may

[1] www.imdb.com

TABLE II
Performance of edge based method

Movie	Det	Miss	False	Rec %	Prec %
Star Wars I	77	20	38	79.4	67.0
13th Floor	46	19	11	70.8	80.7
Matrix	54	6	22	90.0	71.1
Sleepy Hollow	51	10	24	83.7	68.0
Tall Tale	40	11	14	78.5	74.1
Chameleon	38	8	8	82.7	82.7
12 Monkeys	54	14	18	79.5	75.0
Mummy	35	17	17	67.4	67.4
American Beauty	56	18	10	75.7	84.9
The Siege	57	15	18	79.2	76.0
Overall	508	138	180	78.7	73.9

TABLE III
Performance of coherence based method

Movie	Det	Miss	False	Rec %	Prec %
Star Wars I	85	12	31	87.7	73.4
The 13th Floor	55	10	11	84.7	83.4
The Matrix	52	8	5	86.7	91.3
Sleepy Hollow	49	12	10	80.4	83.1
Tall Tale	43	8	17	84.4	71.7
Chameleon	43	3	10	93.5	81.2
12 Monkeys	63	5	17	92.7	78.8
The Mummy	33	19	11	63.5	75.0
American Beauty	67	7	16	90.6	80.8
The Siege	67	5	19	93.1	78.0
Overall	557	89	147	86.3	79.2

be that the colors of The Matrix, Sleepy Hollow and The Mummy are tinted towards specific tones (grayish green, grayish light blue, and earthy respectively). This lowers shot coherence across scenes in these movies. The normalization of average $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$ magnifies slight changes in color atmosphere, hence allowing more indices to be retrieved by the edge based approach. On the other hand, Chameleon and American Beauty contain a wide range of colors making the coherence based method more sensitive to false alarms than the edge based method, as the normalization of average $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{S}$ now would suppress small changes in color atmosphere. This also suggests that the coherence based method could be further improved if such normalization processes were incorporated. The highest results for the edge based approach are obtained with Chameleon (82.7%, 82.7%), The Matrix (90.0%, 71.1%) and American Beauty (75.7%, 84.9%). Similarly, the shot coherence based approach performs best on Chameleon (93.5%, 81.2%), The Matrix (86.7%, 91.3%), American Beauty (90.6%, 80.8%) and The Siege (93.1%, 78.0%).

As the shot coherence based approach yields better results, the remaining part of this paper is focused on this technique. The improvements obtained for shot neighbourhood coherence is shown in Table IV for all four refinement methods. Punctuation detection would correctly add to the list of scene indices or

increase false positives, while extending the temporal window, merging high tempo segments and merging high dramatic segments may reduce the number of false positives or increase the number of missed boundaries. As can be seen from this table, the use of punctuation devices are rather common in film practice. Fades and dark frames alone are used at 10.7% (69/646) of scene boundaries. The detection of film punctuation devices are highly accurate. Only 7 false alarms out of 76 claimed devices are found. Two of them are false fades, and the rest are due to the fact that the dark areas did not signal a scene transition, but only a pause in narration. The elimination of false alarms is also highly effective. It eliminates 34% (78/(78+152)) of the original false alarms. The accuracy rate of the elimination techniques is 86.7% (78/(78+12)). On the negative side, the refinement techniques seems to be counter productive when applied to The Mummy which produces 3 false punctuation devices and eliminates 4 true indices. This is due to the fact that a large part of this movie is shot in a cave which is both very dark and earthy. These dark sequences are sensitive to false detection/interpretation of punctuation devices. On the other hand, due to the same dark tones in all scenes in the cave, when the temporal window is extended, the probability of finding two shots from two adjacent scenes with the same visual property increases, resulting in them being incorrectly merged as the same scene.

TABLE IV
IMPROVEMENTS FOR THE COHERENCE BASED METHOD

Movies	Punctuation		False Elimination	
	Correct	False	Correct	False
Star Wars I	4	2	8	1
The 13th Floor	10	1	7	1
The Matrix	9	0	6	0
Sleepy Hollow	7	0	13	2
Tall Tale	9	0	9	0
Chameleon	3	0	8	2
12 Monkeys	6	1	6	1
The Mummy	10	3	4	4
American Beauty	6	0	8	0
The Siege	5	0	9	1
Overall	69	7	78	12

B. Analysis of missed and false scene boundaries

It is useful to analyze the nature of missed scenes and false alarms. This gives an insightful look into the strengths and limitations of the algorithms and suggests further improvements. As it is not practical to analyze all errors individually, broad categorization of false alarms and missed scene boundaries is adopted here.

False alarms are classified into six major types and the statistics for each type is presented in Table V. The first type is due to *multi-angular camera* shooting where the camera wanders around a continuous space, and this is common in crowd scenes or montage sequences. This accounts for more than half of false alarms (79/147) and some may be explained as a change in location at a finer resolution. Multi-angular cameras cause different parts of the space being represented on the screen frame, and therefore reduces the level of visual coherence between shots. As different camera angles lead to differences not only in color histogram, but also edge structures, textures etc, we consider this kind of error as unrecoverable using visual features alone without significant reduction of detection rate. Type II errors are due to *parallel actions* with fast movement and fast interleaving rhythm. These parallel actions are too short to reinforce a scene change according to Rule 3 in our ground truth setting, but the visual property of the scene has significantly altered next time it is shown, resulting in low visual coherence. This type of error in very hard to overcome even by incorporating other features such as audio. Fortunately, they are less harmful in general as they would be considered as correctly detected if the scene definition would be loosened in some way. False alarms of type III are due to the increase/decrease of shot distances leading to a significant chromatic change between shots (eg. a master shot followed by a close-up shot). These errors can be recovered by a better visual similarity measure such as those incorporating template matching, background segmentation, and/or detection of shot distances. One possible way to measure the shot distance is to locate faces and their size in the shot. The fourth type of false alarms is due to the use of multiple establishing shots and/or clear separation of dramatic build-ups and main events. These kinds of false indices are also not very harmful as they carry some significant dramatic information of their own. Type V errors are caused by false detection of punctuation devices and should be identical to that of Table IV. The last error type includes the rest of false alarms but is generally due to the changes of lighting and/or background without any changes in camera angle or distance, etc. This kind of error can be perhaps recovered by a more sophisticated visual similarity measure.

TABLE V
CATEGORIZATION OF FALSE ALARMS

Movies	I	II	III	IV	V	VI
Star Wars I	14	0	9	4	2	2
13th Floor	4	0	0	4	1	2
The Matrix	1	2	1	0	0	1
Sleepy Hollow	5	1	2	2	0	0
Tall Tale	13	1	2	0	0	1
Chameleon	3	0	2	3	0	2
12 Monkeys	11	1	0	1	1	3
The Mummy	6	0	0	2	3	0
American Beauty	9	3	3	1	0	0
The Siege	13	1	2	1	0	2
Overall	79	9	21	18	7	13

Missed scene boundaries are also classified into four major types. The first one is due to the fact that two adjacent scenes do not show any significant visual changes, taking human recognition capability as the upper limit. This type of error is considered as unrecoverable using visual cues without causing a large number of false alarms. Examples of this are scenes in the cave in the Mummy. Type I errors account for the largest portion of missed boundaries (35/89). The second type of error is due to the inadequacy of our similarity measure and/or coherence model. For example, two scenes that are largely visual

distinctive are merged into one, because there are two similar (or deemed as being similar by our measure) shots in these close to each other. In other cases, it may be that the size of our temporal window is too large to detect a short scene enclosed by two visually similar scenes. This type of error is the second largest and would suggest that a better model be developed. The type III errors are due to missed transitions, often gradual ones, which of course can be recovered by the help of reliable shot transition detection. The fourth type of error is a side-effect of Rule 3 in our scene definition. Consider a shot sequence of the form **..XYXYAX ABAB..** with (**A,B**) and (**X,Y**) are two interleaving parallel actions. According to Rule 3 a boundary is not marked between **Y-A** as the next shot after **A** is **X**. On the other hand, a scene index is placed between **X-A**. However, this index is missed as shot coherence between **X-A** is low due to the repetition of **A** shot. Fortunately, this only accounts for 2/89 missed boundaries. Type V errors result from false elimination of initially detected scene indices and should be identical to that in Table IV. Category VI includes the rest of missed boundaries.

TABLE VI
CATEGORIZATION OF MISSED BOUNDARIES

Movies	I	II	III	IV	V	VI
Star Wars I	3	2	4	0	1	2
13th Floor	3	4	0	1	1	1
The Matrix	4	2	2	0	0	0
Sleepy Hollow	4	3	0	1	2	2
Tall Tale	4	4	0	0	0	0
Chameleon	0	1	0	0	2	0
12 Monkeys	3	0	0	0	1	1
The Mummy	8	4	3	0	4	0
American Beauty	3	4	0	0	0	0
The Siege	3	1	0	0	1	0
Overall	35	25	9	2	12	6

Other important statistics regarding errors are that 19.7% (29/147) of false alarms can be considered as correct boundaries if the scene definition would be relaxed to cater for all their variance (for example, a scene change is marked even if the next portion of two interwoven actions is short) and that 7.9 % (7/89) of missed boundaries are removed if the scene definition is tightened (for example, different parts of a cave are considered as a part of a single 'cave' scene)

X. CONCLUSION

In this paper, we have addressed the problem of automatically determining scene boundaries in motion pictures. We investigate different rules and conventions used in Film Grammar that would underpin and shape an algorithmic solution to this problem. We proposed two different techniques. The first is based on an edge detection mechanism on color signals formed by normalized average Hue, Lightness and Saturation computed for each shot. The second approach estimates the coherence level at each shot by computing colour similarity of neighborhood shots. We proposed different mechanisms for further improvement of the results from our scene detector including film punctuation detection, temporal window extension and scene color likelihood models. Final results show that our technique based on the shot sequence coherence performs well and reasonably better than color edges based approach. In addition, the refinement techniques demonstrate significant improvements in overall performance.

REFERENCES

[1] Chitra Dorai and Svetha Venkatesh, "Bridging the semantic gap in content management systems: Computational media aesthetics," in *2001 International Conference on Computational Semiotics in Games and New Media*, September 2001, pp. 94–99.

[2] ISO/IEC/JTC1/SC29/WG11, "Overview of the MPEG-7 standard," http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm, March 2001.

[3] Brett Adams, Svetha Venkatesh, and Chitra Dorai, "Automatic extraction of expressive elements from motion pictures: Tempo," *To appear in IEEE Transaction in Multimedia*, 2002.

[4] A. Hanjalic, R.L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, June 1999.

[5] Yong Rui, Thomas S. Huang, and Mehrotra S., "Constructing table-of-content for videos," *ACM Multimedia System Journal: Special Issue in Multimedia Systems on Video Libraries*, vol. 7, no. 5, pp. 359–368, 1999.

[6] Hari Sundaram and Shih-Fu Chang, "Determining computable scenes in films and their structures using audio visual memory models," in *ACM Multimedia 2000*, Los Angeles, October 2000.

[7] John R. Kender and Boon-Lock Yeo, "Video scene segmentation via continuous video coherence," in *1998 IEEE International Conference on Computer Vision and Pattern Recognition*, 1998.

[8] J. Nam and A. H. Tewfik, "Combined audio and visual streams analysis for video sequence segmentation," in *ICASSP-97*, April 1997, vol. 4, pp. 2665–2668.

[9] C. Saraceno and R. Leonardi, "Identification of story units in audio-visual sequences by joint audio and video processing," in *ICIP98*, 1998.

[10] A. Mfit Ferman and A. Murat Tekalp, "Multiscale content extraction and representation for video indexing," in *Multimedia Storage and Archival Systems II*, Dallas, November 1997, vol. SPIE-3229, pp. 22–31.

[11] P. Aigrain, P. Joly, and V. Longueville, "Medium knowledge-based macro-segmentation of video into sequences," in *Intelligent Multimedia Retrieval*, Mark T. Maybury, Ed., chapter 8, pp. 159–174. AAAI Press/The MIT Press, 1998.

[12] Andrew Merlino, Daryl Morey, and Mark T. Maybury, "Broadcast news navigation using story segmentation," in *ACM Multimedia 97*, 1997, pp. 381–391.

[13] Stanley Boykin and Andrew Merlino, "Machine learning of event segmentation for news on demand," *Communications of the ACM*, vol. 43, no. 2, pp. 35–41, 2000.

[14] Alexander G. Hauptmann and Michael A. Smith, "Text, speech and vision for video segmentation: The informedia project," in *AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision*, 1995.

[15] James Monaco, *How to Read a Film*, Oxford University Press, 1981.

[16] Ephraim Katz, *The Film Encyclopedia*, Harper Collins Publishers, 2nd edition, 1994.

[17] Daniel Arijon, *Grammar of the Film Language*, Silman-James Press, Los Angeles, 1976.

[18] Stefan Sharff, *The elements of Cinema: Towards a cinesthetic impact*, Columbia University Press, New York, 1982.

[19] Gorham Anders Kindem, *Towards a semiotic Theory of Visual Communication in Cinema*, Arno Press, New York, 1980.

[20] David Bordwell and Kristin Thompson, *Film Art: An Introduction*, The McGraw-Hill Companies, Inc., 5th edition, 1997.

[21] H. Zettl, *Sight Sound Motion: Applied Media Aesthetics*, Wadsworth Publishing, 3rd edition, 1999.

[22] Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh, "Improved fade and dissolve detection for reliable video segmentation," in *IEEE International Conference on Image Processing 2000*, Vancouver, 2000, pp. 961–964.

[23] Minerva Yeung, Boon-Lock Yeo, and Bede Liu, "Segmentation of video by clustering and graph analysis," *Computer Vision and Image Understanding*, vol. 7, no. 1, pp. 94–109, July 1998.

[24] J. Hafner, H.S. Sawhney, w. Esquitz, M. Flicker, and W. Niblack, "Efficient colour histogram indexing for quadratic form distance functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 17, pp. 729–736, 1995.