



## Extraction of Film Takes for Cinematic Analysis

BA TU TRUONG  
SVETHA VENKATESH

*Department of Computing Science, Curtin University of Technology, GPO Box U1987, Perth, 6845,  
Western Australia*

truongbt@cs.curtin.edu.au  
svetha@cs.curtin.edu.au

CHITRA DORAI

*IBM T.J. Watson Research Center, P.O. BOX 704, Yorktown Heights, New York 10598, USA*

dorai@watson.ibm.com

**Abstract.** In this paper, we focus on the ‘reverse editing’ problem in movie analysis, i.e., the extraction of film takes, original camera shots that a film editor extracts and arranges to produce a finished scene. The ability to disassemble final scenes and shots into takes is essential for nonlinear browsing, content annotation and the extraction of higher order cinematic constructs from film. A two-part framework for take extraction is proposed. The first part focuses on the filtering out action-driven scenes for which take extraction is not useful. The second part focuses on extracting film takes using agglomerative hierarchical clustering methods along with different similarity metrics and group distances and demonstrates our findings with 10 movies.

**Keywords:** take extraction, film structure, video analysis

### 1. Introduction

Much of the work in content based video indexing and retrieval (CBVIR) has focused on video segmentation including shot/scene extraction and effective algorithms have been reported in this area. In addition, increasingly popular DVD technology has allowed many features, including chapter/scene selection (manually labelled during DVD production) to be incorporated in a DVD release for consumer ease of access to content. The challenge in video analysis has now turned to developing technologies that take advantage of available shot/scene indices for content annotation and better semantic understanding of audio-visual materials to present useful modes to access and manipulate content. In this work, we study the problem of extracting of original *film takes* from produced video and examine the use of clustering techniques to detect film takes automatically.

A film take is defined as “one uninterrupted run of the camera to expose a series of frames,” according to the Dictionary of Film Terms [2]. A film take is also known as a shot captured during the film shooting<sup>1</sup>, and before the editing stage as opposed to shots in the finished film which are generally understood as the portion of the visual stream between two consecutive cut points, or in edited film, splice points. To avoid confusion, this paper always uses the term ‘shot’ in the context of the finished film.

The left side of figure 1 shows the film production process from shooting raw takes to producing the final edited material. During the shooting, different takes of a scene are acquired from multiple camera setups, angles, and/or different filming sections. The editor creates

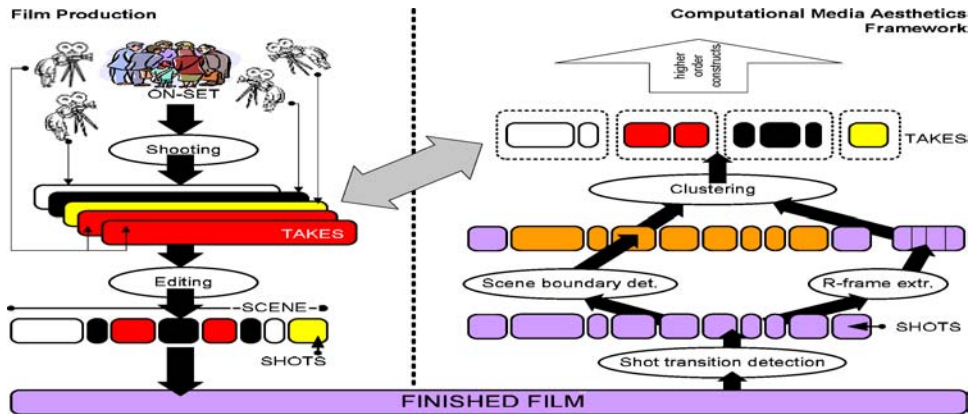


Figure 1. Film production and reverse-editing process.

the final shot sequence of the scene by selecting, mixing, and alternating between different portions of these takes to achieve the desired dramatic intention. Scenes are then assembled into a finished film as seen by viewers. The right side of figure 1 outlines the reverse editing process which uses keyframes/shot/scene indices previously extracted to detect takes that contributed to the final production. It shows, for example, shots of a scene being analyzed and grouped into four clusters which map to five takes captured during the film shooting.

Our survey of CBVIR literature reveals that take extraction problem has never been specifically investigated. The main contribution of this work is in proposing a two-part framework for take extraction. The first part deals with problem of filtering out action-driven scenes, because the extraction of film takes in such scenes is less useful and very difficult. The second part examines a wide range of clustering methods and configurations to identify the best solution for take extraction. Apart from reusing existing techniques, we also devise algorithms to deal with domain specific attributes of film content. The significance of this work is that once take indices are extracted, they can be used for many CBVIR applications ranging from content summarization, navigation and annotation to computing many higher order cinematic constructs in film analysis. We discuss some of these applications in Section 3 of the paper.

The layout of the rest of this paper is as follows. Section 2 reviews previous studies related to our take extraction process. Section 3 describes a wide range of applications for film takes, particularly for extracting higher-order cinematic constructs. The next section overviews essential steps before the clustering process is applied: the use of HLS color space; shot/scene index creation; and keyframe extraction. Section 5 details a technique for filtering out scenes not essential for the take extraction process. The clustering algorithms are described in Section 6. The results are presented in Section 7. Section 8 concludes the paper.

## 2. Previous work

There are many components in our take extraction process, and each component has its own set of related work. In this section, we only review those works related to the last component,

i.e., shot clustering. Good reviews of techniques for shot transition detection/scene boundary detection/key frame extraction can be found in [13, 14, 21, 27].

Clustering of shots for the purpose of content browsing and presentation has been examined in [4, 31]. Zhong et al. [31], uses image features such as color, texture, shape together with temporal features (mean and variance of differences of each frame to the keyframe) and motion features (motion direction histogram, spread distributions) to create a hierarchical view of video content via fuzzy and K-mean clustering techniques. Dimitrova et al. [4], uses a Centroid based clustering technique to cluster keyframes/shots into color superhistograms. The work also outlines some applications of superhistograms including program boundary detection and program classification. However, no performance results are reported in the paper. Recently, we investigated the use of clustering to detect film scenes that are coherent in time/space or mood and present them in a Scene-Cluster Temporal Chart that depicts the alternating themes in a film [23]. Rather than using clustering, content summarization presented in [16] relies on an adaptive and dynamic sampling of the underlying video sequence via the extraction of sub shots and a measure of motion intensity.

Shot clustering/grouping has been often used as an intermediate step in extracting scene boundaries [18, 25, 28, 30]. Hence, these methods do not demand that shots clustered together come from the same take, but from the same scene. They then use overlapping link reasoning to merge separated clusters into scenes. Rui et al. [18] proposes a technique called time-adaptive grouping to create a table-of-content for a video document. They attempt to incorporate other features such as shot length, and shot activity into the measure of similarity between shots. Zhao et al. [30] investigates the use of probabilistic clustering based on best-first model merging to group shots into scenes. Corridoni and Bimbo [3] uses a similarity obtained by integrating a local measure over pixel positions to group similar shots into scenes under the constraint that the sequence has been constructed using shot/reverse-shot technique. Yeung et al. [28] proposes the notion of Scene Transition Graph which organizes clustered shots into a directed graph for compact representation and scene segmentation in a video. Our work alternatively uses scene indices available through other methods (some we have developed) as the temporal constraints in our clustering analysis. Rather than extracting takes, [20] aims at detecting different classes of soccer shots including long shot, in-field medium shot, close-up shot, and out of field shot. Moriyama et al. [15] uses a track-structure based approach, in particular from the point of view of montage components and how they relate to psychological of drama video to provide video summarization.

Clustering has also been used to extract a set of representative frames from the video [5, 6, 8, 9, 17, 32]. These algorithms, as opposed to methods based on shot indices, cluster all video frames regardless of shot boundaries. One frame is then selected from each cluster to create a list of representative frames for the whole video sequence.

The common problem with previous studies is that they tend not to explicitly specify what the extracted clusters represent, other than to describe them in terms of the results obtained (e.g., indoor, coffee shop scenes), and neither do they specify any consistent groundtruth nor measure the system performance on a large set of data. These studies also use clustering in a general manner without investigating the domain specific features of film data such as shot ordering and editing practices.

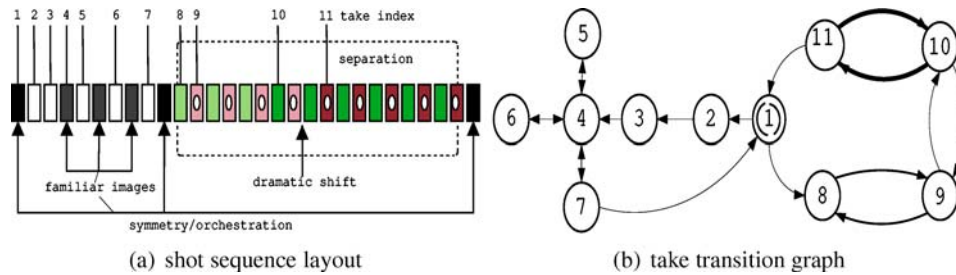


Figure 2. A hypothetical scene.

### 3. Potential utility of film take extraction

This section outlines how automated extraction of film takes via shot clustering plays an essential role in computing many higher order cinematic constructs and tasks in film analysis. For illustration purposes, we use a hypothetical scene. The annotated shot sequence of this scene and its shot/take transition graph are shown in figure 2. In this graph, each take is represented by one node. There is a directed edge from one take to another if there is at least one shot from the first take that precedes another shot from the second take. The numbers denote the take indices, and shots of the same shading belong to the same take. The take transition graph has two added features: the thickness of an edge indicates how often a transition is made from one take to another; the two half circles indicate if the take contains the start shot (I) or the end shot (J) of the scene. Note that the paper will not aim at addressing the realization of these potentialities; the purpose of this section is to call for the attention to take extraction problem by listing its many interesting applications in content summarization and cinematic analysis.

- Content Summarization/Annotation: The identification of film takes will enable more compact representation of the video under analysis. Rather than being overwhelmed by all the shots of the scene, only one shot from each take needs to be presented to the user. The reduction factor for the above scene is 11/29. In real sequences, the reduction would be much higher. It also allows the user to browse the video content in a graph-like structure rather than linearly going through all shots. Many shot features can be annotated for the whole take and these include distance, angle, color, lighting, framing, and composition.
- Shot Flow: We can extract certain shot flow characteristics from their patterns of alternation. Takes 8 and 9, 10, and 11 alternate with each other suggesting a dialogue scene. Take 4 branches to shots of different takes suggesting that it is the centre of action around which other shots revolve. In addition, takes 2, 3, 4, 5, 6, 7 seem to be separated from takes 8, 9, 10, 11.
- Shot Associations: By identifying film takes we have already detected the association between shots. There are also associations between different takes which can be inferred

from shot flows. For example, since there is neither a flow from takes 10 and 8 nor from takes 11 and 9, we can generally deduce that the transition between them would break the flow of the story. Therefore, it is likely that that takes 10 and 8 (11 and 9) shoot the same character using different focal lengths.

- Dramatic Shift: Certain shift in drama/action are reflected in the shift of shot patterns. There is a shift between takes 8/9 to takes 10/11. We can also interpret the kind of dramatic shift within the scene by measuring the shot distance (e.g., via face sizes). If take 8/9 is a medium-shot and take 10/11 is a close-up-shot, we can infer that the drama has probably increased toward the end of the scene.
- Movement Within Scene: Certain aspects of character/camera movements within the scene can also be interpreted. Assume that there is some motion in the first shot of take 8 and the last shot of take 10 and there is no motion in between; it is likely that these two shots involve characters entering/leaving the position of action.
- Relative Difference/Contrast: If a detected take is in a cold tone while another is in a warm tone, we can conclude that there is different state of mind associated with characters in these takes. Likewise, if one contains motion whilst another is static, we can probably deduce that one character is volatile or unsettled while the other is calm.
- Measuring Shot/Take Importance: An essential component of the scene can be measured by how many times the shot is repeated or how long the total duration of all shots of the same take are. Takes 1, 4, 8, 9, 10, 11 seem to contain essential story information while takes 2, 3, 5, 6, 7 are likely to be peripheral. Also, takes 10/11 are likely to be more important than takes 8/9.
- Cinesthetic Elements: Stefan Sharff [19] states that cinema has its own unique method of providing aesthetic gratification and composing cinematic sentences, called ‘cinesthetic elements.’ Four of eight different elements described by Sharff can benefit from the extraction of film takes: separation, familiar image, orchestration, and multi-angularity.
  - Separation: Separation is the fragmentation of a scene into single images, seen in alternation, A, B, A, B, A, B, etc. Separation is a particularly strong element in cinema. In the example, separation starts at the first shot of take 8 and ends just before the last shot. Separation would be detected based on the alternation between takes. In a take transition graph, separation elements are often visible in two-way heavily-connected nodes (Take 8, 9, 10, 11).
  - Familiar Image: This element refers to the repetition of certain images which thus become familiar and are used as the means of keeping together continuity. Familiar images would be detected by looking for takes with at least 3 shots and not alternating with other takes. Takes 1 and 4 are familiar images in the sample scene.
  - Orchestration: This is an open concept and includes symmetry of shot arrangements. Take 1 includes the opening and ending shots and can be seen as an orchestration element.

## 4. Primitive features and temporal segmentation

### 4.1. HLS color space

For this work, the HLS color model, rather than the RGB color model, 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 ( $\mathcal{S}$ ), three basic color sensation in our objective perception of color [29]. 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  $113 = 1 + 1 + (5 - 2) + 12(4 - 1)(5 - 2)$  different colors in our final quantized color palette. This quantization is chosen because the palette produced roughly matches the colors palette used in color studies [12]. The color palette is used to compute the histogram for each keyframe and is the basis for subsequent computation described in Section 6.1.

### 4.2. Temporal segmentation

**4.2.1. Shot boundaries.** The first step in our approach is to extract a list of shot indices from the movies. We used techniques developed previously [22] that detect different types of shot transition effects such as cuts and fades. In this work, we improve conventional cut detection methods using color histogram differences by utilizing an adaptive threshold computed from a local window on the luminance histogram difference curve. Based on the mathematical models for producing ideal fades, different clues (e.g., monochrome frames) for discovering the existence of these effects are proposed, and constraints on the characteristics of frame luminance mean and variance curves are derived analytically to eliminate false positives caused by camera and object motion during gradual transitions. We then use an effective technique for eliminating false positives from a list of detected transitions. Truong et al. [22] also describe a technique for detecting dissolves. However, we chose to skip it as the technique is sensitive to false positives when applied to film data.

**4.2.2. Scene boundaries.** As takes are extracted for individual scenes, a list of scene indices needs to be created. Our previous work in the area of scene boundary extraction [21] provides us with two set of indices:

*Groundtruthed indices:* This set is used as the groundtruth in our scene index extraction work and it reflects the ideal case for non-noisy input data.

*Detected scene indices:* This set is created by one of techniques we propose in [21]. This technique estimates the coherence level at each shot by computing color similarity of neighborhood shots. These coherence values are used to extract a set of raw scene indices.

Different mechanisms are then used for further improvement of the results from our scene detector including film punctuation detection, temporal window extension, color analysis and tempo analysis. This set contains noise and reflects the situation where scene indices are extracted by any automatic method.

We use these two sets of indices to evaluate if the noisy scene detection has a significant impact on take detection results.

#### 4.3. Representative ( $\mathcal{R}$ ) frames

As will be detailed later in the paper, the shot similarity is computed from the similarity of their respective representative frame ( $\mathcal{R}$ -frame)- sets. There are many techniques for extracting these  $\mathcal{R}$ -frames. However, 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  $\mathbf{F}_m, \mathbf{F}_{m+1}, \dots, \mathbf{F}_n$  are  $n - m + 1$  frames making up a shot  $\mathbf{S}$ ,  $\mathcal{R}$ -frames  $\mathbf{F}_{k_1}, \mathbf{F}_{k_2}, \dots, \mathbf{F}_{k_t}$  are selected as:  $k_1 = m$ , and for all  $1 \leq i \leq t - 1$ ,  $\mathbf{F}_{k_{i+1}}$  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  $\mathbf{F}_j$  and  $\mathbf{T}$  being the minimum difference between two frames computed across cut points. 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.

### 5. Scene filtering

Our preliminary study of the problem reveals that it is not always useful and easy to extract film takes, especially for action driven scenes. The emphasis of these scenes is more on events, pace and visual impressions than the repetition of individual shots. Due to substantial object and camera movement, it is very difficult to set up the groundtruth for these scenes. Takes produced for these scenes are prone to errors and would incorrectly depict what is going on in the scene. The use of film takes as outlined earlier is more applicable to drama driven scenes. Therefore, before setting up the groundtruth and applying clustering algorithms over the shot data, it is essential to filter out action driven scenes. The rest are considered as drama driven scenes. It should be noted that our objective here is not to robustly discriminate between action driven and drama driven scenes. Such work requires thorough investigation and a complete feature space including shot distance and audio analysis. The labelling done here is approximate. In this section, we outline an automatic method for filtering out action driven scenes based on the film tempo [1]. This is because 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.

The initial tempo function proposed in [1] that defines the tempo at shot  $\mathbf{S}_i$  as:

$$\mathbb{P}(\mathbf{S}_i) = \alpha \frac{\mu^l - \mathbf{S}_i^l}{\sigma^l} + \beta \frac{\mathbf{S}_i^m - \mu^m}{\sigma^m},$$

where  $\mathbf{S}_i^l$  refers to shot length in frames,  $\mathbf{S}_i^m$  to shot motion magnitude and  $i$  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  $\mu$  and standard deviation  $\sigma$ ) of shot length and motion magnitude are calculated for entire film.

This function is refined in subsequent work using different weighting schemes [1]. We use a simpler version in our implementation, in which the shot length and motion are normalized using the median and the motion was calculated from intensity differences across frames. We note that the use of locally computed means and variances (for individual movie) are appropriate for detecting events and story units as they are influenced by the relative tempo of a film [1]. The concept of action driven and drama driven scenes are rather absolute/global. This also means that it is not necessary to have at least one action driven/drama driven scene for an entire film. Therefore, it is worthwhile to experiment with the motion and shot length statistics computed globally.

We measure the characteristics of a scene  $\mathcal{S}_i$  via two features: its average tempo and the ratio of high tempo shots HTSR. The latter is calculated from the number of shots having tempo value above 0.

$$\mathbb{P}(\mathcal{S}_i) = \frac{\sum_{\mathbf{S}_x \in \mathcal{S}_i} \mathbb{P}(\mathbf{S}_x)}{\|\mathcal{S}_i\|} \quad \text{HTSR}(\mathcal{S}_i) = \frac{\sum_{\mathbf{S}_x \in \mathcal{S}_i} (\mathbb{P}(\mathbf{S}_x) > 0?1 : 0)}{\|\mathcal{S}_i\|}$$

Note that we do not need to smooth the tempo signal as done in [1].

## 6. Algorithms used in take detection

### 6.1. Measuring shot similarity

One of the most popular method for measuring the similarity between two images  $\mathbf{F}_i$  and  $\mathbf{F}_j$  represented by histograms  $\mathbf{H}_i$  and  $\mathbf{H}_j$  is traditional bin-wise intersection:

$$\mathbb{S}(\mathbf{F}_i, \mathbf{F}_j) = \mathbb{S}(\mathbf{H}_i, \mathbf{H}_j) = \frac{\sum_u \min(\mathbf{H}_i[u], \mathbf{H}_j[u])}{w * h},$$

where  $u$  represents the bin index and  $w * h$  is the number of pixels in the image. This measure essentially ignores the spatial distribution of color within the image. A simple method for incorporating the spatial distribution of color is pixel-by-pixel matching:

$$\mathbb{S}(\mathbf{F}_i, \mathbf{F}_j) = \frac{\sum_u (\mathbf{F}_i[u] \approx \mathbf{F}_j[u]?1 : 0)}{w * h},$$



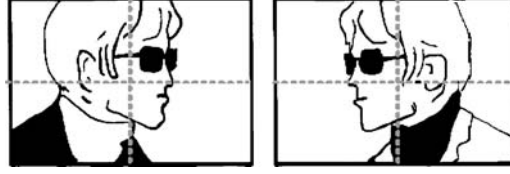


Figure 3. Shot-reverse-shot editing.

where  $u$  is now the pixel index. However, we also need to tolerate variances due to motion and camera adjustments and avoid producing spurious clusters. One method to exploit the advantages of both histogram intersection and pixel-by-pixel matching is to use sub-blocks. We chose to match 4 blocks of the frames separately and combine the results. This essentially addresses the arrangement of colors on the top-left (tl), top-right (tr), bottom-left (bl) and bottom-right (br) parts of the frame. Each block is represented by a color histogram and their similarity is calculated using histogram intersection. The similarity between two frames  $\mathbf{F}_i, \mathbf{F}_j$  can be measured as:

$$\mathbb{S}(\mathbf{F}_i, \mathbf{F}_j) = \frac{\mathbb{S}(\mathbf{H}_i^{\text{tr}}, \mathbf{H}_j^{\text{tr}}) + \mathbb{S}(\mathbf{H}_i^{\text{tl}}, \mathbf{H}_j^{\text{tl}}) + \mathbb{S}(\mathbf{H}_i^{\text{br}}, \mathbf{H}_j^{\text{br}}) + \mathbb{S}(\mathbf{H}_i^{\text{bl}}, \mathbf{H}_j^{\text{bl}})}{4}$$

Figure 3 illustrates the possible advantage of using sub-blocks over global histogram intersection in extracting film takes. A very popular scenario in film editing is one in which two frames/shots from two different takes form a shot-reverse-shot pattern. The global histogram intersection would indicate that two frames/shots are similar (especially in dark and monotone scenes), whereas the sub-block method would pick up their differences.

A shot is represented by a set of key-frames. We need to devise a method for measuring the shot similarity from keyframe similarities. One way to do is to formulate the similarity between two shots as the maximum similarity between any pair of keyframes of these shots:

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

Where  $\mathbf{S}_i^{\mathcal{R}}$  denotes the set of  $\mathcal{R}$ -frames of shot  $\mathbf{S}_i$ .

Alternatively, one may want to experiment with the average version:

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

Using a shot similarity measure we can produce a proximity matrix for each scene in a film. This matrix contains the similarity value for any two shots within the scene. This proximity matrix is used as input for a clustering algorithm to extract film takes.

### 6.2. Fundamental clustering techniques

Due to the fact that two shots of the same take have a strong visual similarity while shots of different takes tend to differ visually, the purpose of this step is to cluster shots into ‘raw’ sets of take indices based purely on shot similarity. There are various agglomerative hierarchical clustering techniques which proceed by producing a series of partitions of data,  $P_n, P_{n-1}, \dots, P_1$  with  $P_i$  consisting of  $i$  clusters. They share the same basic operation as outlined in Algorithm 1 [7].

---

#### Algorithm 1. Hierarchical clustering procedure

---

1. Form  $n$  single-member clusters.
  2. Find the closest pair of distinct clusters, merge them as a new cluster. Delete old clusters and decrement number of clusters by one.
  3. Stop if the number of clusters equals 1, else goto 2
- 

Several methods emerge because of the different ways of measuring the closeness between groups. Some of more popular methods are:

- *Complete linkage (CL)*: This method defines the distance between groups as that of the furthest pair of individuals, one from each group.
- *Group-average linkage (GAL)*: The defining feature this method is that distance between two groups is the average of the distances between all pairs of individuals.
- *Median (MED)*: The distance between two groups in this method is measured as the distance between the centroid of two groups, assuming that they are of equal size, the mean of new group will always be between two the two component groups.
- *Ward’s minimum variance (WARD)*: The aim of this method is to form the partitions in a way that minimizes the losses associated with each grouping. At each step, all possible pairs of clusters are considered and two clusters whose fusion results in the minimum increase in ‘information’ losses are combined. Information loss is defined here as error sum-of-squares criterion.

It should be noted that we perform the clustering procedure on a proximity matrix. An exact recurrence formulas is often used to manipulate this similarity matrix at each clustering step. For **WARD** method which requires the information about the centroid of each cluster, the recurrence is approximate.

### 6.3. Clustering refinements/post processing

General clustering techniques do not take into account specific characteristics of the data domain. Based on the understanding of underlying film production process and film techniques,

we devise algorithms for recursively merging and splitting clusters to further improve the results.

First, we need to deal with *consecutive shots* that are grouped into the same cluster. Other than for some rare ‘staccato’ effects such as those found in ‘Run Lola Run’, it is very unlikely that two consecutive shots are edited from the same take. The grouping of two consecutive shots into a cluster is either due to noisy shot indices or the failure of our similarity metric to discriminate these two shots. Errors of the first kind are rare due to the reliability of shot indexing process. Most errors are of the second type and these clusters need to be spilt. The splitting algorithm is shown in Algorithm 6.3 and it proceeds by choosing two consecutive shots with the least similarity as seeds for two new clusters. The rest are assigned to the closest cluster while maintaining the minimum fusion level.

Secondly, *movements within the scene* may cause shots of one single take to be grouped into at least two clusters. Since the camera follows character movements and actions so as to maintain continuity, the viewer is presented with cues to perceive that the shot sequences in old and new positions are of the same take. However, clustering techniques like **CL** and **WARD** may fail as they measure the distance using all shots in two clusters. The shot with movements is either the last shot of first cluster or the first shot of the second cluster. For the first case, the last shot of a cluster is most similar to shots of the other cluster. Algorithm 3 shows how these movements can be detected to merge the clusters.

---

#### Algorithm 2. Splitting Clusters

---

1. Search all consecutive shots pairs of this cluster.
  2. Select the least similar pair as seeds for two new clusters.
  3. Stop if there are no remaining shots, else select the next shot that have the smallest distance to either of the cluster and assign it to the closer cluster. Repeat 3.
- 

---

#### Algorithm 3. Merging Clusters

---

1. Search all cluster pairs ( $C_1, C_2$ ) satisfying the condition that the last shot of  $C_1$  is 2 shots before the first shot of  $C_2$ , i.e.,  $C_2[1] - C_1[m] = 2$  and  $m + n \geq 4$ . Goto 4.
  2. If  $(\alpha_1 < T$  and  $\alpha_1 > \beta_1)$  or  $(\alpha_2 < T$  and  $\alpha_2 > \beta_2)$ , merge  $C_1$  and  $C_2$
  3. Select the next cluster pair and goto 2, else stop.
- 

A similar situation to movements within the scene is the use of *fluid camera movements* that spans several shots for dramatic impact. For example, a zoom shot is cut to another shot and back to the old shot where the zooming is still on. Due to visible camera movements, those zoom shots are perceived as the same take; however, the differences between two images tend to be larger than the threshold set during the clustering as the zoom continues while the other shot is shown. Algorithm 4 outlines how clusters would be merged in this situation. Currently, fluid camera shots are manually identified to facilitate this step.

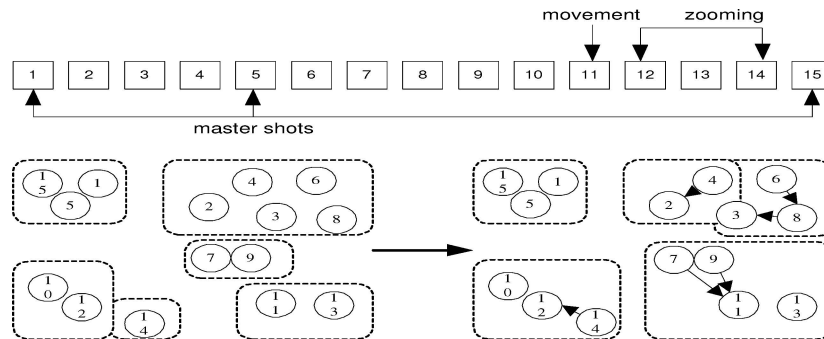


Figure 4. Cluster merging and cluster splitting.

---

**Algorithm 4.** Merging Clusters (Fluid camera movement)

---

1. Search all cluster pairs  $P(C_1, C_2)$  satisfying that the last shot  $C_1$  (size  $m$ ) is 2 shots before the first shot of  $C_2$  (size  $n$ ), i.e.,  $C_2[1] - C_1[m] = 2$  and  $m \geq 1, n \geq 1$ . Goto 3.
  2. If  $C_1[m]$  and  $C_2[1]$  are both classified ‘fluid’ and their difference is not too large merge  $C_1$  and  $C_2$ .
  3. Select the next cluster pair and goto 2, else stop.
- 

Figure 4 shows how these methods have refined 6 raw clusters into 5 final clusters for a hypothetical scene. The cluster  $\{2, 3, 4, 6, 8\}$  is split into 2 clusters  $\{2, 4\}$  and  $\{3, 6, 8\}$  as  $\{2, 3, 4\}$  are three consecutive shots.  $\{2, 3\}$  is assumed to be more similar than  $\{3, 4\}$  and they set up seed points for the new clusters.  $\{4\}$  is placed into the same cluster with 2 as they are highly similar. Likewise, shots  $\{6, 8\}$  are placed into the same cluster as  $\{3\}$ . Two clusters  $\{7, 9\}$  and  $\{11, 13\}$  are merged because shot  $\{11\}$  is the shot where some character movements occur. This shot (the first frame) is similar to shots  $\{7, 9\}$ , although shot  $\{13\}$  may be notably different to them. Shots  $\{12, 14\}$  are labeled with ‘fluid’ camera movements and their similarity is rather high, so they are detected as part of the same take, hence allowing two clusters  $\{10, 12\}$  and  $\{14\}$  to be merged.

#### 6.4. Extracting a partition from the cluster hierarchy

For the purpose of take extraction, we are not interested in the entire cluster hierarchy, but only one partition that is most likely dividing shots into take clusters. In this work, we test four different stopping rules:

1. *C-Index*: The *C*-index is computed as  $[d_w - \min(d_w)] / [\max(d_w) - \min(d_w)]$ , where  $d_w$  is the sum of all  $n_d$  within cluster distances,  $\min(d_w)$  is the sum of the  $n_d$  smallest pairwise distances in the data set, and  $\max(d_w)$  is the sum of  $n_d$  biggest pairwise distances [26].

2. *Mojena*: The Mojena stopping rule is based on the relative sizes of the fusion levels in the dendrogram. In detail, the proposal is to select the number of groups corresponding to the first stage in the dendrogram satisfying:

$$\alpha_{j+1} > \bar{\alpha} + ks_{\alpha}$$

where  $\alpha_0, \alpha_1, \dots, \alpha_n$  are the fusion levels corresponding to stages with  $n, n-1, \dots, 1$  clusters. The term  $\bar{\alpha}$  and  $s_{\alpha}$  are the mean and unbiased standard deviation of the  $\alpha$  value respectively, and  $k$  is a constant [26].

3. *Stepsize*: This simple criterion involves examining the difference in fusion values between hierarchy levels. A large difference would suggest that data was overclustered in the last merger. Thus, the maximum difference was taken as indicator of the optimal number of clusters in the data [26].
4. *Curve-Knee*. This technique involves the detection of the knee, or the point of maximum curvature of the fusion curve. That point is considered as the cut point. Knee detection is done by finding the area between two lines that most closely fit the curve.

#### 6.5. Cluster validation

In order to evaluate how well the clustering algorithms perform, we need to measure the agreement between clusters produced by these algorithms and those set up as the groundtruth. One of the common methods is to use the Rand Index [11]. Let  $U = u_1, u_2, \dots, u_{n_U}$  and  $V = v_1, v_2, \dots, v_{n_V}$  represent the groundtruth and detected clusters respectively. Let  $n$  be the number of elements to be clustered. Let  $a$  be the number of distinct pairs that belong to the same cluster in both  $U$  and  $V$ , and  $d$  be the number of pairs that belong to different clusters in both  $U$  and  $V$ . The Rand index is defined as:

$$\mathbf{R} = \frac{a + d}{\binom{n}{2}}$$

A problem with Rand index is the expected value of the Rand index of two random partitions does not take a constant value and when the cluster size is small it moves toward 1 as the number of cluster increases. Hubert and Arabie [11] proposed the adjusted Rand index. The adjusted Rand index assumes the generalized hypergeometric distribution as the model of randomness, i.e., the  $U$  and  $V$  partitions are picked at random such that the number of objects in the classes and clusters are fixed. Let  $\max(\mathbf{R})$  and  $\mathcal{E}(\mathbf{R})$  are the maximum and expected value of Rand index under this model. The adjusted Rand index is defined as:

$$\mathbf{R}^* = \frac{\mathbf{R} - \mathcal{E}(\mathbf{R})}{\max(\mathbf{R}) - \mathcal{E}(\mathbf{R})}$$

The upper-bound of the adjusted Rand index is 1, and its expected value is 0.

We proposed two new measures namely cluster recall (**CR**) and cluster precision (**CP**). Analogously to recall and precision in information retrieval, **CR** is defined as the ratio of correctly detected pairs to the all possible pairs in the groundtruth, while **CP** is defined as the ratio of correctly detected pairs to the all possible pairs reported:

$$\mathbf{CR} = \frac{a + n}{\sum_{i=1}^{n_U} \binom{\|U_i\|}{2} + n} \quad \mathbf{CP} = \frac{a + n}{\sum_{i=1}^{n_V} \binom{\|V_i\|}{2} + n}$$

$n$  is added, as we include  $n$  non-distinct pairs in the counting of number of pairs that belong the same clusters. This is required to account for the contribution of clusters consisting of single elements to the measure (especially when the cluster size is small). Two partitions agree well when both **CP** and **CR** are high. When two partitions agree perfectly, both **CR** and **CP** are 1. High **CR** and low **CP** imply that small clusters in the groundtruth are grouped into bigger ones in detected clusters. On the other hand, low **CR** and high **CP** imply that clusters in the groundtruth are broken into smaller ones in detected clusters. Thus, **CR** is 1 when there is only one output cluster, while **CP** is 1 when all output clusters contain one single element. The adjusted Rand index can serve as the overall measure of the performance, whilst **CR** and **CP** provide more insight into the nature of clustering outputs.

## 7. Experimental results

### 7.1. Data set and groundtruthing

Currently, we limit this 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 genre would ensure that 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 genre including action (Act), horror (Hrr), science fiction (Scifi), adventure (Adv), thriller (Thrl), fantasy (Fts), family (Fml), drama (Drm), comedy (Cmd) and mystery (Mys). The basic information about each movie is represented in Table 3. The genre classification is taken from The Internet Movie Database Web site (IMDB)<sup>2</sup>.

While groundtruthing, we use the following three guidelines to decide if two shots belong to the same take:

1. Both shots must belong to the same scene.
2. The last frame of the first shot must have similar camera parameters (framing, angle, composition) as the first frame of the second shot.
3. Special case with fluid camera movements: The filmmaker did indeed signal to the viewer that two shots are from the same take through continuous zooming. This is a common technique used in film for time compression and for increasing dramatic impact.

Table 1. Do two shots belong to the same take?





















Case description	Shot A		Shot B		
	first frame	last frame	first frame	last frame	
1. Static shots with the same object and framing					YES
2. Motion shots that maintain the continuity					YES
3. The zoom in the 1st shot is continued in the 2nd shot					YES
4. Shots of the same object with different sizes					NO
5. Shots with different framings that pan to the same framing					NO

Table 1 shows different relations between a shot **A** and a shot **B** and indicates how we decide if two shots belong to the same take during the groundtruthing process. Note that each shot is presented by the model of its first and last frames. Each shape denote objects in the frame and their relative sizes.

The take groundtruths are created for both sets of scene indices described in Section 4.2.2.

Apart from filtering out action driven scenes as described above, we also exclude ‘montage’ scenes without repeated shots from analysis as the ‘best match’ method always returns the perfect results for these scenes. The adjusted Rand Index ( $\mathbf{R}^*$ ), cluster recall ( $\mathbf{CR}$ ) and cluster precision ( $\mathbf{CP}$ ) are used to measure the performance.

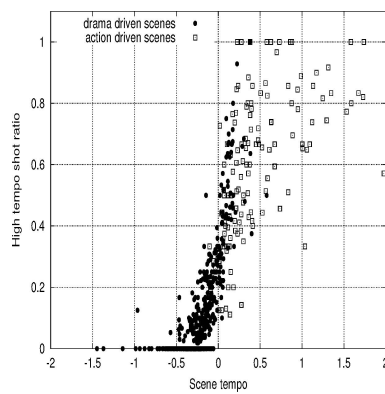
## 7.2. Scene filtering

Using features discussed in Section 5, we build decision trees to classify action driven and drama driven scenes. Table 2 shows the results for different configurations on  $\alpha$ ,  $\beta$  and the scope of  $\{\mu^l, \sigma^l, \mu^m, \sigma^m\}$ . The trees are built from 66% of data set and the reported results are for the whole data set.

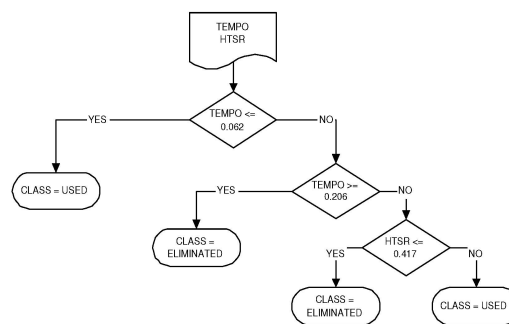
Figure 5(a) plots the tempo against HTSR (computed globally) and figure 5(b) shows the decision tree produced for the best configuration of  $(\alpha = 0.3, \beta = 0.7)$  and using global statistics). These figures indicate that when tempo is very high, the scene is classified as action driven. Conversely, when the tempo is low the scene is classified as drama driven. When the tempo is average, scenes with a high rate of ‘high’ tempo shots are considered as drama driven which are probably scenes involving background motion or moving shots which are consistent throughout the scene. If HTSR is low, it indicates that there is some significant build-up in the action driven scene.

Table 2. Scene filtering results.

		Global parameters		Local parameters	
	$\alpha$	0.3	0.5	0.3	0.5
	$\beta$	0.7	0.5	0.7	0.5
Using only $\mathbb{P}(\mathcal{S}_i)$	Precision	91.7	74.5	69.8	68.1
	Recall	71.6	85.1	73.8	74.5
Using both $\mathbb{P}(\mathcal{S}_i)$ and $\text{HTSR}(\mathcal{S}_i)$	Precision	91.7	74.5	70.6	68.1
	Recall	78.7	85.1	85.1	74.5



(a) tempo vs HTSR



(b) classification tree

Figure 5. Scene filtering.

There are different kinds of errors in filtering out action driven scenes. Most of them involve motion computation. This is understandable since the editing is controlled by the filmmaker and the measure of shot length is always concrete and accurate.

- Some drama driven sequences contain background motion (two people talking in a night club) or moving shots (traveling in the car with a changing background). The focus of these scenes is still on the characters and their interactions.
- The motion in close-up shots tends to be manifested in our computation and this differs largely to the level of motion perceived by the viewer.
- Action driven sequences may contain build-up sections that have slow pace. The average pace is low, but they are marked as being filtered out.
- Movements in dark or toned scenes manifest less in our computation. This results in the lower computed tempo values.

### 7.3. Clustering

First we examine the ability of hierarchical clustering method and similarity measures in identifying a partition of shots into takes. This is done by finding the partition in the



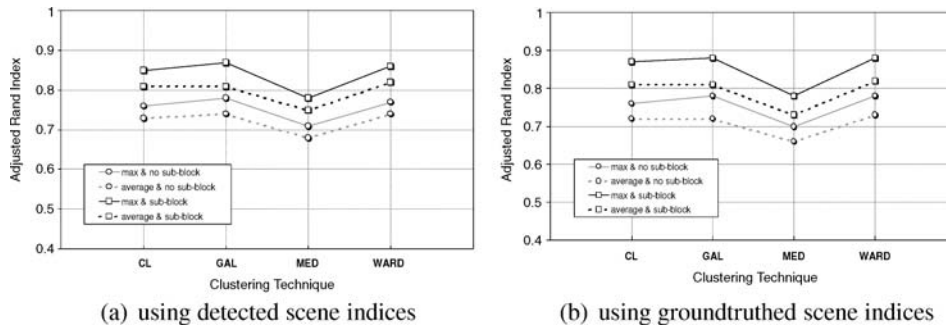


Figure 6. Performance statistics ( $R^*$ ) for different clustering configurations.

hierarchy that ‘best’ match (i.e. score the highest adjusted Rand index) the groundtruth partition.

Figure 6 shows the performance ( $R^*$ ) on all movies for 4 clustering techniques (**CL**, **GAL**, **MED**, **WARD**) and different configurations regarding (a) dividing/not dividing the frame into 4 sub-blocks and (b) calculating the shot similarity as the average/maximum similarity of keyframes. The best results are obtained for **GAL** and **WARD**. It is also evident that the division of the image into sub-blocks and the ‘maximum’ approach offers better performance. Figure 6(a) uses detected scene boundaries, whilst figure 6(b) uses groundtruthed scene boundaries (see Section 4.2.2).

The performance of clustering algorithms on groundtruthed scene index set is slightly better compared to that of detected scene index set. It should be noted however that errors in scene indices does not necessarily mean a decrease in performance. However, false scene indices (i.e., those not in groundtruthed set but detected by a scene boundary detection algorithm) may improve overall clustering performance statistics. This is because applying clustering algorithms on a smaller data subset may be more accurate than on the whole data set.

The performance with the best configuration and **WARD** method for individual movies is shown in Table 3. Lower results are obtained for *The Mummy*, *Sleepy Hollow*, and *12 Monkeys* while better results are obtained *American Beauty*, *Chameleon*, and *Erin Brockovich*.

*American Beauty* and *Erin Brockovich* are two drama films with very few motion sequences. The editing of this film consistently follows the film grammar. Most of shots in the film are static and each scene is also edited from relatively few selective takes. This is to avoid distracting the viewer’s attention from the drama of the story. *Chameleon* is a mysterious/scifi film which contains some action scenes, which are eliminated by our filtering process. The rest of the film is dialogue oriented and shares the same characteristics as *American Beauty* and *Erin Brockovich*. This film is also produced on limited resources, which means only few takes are shot for each scene.

*The Mummy* and *Sleepy Hollow* contain many dark, toned sequences which cause more shots to be merged even though they are not from the same take. These films also contain a

Table 3. Take extraction results.

Movie	Genre	Detected boundaries			Groundtruthed boundaries		
		CP	CR	R*	CP	CR	R*
<i>The 13th Floor</i>	Mys/Scifi/Thrl	0.96	0.94	0.88	0.97	0.94	0.88
<i>The Matrix</i>	Act/Thrl/Scifi	0.95	0.87	0.81	0.97	0.92	0.87
<i>Sleepy Hollow</i>	Fts/Hrr/Mys	0.95	0.90	0.84	0.94	0.90	0.82
<i>Erin Brockovich</i>	Drm	0.97	0.96	0.92	0.99	0.96	0.93
<i>12 Monkeys</i>	Drm/Thrl/Scifi	0.95	0.89	0.75	0.95	0.90	0.81
<i>American Beauty</i>	Drm/Cmd	0.99	0.98	0.96	0.98	0.97	0.94
<i>The Siege</i>	Act/Thrl/Drm	0.95	0.94	0.88	0.93	0.92	0.85
<i>Truman Show</i>	Fts/Cmd/Drm	0.94	0.92	0.83	0.96	0.94	0.89
<i>Chameleon</i>	Scifi/Thrl	0.99	0.96	0.93	0.99	0.97	0.95
<i>The Mummy</i>	Adv/Act/Hrr	0.93	0.89	0.78	0.92	0.88	0.77
Average		0.96	0.92	0.85	0.96	0.93	0.87

significant level of motion throughout the film. *12 Monkeys* is a ‘travelling’ film with many driving sequences.

There are two different kinds of errors in the clustering process. First, a shot is inserted into an incorrect cluster (over-clustering). Second, more shots are grouped into one cluster than necessary (under-clustering).

Errors of the first kind are mainly due to:

- Cross-shots: Shots from two different takes may contain a similar framing at one point during camera movement (e.g., Case 5 in Table 1). The similar framing makes two shots incorrectly grouped into the same cluster.
- The inadequacy of our similarity measure: our similarity measure fails to distinguish between different compositions or dark shots.

Errors of the second kind are mainly due to:

- False shot indices: An incorrect shot index breaks a real shot into two different shots. They are grouped into two different takes during either the clustering or splitting process.
- Lighting/visual inconsistency: For some dramatic impact, the filmmaker may chose to darken a shot making it inconsistent with other shots that also come from the same take. In other cases, the filmmaker actually uses two almost identical physical takes. They are considered as belong to the same ‘perceptive take’. These takes may have inconsistency in lighting due to the fact they are taken in different camera runs or filming sections.
- Camera/object movement: Background motion may completely change the color histogram of an image. Our similarity measure may fail to spot two shots that belong to the same take due to some slight changes in camera angle.

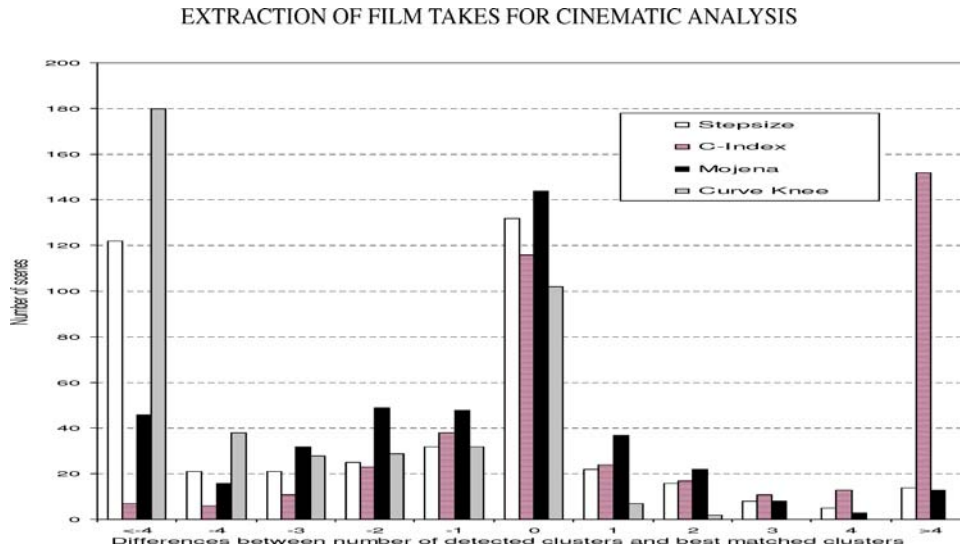


Figure 7. Performance of different stopping rules.

#### 7.4. Extracting partition from hierarchy

The evaluation of the stopping rules is shown in figure 7. The  $x$ -axis indicates the difference between number of extracted clusters (returned by a stopping rule) and the ‘best’ number of clusters (of the partition that best matches the groundtruth). The  $y$ -axis indicates a count of number of scenes. The Mojena rule delivers the best and most stable performance, as it has the highest score when  $x = 0$  and lower variance.  $C$ -index often produces too many clusters while stepsize and curve knee methods tend to produce fewer clusters than expected.

#### 7.5. Demonstration of utilities of film takes

In order to further showcase the application of film take extraction, we have devised a visualization concept called Double-Ring Take-Transition-Diagram (DR-TTD) that is based on STG and can be automatically generated [24]. This visualization allows a quick recognition of subordinate and main takes of a scene depending on their positions on two rings. For two real film scenes in *Erin Brockovich*, 14 and 20 shots can be summarized by 3 and 7 nodes in figures 8(a) and (b) respectively. Dialogue and separation element can be recognized in (1, 2) and (2, 3) from figure 8(a), and (4, 5) and (6, 7) from figure 8(b). For shot association inference described in Section 3, Takes (1, 3) show the same character in the scene. There is a dramatic shift via cut edge (4–6) in Figure 8(b).

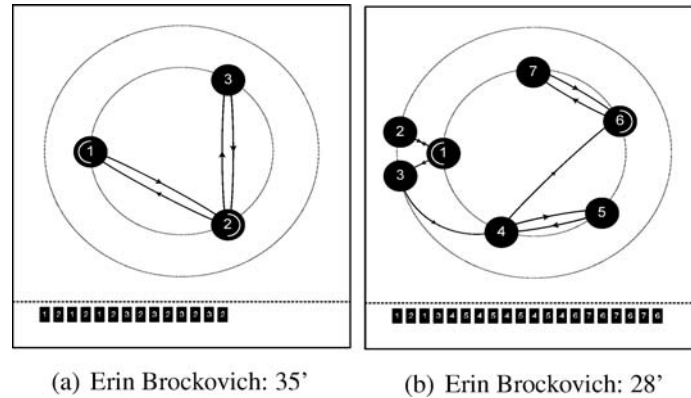


Figure 8. DR-TTD examples.

## 8. Conclusion

We have described our investigation into techniques for extracting film takes, a cinematic element with many useful applications. Action driven scenes are first filtered out, then take extraction for drama driven scenes are investigated by combining traditional hierarchical clustering algorithms with three different post processing methods that handle different aspects of film editing. Our experimental results on 10 movies show the usefulness of dividing the frame into sub-blocks and measuring shot similarity as the maximum of keyframe similarities.

There are other aspects that need further investigation and this includes:

- Investigating in detail the application of film take extraction in content annotation, summarization and semantic extraction as outlined in this paper.
- Incorporating more spatial information into the shot similarity measure. The use of color auto-correlogram would further improve the results, as [10] reports a superior performance of this measure in matching images.
- Visualizing extracted film takes. We are developing techniques for visualizing extracted film takes. The visualization should express as much as possible semantic information that would be indicated by extracted takes and their arrangements.
- Incorporating domain knowledge to construct a better stopping mechanism. This mechanism needs to be based on the condition of the scene such as lightness, motion, hue variance and patterns in the film editing practice.

## Notes

1. During film shooting, a (production) shot is a set of production takes and the notation “Shot X, Take Y” is used to distinguish between them.
2. [www.imdb.com](http://www.imdb.com)

## References

1. B. Adams, C. Dorai, and S. Venkatesh. "Automatic extraction of expressive elements from motion pictures: Tempo," *IEEE Transactions on Multimedia*, Vol. 4, No. 4, pp. 472–481, 2002.
2. F.E. Beaver, *Dictionary of Film Terms: The Aesthetics companion to Film Analysis*, New York: Twayne Publisher, 1994.
3. J.M. Corridoni and A.D. Bimbo, "Structured representation and automatic indexing of movie information content," *Pattern Recognition*, Vol. 31, No. 12, pp. 2027–2045, 1998.
4. N. Dimitrova, J. Martino, L. Agnihotri, and H. Elenbaas, "Color superhistograms for video representation," in *ICIP'99, Kobe, 1999*, Vol. 3, pp. 314–318.
5. N.D. Doulamis, A.D. Doulamis, Y.S. Avrithis, and S.D. Kollias, "Video content representation using optimal extraction of frames and scenes," in *ICIP*, Vol. 1., Chicago, Illinois, 1998, pp. 875–879.
6. M.S. Drew and J. Au, "Video keyframe production by efficient clustering of compressed chromaticity signatures," in *ACM Multimedia 2000*, Los Angeles, 2000.
7. B.S. Everitt, *Cluster Analysis 3rd edition*. Edward Arnold, 1993.
8. D. Farin, W. Effelsberg, and P. de With, "Robust clustering-based video-summarization with integration of domain-knowledge," in *IEEE International Conference on Multimedia and Expo(ICME)*, Lausanne, 2002.
9. A. Girgensohn and J.S. Boreczky, "Time-constrained keyframe selection technique," *Multimedia Tools and Applications*, Vol. 11, No. 3, pp. 347–358, 2000.
10. J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zahib, "Spatial Color Indexing and Applications," *International Journal of Computer Vision*, Vol. 35, No. 3, pp. 245–268, 1999.
11. L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, Vol. 2, pp. 193–218, 1985.
12. S. Kobayashi, *Colorist: A Practical Handbook for Personal and Professional Use*, Kodansha International: Tokyo, 1998.
13. Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," Technical Report HPL-2001-191, HP Laboratory, 2001.
14. R. Lienhart, "Reliable transition detection in videos: A survey and practitioner's guide," *International Journal of Image and Graphics*, Vol. 1, No. 3, pp. 469–486, 2001.
15. T. Moriayama and M. Sakauchi, "Video summarisation based on the psychological content in the track structure," in *ACM multimedia workshops 2000*, 2002, pp. 191–194.
16. J. Nam and A.H. Tewfik, "Dynamic video summarization and visualization," in *The 7th ACM Conference on Multimedia, ACM MM'99*, Vol. 2. Orlando, Florida, 1999, pp. 53–56.
17. C.-W. Ngo, T.-C. Pong, and H.-Z. Zhang, "On Clustering and Retrieval of video shots," in: *ACM Multimedia'01*, Ottawa, 2001, pp. 51–60.
18. Y. Rui, T.S. Huang, and M.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.
19. S. Sharff, "The Elements of Cinema: Towards a Cinesthetic Impact," Columbia Uni Press: New York, 1982.
20. E.A. Tekalp and A. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transaction on Image Processing*, 2003.
21. B.T. Truong, C. Dorai, and S. Venkatesh, "Automatic scene extraction in motion pictures," *IEEE Transactions in Circuits and Systems for Video Technology*, Vol. 13, No. 1, pp. 5–10, 2003.
22. B.T. Truong, S. Venkatesh, and C. Dorai, "New enhancements to cut, fade, and dissolve detection process in video segmentation," in *ACM Multimedia 2000, LA, 2000*, pp. 219–227.
23. B.T. Truong, S. Venkatesh, and C. Dorai, "Application of computational media aesthetics methodology to extracting color semantics in film," in *ACM Multiemdia 2002, France Les Pins, 2002*, pp. 339–342.
24. B.T. Truong, S. Venkatesh, and C. Dorai, "Discovering semantics from the visualization of film takes," in *Accepted for IEEE Multimedia Modelling 2004, Brisbane, Australia, 2004*.
25. E. Veneau, R. Ronfard, and P. Bouthemy, "From video shot clustering to sequence segmentation," in *ICPR'00*, Vol. 4, Barcelona, 2000, pp. 254–257.
26. G.W. Milligan and M.C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, Vol. 50, No. 2, pp. 159–179, 1985.
27. J. Wang and T.-S. Chua, "A cinematic-based framework for detecting scene boundaries in video," *The Visual Computer*. To appear, 2003.

28. M. Yeung, B.-L. Yeo, and B. Liu, "Segmentation of video by clustering and graph analysis," *Computer Vision and Image Understanding*, Vol. 7, No. 1, pp. 94–109, 1998.
29. H. Zettl, *Sight Sound Motion: Applied Media Aesthetics*, 3rd edition, Wadsworth Publishing, 1999.
30. L. Zhao, W. Qi, S. Yang, and H. Zhang, "Video shot grouping using best-first model merging," in *Proc. 13th SPIE Symposium on Electronic Imaging—Storage and Retrieval for Image and Video Databases*, San Jose, 2001, pp. 262–267.
31. D. Zhong, H. Zhang, and S.-F. Chang, "Clustering methods for video browsing and annotation," in *Storage and Retrieval for Still Image and Video Databases IV*, 1996, pp. 239–246.
32. Y. Zhuang, Y. Rui, T. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *ICIP'98*, Chicago, 1998, pp. 866–870.