Scene Segmentation and Categorization Using NCuts

*Yanjun Zhao, Tao Wang, Peng Wang, Wei Hu, Yangzhou Du, Yimin Zhang, *Guangyou Xu
*Dept. CS. &Tech. of Tsinghua University, Beijing, P.R. China 100084
Intel China Research Center, Beijing, P.R. China 100080
zhaoyj@gmail.com, { tao.wang, patricia.p.wang, wei.hu, yangzhou.du, yimin.zhang }@intel.com

Abstract

For video summarization and retrieval, one of the important modules is to group temporal-spatial coherent shots into high-level semantic video clips namely scene segmentation. In this paper, we propose a novel scene segmentation and categorization approach using normalized graph cuts (NCuts). Starting from a set of shots, we first calculate shot similarity from shot key frames. Then by modeling scene segmentation as a graph partition problem where each node is a shot and the weight of edge represents the similarity between two shots, we employ NCuts to find the optimal scene segmentation and automatically decide the optimum scene number by Q function. To discover more useful information from scenes, we analyze the temporal layout patterns of shots, and automatically categorize scenes into two different types, i.e. parallel event scenes and serial event scenes. Extensive experiments are tested on movie, and TV series. The promising results demonstrate that the proposed NCuts based scene segmentation and categorization methods are effective in practice.

1. Introduction

As digital video data becomes more and more pervasive, video summarization and retrieval, i.e., video mining, becomes increasingly important. Similar to text mining based on parsing of word, sentence, paragraph and whole document, video mining can be analyzed from four levels, i.e., frame, shot, scene and the whole video sequence level. In cinematography, scene is the basic story unit that the directors use to convey their ideas. To analyze the video content, scene segmentation is an important module for video mining that clusters temporal-spatial coherent shots into high-level semantic video clips, i.e., scenes.

There is much literature on scene segmentation approaches. Rui [9] proposed a table-of-contents (ToC) method. The method first computes the similarity between shots which is a function of color and activity features of the shots. Shots are then merged together to form groups by a group threshold, and scenes are segmented by merging these groups. Tavanapong [8] extracted visual features from local regions of key frames of shots. Then features are compared by the continuity-editing techniques used in film making. Zhai [10] proposed a scene segmentation method using Markov Chain Monte Carlo (MCMC) algorithm. The scene boundaries are first initialized randomly and then changed using two types of updates: diffusion and jumps. At last, scene boundaries are predicted by the converged result. Rasheed [6] proposed a novel approach by transforming scene segmentation into a graph partitioning problem. This is achieved by constructing a weighted undirected graph called shot similarity graph (SSG), where each node represents a shot and the edges between the shots are weighted by their similarity based on color and motion information. The clustering algorithms in Rui [9] and Tavanapong [8] are based on the local similarity of neighboring shots, which does not perform the best in the global optimization standard. The MCMC method in Zhai [10] and the graph cuts algorithm in Rasheed [6] are both global criterion. But MCMC is much complicated and slower than normalized cuts and has small improvement in our experiments.

Besides scene segmentation, scene categorization and representation is another important research topic for video summarization and retrieval. Rasheed [5] did some work on scene representation but did not refer to scene classification. Huang [3] developed a scene categorization scheme using a Hidden Markov Model (HMM)-based classifier. However, the scene types in [3] are limited to several special videos such as basketball, football, commercial, news and so on, which can not be applied to general videos, e.g. movie. In addition, the training step is complex and needs large annotation data.

Motivated by the good theory of graph partition model [2][6], this paper proposes a novel scene segmentation
and categorization approach using Normalized Graph Cuts (NCuts). Starting from a set of shots, we first calculate shots similarity from their key frames. Then by modeling scene segmentation as a graph partition problem, we employ NCuts to find the optimal scene segmentation. To handle videos with different length and types, we improved the accuracy performance by adaptive parameter setting and shot similarity matrix adjustment. To further discover useful information from scenes, we analyze the temporal layout patterns of shots and automatically categorize scenes into two different types, i.e., parallel scene and serial scene. The scene categorization is very useful for semantic event detection, e.g. a dialog can be easily detected from the parallel scenes with interacting events.

The paper is organized as follows. Our NCuts based scene segmentation and categorization method is presented in Section 2. Extensive experimental results are reported in Section 3, followed by the conclusions in Section 4.

2. Method Details

We view scene segmentation from the perspective of a video program editor. Based on the semantic intention, the editor combined content elements (shots and music) and their temporal layout to express predefined scenarios. Generally, shots of a scene have similar visual contents, which are filmed in a fixed physical setting. Then several transitions from different camera views are incorporated which result in high visual correlations among shots in the same scene. Hence, scene analysis can be viewed as a reverse process of authoring.

Fig. 1 illustrates our system framework. Given an input video, the system first detects shots and measure shot similarities between shots. Then NCuts algorithm clusters temporal-spatial coherent shots into scenes by graph partition. Finally, according to the temporal layout pattern of shots of a scene, the scenes are categorized into different types, i.e., parallel scenes and serial scenes, and representative key frames of each scene are selected.

2.1 Scene Segmentation

The scene segmentation consists of two main modules: shot similarities calculation and normalized cuts to cluster temporal-spatial coherent shots.

2.1.1 Shot Similarities Calculation

Like parsing sentences in a doc, shot detection is the prerequisite step to analyze the video content, and the basic processing unit of scene segmentation. A shot is a set of video frames captured by a single camera in one consecutive recording action. Our system uses a shot detection algorithm [1] to detect shots.

Although motion, audio and local visual features are important auxiliary cues for scene segmentation, they are not robust for complex scenes of movie and home videos. Here, we use a 48 bin RGB color histogram with 16 bins for each channel as the visual feature of a frame. Let \( H_x \) be the normalized color histogram of the \( x \)th frame. The color similarity between frames \( x \) and \( y \) is defined as \( ColSim(x, y) \):

\[
ColSim(x, y) = \sum_{h \in \text{hist}} \min(H_x(h), H_y(h))
\]

Assuming a shot \( S_i \) is a frame set \( S_i = \{f_1, f_2, \ldots, f_j\} \). The shot key frames can be efficiently extracted by the following algorithm [5]:

Step 1: Select middle frame as the first key frame

\[ K'_1 \leftarrow \{f_{\lceil (a+b)/2 \rceil} \} \]

Step 2: for \( j = a \) to \( b \)

Figure 1. Framework of scene segmentation and categorization system.
The goal is to seek the optimal partition \( \text{SSG} \). For scene segmentation, \( j \) \( V \) problem, i.e. graph cut. All shots are represented as a scene segmentation is modeled as a graph partition.

After shot similarity calculation between two shots, \( \frac{\sum_{v_i,v_j} w_{ij}}{\sigma^2} \) is the one that minimize the normalized cut value \( \text{Ncut} \).

For NCuts, the optimal bipartition \( (V_1, V_2) \) of a graph \( V \) is the one that minimize the normalized cut value \( \text{Ncut}(V_1, V_2) \):

\[
\text{Ncut}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\text{assoc}(V_1, V)} + \frac{\text{cut}(V_1, V_2)}{\text{assoc}(V_2, V)}
\]

With
\[
\text{cut}(V_1, V_2) = \sum_{v_i \in V_1, v_j \in V_2} w_{ij},
\]
\[
\text{assoc}(V_1, V) = \sum_{v_i \in V_1, v_j \in V} w_{ij},
\]

where \( w_{ij} \) is the similarity between node \( v_i \) and node \( v_j \). Let \( x \) be a \( |V| \) dimensional indicator vector, \( x_i = 1 \) if node \( i \) is in \( V_1 \) or -1 otherwise. NCut satisfies both the minimization of the disassociation between the sub-graphs and the maximization of the association within each sub-graph. The approximate discrete solution to minimize \( \text{Ncut}(V_1, V_2) \) can be found efficiently by solving the equation as follows:

\[
\min_x \text{Ncut}(x) = \min_y \frac{y^T (D - W) y}{y^T D y}
\]

where \( d_i = \sum_j w(i, j) \), \( D = \text{diag}(d_1, d_2, \ldots, d_N) \).

For \( M \)-cut partitioning, Graph \( G \) could be recursively partitioned into \( M \) parts by \( M \)-1 bipartition operations.

The shot similarity graph \( W(i, j) \) is the important factor in scene segmentation. Since two shots which are near in temporal layout will belong to a scene more possibly than two distant shots, \( W(i, j) \) is proportional not only to the \( \text{ShotSim}(i, j) \) but also to their temporal/frame distance as follows:

\[
W(i, j) = \exp \left( -\frac{1}{d} \left( \frac{m_i - m_j}{\sigma} \right)^2 \right) \times \text{ShotSim}(i, j)
\]

where \( m_i \) and \( m_j \) are the middle frame number shot \( i \) and \( j \) respectively; \( \sigma \) is the standard deviation of shot durations in the entire video; and \( d \) is the temporal decreasing factor. A large value \( d \) would result in higher similarity between two shots even if they are temporally far apart. While smaller value \( d \), shots will be forgotten quickly, thus forming numerous over-segmented scenes. In [6], Rasheed simply chooses \( d \) as a “constant” value, e.g. 20, which will make it work bad for videos with different length and types. We find that \( d \) is related to the shots number \( N \). When the shot number \( N \) is large/small, \( d \) should correspondently increase/decrease to avoid over-segmentation/under-segmentation. Generally, the square of \( d \) is proportional to the shots number \( N \), i.e., \( d \propto \sqrt{N} \). Therefore, we improve Equation (5) by an auto-adaptive value \( d \) as following Eq(6).

\[
W(i, j) = \exp \left( -\frac{1}{N^2} \frac{(m_i - m_j)^2}{\sigma^2} \right) \times \text{ShotSim}(i, j)
\]

To enhance the inner correlation of parallel scene mentioned in next section, we further adjust the shot similarity matrix \( W \) as follows:

\[
\text{if } (w(i, i + d) > 0.9), \quad d \leq 5
\]
\[
\text{then } W(k, l) = W(i, i + d), \quad i \leq k, l \leq i + d
\]

The Eq(7) is useful to avoid that a parallel scene is broken into a few segments in following partitions, i.e., over-segmentation.

The number of partitioning parts \( M \) can be decided through three approaches. The first one is to manually specify \( M \) partitions directly, which is simple but not suitable to variant videos. The second one is to give a maximum threshold of NCut value. Since the NCut
value would increase with recursively partitioning of the graph, it will automatically stop the partitioning when the NCut value is bigger than a given threshold. The scene number \( M \) generally become larger when increasing the shot number \( N \), but the increasing rate is much smaller than that of \( N \). Therefore, we define the NCut value threshold \( T_{cut} \) is proportional to \( N \), i.e.

\[
T_{cut} = a\sqrt{N} + c,
\]

where \( a = 0.02 \) and \( c = 0.3 \) are good parameters in our experiment. The third approach decides the optimal scene number \( M \) by an optimum function. In this paper, we use the \( Q \) function that recently proposed by Newman and Girvan[7] to decide scene number automatically:

\[
Q(P_m) = \sum_{c=1}^{m} \left[ \frac{assoc(V_c, V_c)}{assoc(V, V)} - \left( \frac{assoc(V_c, V)}{assoc(V, V)} \right)^2 \right] (8)
\]

where \( P_m \) is a partition of the shots into \( m \) subgroups/scenes by \( m-1 \) cuts. It is proved that the higher value of the \( Q(P_m) \) function generally corresponds to a better graph partition. Thus, we decide the scene number \( M \) by \( M = \arg \max_m Q(P_m) \).

Fig. 2 shows a scene segmentation result using our normalized cuts approach. The detected scene boundaries matched well with the annotated scene boundaries.

### 2.2 Scene Categorization and Representation

After NCuts based scene segmentation in section 2.1, a video is partitioned into \( M \) video clips, i.e. scenes. To analysis the scenes content, we further categorize scenes in two different types, i.e. parallel scenes and serial scenes, and extract representative key frames of each scene for efficient summarization.

#### 2.2.1 Parallel Scene and Serial Scene

In Webster’s dictionary, a scene is defined as one of the subdivisions of a play in which the setting is fixed, or when it presents continuous action in one place. This definition is good but may not cover all cases which happened in videos. For example, outdoor scene may be shot with moving cameras and variable background. In this paper, we propose an event-based scene definition in table 1, which categorize scenes into two different types: parallel scene and serial scene. The interacting event is an event in which two or more characters interact or characters interact with objects of interest, and in serial event consecutive shots happens without interactions.

Fig. 3 shows the temporal layout patterns of shots of different scenes. Each circle represents one shot, and the same letter in different circles illustrates that these shots are similar. For parallel scene with interacting event (PI scene), such as two actors speaking with each other, there are often two fixed cameras capturing two persons, and then two viewpoints are switched alternately, just like the fig. 3(a). For the Parallel scene with simultaneous serial events (PS scene), video often switches between two serial events as illustrated in fig.

![Figure 3. parallel and serial scenes. (a) Parallel scene with interacting events; (b) Parallel scene with simultaneous serial events; (c) Serial scene.](image)
For serial scene (SS), such as a man traveling from one place to another site, camera setting often keeps changing and shots also changes consecutively as shown in fig. 3(c).

**Table 1. parallel and serial scene definition.**

<table>
<thead>
<tr>
<th>Parallel scene</th>
<th>Serial scene</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Including at least one interacting event (PI), e.g. dialog between two persons.</td>
<td>Including neither interacting events nor serial events happening simultaneously (SS), e.g., a man is driving a car with his girl friend from one city to a mountain.</td>
</tr>
<tr>
<td>2. Including two or more serial events happening simultaneously (PS), e.g. a man is going home on the road while his child is fighting with thieves at home.</td>
<td></td>
</tr>
</tbody>
</table>

2.2.2 Scene Categorization

We make use of shot similarity matrix \( W(i, j) \) to categorize scenes into different types. In Section 2.1.1, \( \text{ShotSim}(i,j) \) has been acquired, which is in the range of [0, 1]. If \( \text{ShotSim}(i,j) > S_t \) (experimentally \( S_t =0.8 \)), Shots \( i \) and \( j \) are generally captured consecutively from a fixed camera view. So they are similar and can be labeled with the same letter but different sequential number, such as A1, A2 ... as shown in Fig. 3 (b). Especially if \( \text{ShotSim}(i,j) > S_h \) (experimentally \( S_h =0.9 \)), there is almost not any change between shot \( i \) and shot \( j \), so they are deemed as the same one and labeled with the same letter such as both A in Fig. 3 (a). The scene categorization algorithm is described as follows.

**Categorize a scene which consists of shots, \text{shot}_i, \ldots, \text{shot}_b**

/* Label shots by their temporal layout */
Label shot \( a \) with letter A
FOR shot \( i = a \) to \( b - 1 \)
    FOR shot \( j = i + 1 \) to \( b \)
        IF \( \text{ShotSim}(i,j) > S_h \), Label shot \( j \) the same letter with shot \( i \).
        ELSE IF \( \text{ShotSim}(i,j) < S_h \), Label shot \( j \) with a new sequential character e.g. B, C, D...
    END
END
WHILE not all shots are labeled
    IF \( S_t < \text{ShotSim}(i,j) < S_h \), Label shot \( j \) with the same letter and different sequential number of shot \( i \).
END

/* Scene Categorization: */
1. Two letters switching regularly: Parallel scene;
2. Two letter groups switching regularly (Group length not exceed \( L = 5 \)): Parallel scene;
3. Shots with same letter and consecutive number: Serial scene
4. Other situations: Serial scene

It is worth pointing out that the Serial Scene and PS scene are variable. If the serial events in PS scene are very long, we may segment them as individual serial scenes. Such situation often exists in films or TV shows. That’s the reason why scene is so subjective and the same video can be segmented into different granularity, i.e., different number of scenes.

By scene categorization, we acquire useful cues for content analysis and semantic event detection. For example, the PI scene with constant appeared faces generally corresponds to the human dialogue as shown in fig 4 (a). We can select the key frames with frequently appeared characters for scene representation.

2.2.3 Scene Representation

Scene representation is to select one or multiple key-frames from its representative shots to represent a scene’s content. Based on shot similarity extracted in section 2.1.1, it is intuitive that a representative shot should have high similarity with other shots and spans a long period of time. Therefore, the shot goodness \( G(i) \) is defined as:

\[
G(i) = C(i)^2 \cdot \text{Length}(i)
\]

(9)

\[
C(i) = \sum_{j \in \text{Scene}} \text{ShotSim}(i,j)
\]

With

The more similar shot \( i \) with other shots \( j \) in the scene, the larger \( C(i) \) and \( G(i) \) are. Furthermore, \( G(i) \) is also proportional to the duration of shot \( i \). For a PI scene, we can simply select key frames from both good shot A and good shot B. For the PS scene, key-frames could be extracted from its sub serial-event group shots.

3. Experiments

To test performance of the proposed approach, we conducted experiments on four videos. The testing videos are one TV series “DaChingJin” (DCJ) and three movies “007 Die Another Day” (007), “The Fifth Element” (FE) and “Pirates of the Caribbean” (PC). For each video, 5 human observers annotated the scene boundaries which are used as the ground truth. The testing videos are listed in Table 2.

**Table 2. Statistic of four testing videos**

<table>
<thead>
<tr>
<th>Testing video</th>
<th>DCJ</th>
<th>007</th>
<th>FE</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>0:45</td>
<td>1:07</td>
<td>1:40</td>
<td>0:30</td>
</tr>
<tr>
<td>Num. of Frames</td>
<td>67004</td>
<td>100451</td>
<td>150960</td>
<td>47778</td>
</tr>
<tr>
<td>Num. of Shots</td>
<td>596</td>
<td>1110</td>
<td>1688</td>
<td>623</td>
</tr>
<tr>
<td>Ground Truth</td>
<td>46</td>
<td>55</td>
<td>81</td>
<td>25</td>
</tr>
</tbody>
</table>
3.1 Scene Segmentation Results

The recall and precision measure are used to evaluate the performance as follows:

\[
\text{Precision} = \frac{\# \text{Matched}}{\# \text{Detected}}, \quad \text{Recall} = \frac{\# \text{Matched}}{\# \text{Ground}}
\]

\[
F_{\text{Score}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]  

where \(\#\text{Detected}\) is the number of total detected scenes; \(\#\text{Ground}\) is the number of total ground truth; \(\#\text{Matched}\) is the number of correct matches between the detected scenes and the ground truth. A detected scene is matched if it has at least 80% overlap with the ground truth.

Table 3 compares our scene segmentation approach with Rasheed method [6] on four testing videos. From the table 3, it can be observed that the average F-score 74.4% of our approach outperforms the performance 63.6% of the Rasheed method [6]. By proposed adaptive \(T_{\text{cut}}\) threshold, temporal decreasing factor \(d\) in Eq(6), and shot similarity matrix adjustment \(w(i, j)\) in Eq(7), our NCuts based scene segmentation approach can automatically segment scenes better for different length/type videos.


<table>
<thead>
<tr>
<th>Testing videos</th>
<th>DCJ</th>
<th>007</th>
<th>FE</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detected Scenes</td>
<td>41</td>
<td>50</td>
<td>76</td>
<td>22</td>
</tr>
<tr>
<td>Recall</td>
<td>72%</td>
<td>64%</td>
<td>68%</td>
<td>80%</td>
</tr>
<tr>
<td>Precision</td>
<td>80%</td>
<td>70%</td>
<td>72%</td>
<td>91%</td>
</tr>
<tr>
<td>F-Score</td>
<td>75.8%</td>
<td>66.9%</td>
<td>69.9%</td>
<td>85.1%</td>
</tr>
<tr>
<td>Average F-Score</td>
<td>74.4%</td>
<td>66.9%</td>
<td>69.9%</td>
<td>85.1%</td>
</tr>
<tr>
<td>Rasheed method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detected Scenes</td>
<td>41</td>
<td>50</td>
<td>76</td>
<td>22</td>
</tr>
<tr>
<td>Recall</td>
<td>70%</td>
<td>58%</td>
<td>65%</td>
<td>68%</td>
</tr>
<tr>
<td>Precision</td>
<td>65%</td>
<td>60%</td>
<td>49%</td>
<td>77%</td>
</tr>
<tr>
<td>F-score</td>
<td>67.4%</td>
<td>59.0%</td>
<td>55.9%</td>
<td>72.2%</td>
</tr>
<tr>
<td>Average F-Score</td>
<td>63.6%</td>
<td>59.0%</td>
<td>55.9%</td>
<td>72.2%</td>
</tr>
</tbody>
</table>

3.2 Scene Categorization and Representation

We tested scene categorization performance on one movie “Pirates of the Caribbean” and one TV series “DaChingJin”. Table 4 lists the scene categorization results. It can be observed that the average F-score 85.9% of serial scene detection performs better than the one 80.7% of PI scene and the one 70.2% of PS scene since serial scenes are easier to be accurately detected than complex parallel scenes. The total a) Examples of detected parallel scenes. The upper one is a parallel scene with interacting dialog events in “DaChingJin”, and the lower one is a Parallel scene with two simultaneous serial events in “The Fifth Element”.

b) Examples of detected serial scenes. The upper one is a serial scene in “DaChingJin”, and the lower one is a serial scene in “Pirates of the Caribbean”.

Figure 4. Examples of scene categorization and representation. Thumbnails are the main key frames of each shot. The thumbnails with red rectangles are detected scene key-frames for scene representation.
average F-score is 79%. Figure 4 shows examples of detected parallel scenes and serial scenes. The thumbnail images are main key frames of each shot and the thumbnails with red rectangles are detected scene key-frames for scene representation. From Fig 4(a), it can be observed that two key frames are well selected to describe the main characters or simultaneous events in parallel scenes. In Fig 4(b), the frequently appeared (or long duration) key frames are selected to represent the serial scenes.

**Table 4 Scene categorization results**

<table>
<thead>
<tr>
<th>Testing videos</th>
<th>DCJ</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI Scenes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ground Truth</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>Detected Scenes</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Recall</td>
<td>76%</td>
<td>75%</td>
</tr>
<tr>
<td>Precision</td>
<td>100%</td>
<td>75%</td>
</tr>
<tr>
<td>F-Score</td>
<td>86.3%</td>
<td>75%</td>
</tr>
<tr>
<td>Average F-Score</td>
<td>80.7%</td>
<td></td>
</tr>
<tr>
<td>PS Scenes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ground Truth</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Detected Scenes</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Recall</td>
<td>71.4%</td>
<td>66.7%</td>
</tr>
<tr>
<td>Precision</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>F-Score</td>
<td>83.3%</td>
<td>57.1%</td>
</tr>
<tr>
<td>Average F-Score</td>
<td>70.2%</td>
<td></td>
</tr>
<tr>
<td>Serial Scenes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ground Truth</td>
<td>22</td>
<td>16</td>
</tr>
<tr>
<td>Detected Scenes</td>
<td>28</td>
<td>13</td>
</tr>
<tr>
<td>Recall</td>
<td>100%</td>
<td>75%</td>
</tr>
<tr>
<td>Precision</td>
<td>79%</td>
<td>92%</td>
</tr>
<tr>
<td>F-Score</td>
<td>89.2%</td>
<td>82.6%</td>
</tr>
<tr>
<td>Average F-Score</td>
<td>85.9%</td>
<td></td>
</tr>
</tbody>
</table>

4. Conclusion

In this paper, we propose a novel NCuts based scene segmentation and categorization approach. Starting from a set of shots, we first calculate shot similarity from shot key frames. Then by modeling scene segmentation as a graph partition problem, we employ NCuts to find the optimal scene segmentation and automatically decide the optimal scene number by Q function. To discover more useful information from scenes, we analysis temporal layout patterns of shots, and automatically categorize scenes into two different types, i.e. parallel scene and serial scene. Extensive experiments are tested on movie, and TV series. The promising results demonstrate the good performance of the proposed scene segmentation and categorization method.

5. References


