EXPLORING VIDEO STRUCTURE BEYOND THE SHOTS

Yong Rui, Thomas S. Huang and Sharad Mehrotra

Beckman Institute for Advanced Science and Technology
Dept. of Electrical and Computer Engineering and Dept. of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
E-mail:{yrui,huang}@ifp.uiuc.edu and sharad@cs.uiuc.edu

ABSTRACT

While existing shot-based video analysis approaches provide users with better access to the video than the raw data stream does, they are still not sufficient for meaningful video browsing and retrieval, since: (1) the shots in a long video are still too many to be presented to the user; (2) shots do not capture the underlying semantic structure of the video, based on which the user may wish to browse/retrieve the video. To explore video structure at a semantic level, this paper presents an effective approach for video scene structure construction, in which shots are grouped into semantic-related scenes. The output of the proposed algorithm provides a structured video that greatly facilitates user's access. Experiments based on real-world movie videos validate the effectiveness of the proposed approach.

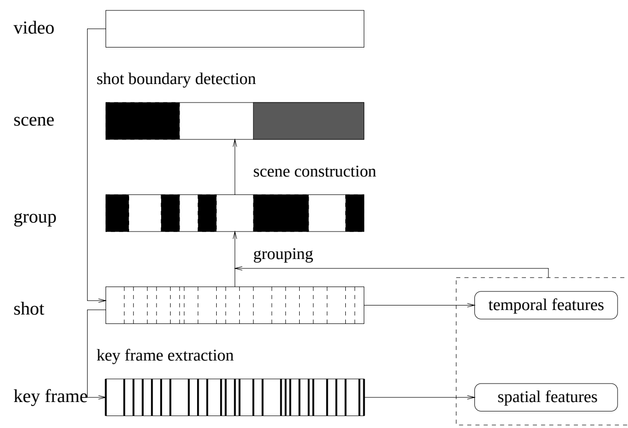

Figure 1: A hierarchical video representation

1. INTRODUCTION

Recent years have seen a rapid increase of the usage of digital video information. However, because of its length and rich content, efficient access to video is not an easy task. Raw video is an unstructured data stream, consisting of a sequence of video *shots*. Major visual content of shots can be represented by *key frames*. Similar shots can be grouped into *groups*. Semantically related shots can be merged into *scenes*, which depict and convey high-level concept or story. While shots are marked by physical boundaries, scenes are marked by semantic boundaries [1]. Scene boundary detection is a far more difficult research task compared with shot boundary detection and is the major focus of this paper. The above discussed video structure hierarchy is illustrated in Figure 1.

Most of the existing research effort has been devoted to the shot-based video analysis. In general, automatic shot boundary detection techniques can be classified into five categories, i.e. *pixel based*, *statistics based*, *transform based*, *feature based*, and *histogram based*. Several researchers claim that the histogram based approach achieves good trade-off between accuracy and speed [6].

After the shot boundaries are detected, corresponding key frames can then be extracted. Simple approaches may just extract the first and last frames of each shot as the key frames. More sophisticated key frame extraction techniques can be based on shot activity indicator [3] and shot motion indicator [4].

Reliable and accurate shot boundary detection and key frame extraction are important to successful video analysis. However, it is not uncommon that a modern movie contains a few thousand shots and key frames. This is evidenced in [5] – there are 300 shots in a 15-minute video segment of the movie "Terminator 2 - the Judgment Day" and the movie lasts 139 minutes. Because of the large number of key frames, a simple 1D array presentation of key frames for the underlying video is almost meaningless. More importantly, people watch the video by its semantic scenes not the physical shots or key frames. Shots can not convey meaningful semantics unless they are purposely organized into scenes. The construction of *scene* is thus of fundamental importance to many video applications [5, 2, 1].

This paper presents a novel framework in scene structure construction for video. The rest of the paper is organized as follows. In section 2, a novel framework for

This work was supported in part by ARL Cooperative Agreement No. DAAL01-96-2-0003 and in part by a CSE Fellowship, University of Illinois.

[1]Some of the early literatures in video parsing misused the phrase *scene change detection* for *shot boundary detection*. To avoid any later confusion, we will use *shot boundary detection* for the detection of physical shot boundaries while using *scene boundary detection* for the detection of semantic scene boundaries.

scene structuring is proposed. In section 3, the effectiveness of the proposed approach is validated by experiments over real-world movie video clips. Concluding remarks and future work are in section 4.

2. THE PROPOSED APPROACH TO SCENE STRUCTURE CONSTRUCTION

The proposed approach to scene structure construction consists of four modules: shot boundary detection and key frame extraction, spatio-temporal feature extraction, time-adaptive grouping, and scene structure construction. We discuss each of the modules in turn below:

2.1. Shot Boundary Detection and Key Frame Extraction

In the current implementation of the proposed approach, we use an approach similar to that in [6] for shot boundary construction and select the beginning and ending frames of a shot as the two key frames.

2.2. Spatio-Temporal Feature Extraction

At the shot level, the shot activity measure is extracted to characterize the temporal information of the shot:

$$\begin{aligned} Act_i &= \frac{1}{N_i - 1} \sum_{k=1}^{N_i - 1} Diff_{k,k-1} \\ Diff_{k,k-1} &= Dist(Hist(k), Hist(k-1)) \end{aligned}$$

where Act_i and N_i are the activity measure and number of frames for shot i; $Diff_{k,k-1}$ is the color histogram difference between frames k and $k-1$; $Hist(k)$ and $Hist(k-1)$ are the color histograms for frames k and $k-1$; $Dist()$ is a distance measure between histograms.

At the key frame level, visual features are extracted to characterize the spatial information. In the current algorithm, color histograms of the beginning and ending frames, $Hist(b_i)$ and $Hist(e_i)$, are used as the visual feature for the shot, where b_i and e_i are the beginning and ending frames of shot i. Based on the above discussion, a shot is modeled as:

$$shot_i = shot_i(b_i, e_i, Act_i, Hist(b_i), Hist(e_i)) \tag{1}$$

which captures both the spatial and the temporal information of a shot.

2.3. Time-Adaptive Grouping

To facilitate the later process, similar shots are grouped into a group, since similar shots have high possibility to be in the same scene. To be called similar shots, the following properties should be satisfied:

- *Visual similarity:* Similar shots should have similar spatial ($Hist(b_i)$ and $Hist(e_i)$) and temporal (Act_i) features.
- *Time locality:* Similar shots should be close to each other temporally [5]. For example, visually similar shots, if far apart from each other in time, seldom belong to the same scene and hence not to the same group.

In [5], Yeung et al. proposed a *time-constrained clustering* approach to group shots, where the similarity between two shots is set to 0 if their time difference is greater than a predefined threshold. We propose a more general *time-adaptive grouping* approach based on the two properties for similar shots described above. In our proposed approach, the similarity of two shots is an increasing function of visual similarity and a decreasing function of frame difference. Let i and j be the indexes for the two shots whose similarity is to be determined, where $shot\ j > shot\ i$. The calculation of the shot similarity is described as follows:

2.3.1. Calculate the shot color similarity:

1. Calculate the four raw frame color similarities: $FrmClrSim_{b_j,e_i}$, $FrmClrSim_{e_j,e_i}$, $FrmClrSim_{b_j,b_i}$, and $FrmClrSim_{e_j,b_i}$, where $FrmClrSim_{x,y}$ is defined as:
$$FrmClrSim_{x,y} = 1 - Diff_{x,y} \tag{2}$$
where x and y are two arbitrary frames.
2. To model the importance of time locality, we introduce the concept of *temporal attraction*, $Attr$, which is a decreasing function of the frame difference:
$$\begin{aligned} Attr_{x,y} &= max(0, 1 - \frac{|y-x|}{bLength}) \\ bLength &= MUL * avgShotLength \end{aligned}$$
where $avgShotLength$ is the average shot length of the whole video stream; MUL is a constant which controls how fast the temporal attraction will decrease to 0. The above definition of *temporal attraction* says that the farther apart the frames, the less the *temporal attraction*. Experimentally, we find that $MUL = 10$ gives good results.
3. Convert the raw similarities to *time-adaptive* similarities, which capture both the visual similarity and time locality:
$$FrmClrSim'_{x,y} = Attr_{x,y} \times FrmClrSim_{x,y} \tag{3}$$
4. The color similarity between shots i and j is defined as the maximum of the four frame similarities, e.g. $ShtClrSim_{i,j} = max(FrmClrSim'_{b_j,e_i}, FrmClrSim'_{e_j,e_i}, FrmClrSim'_{b_j,b_i}, FrmClrSim'_{e_j,b_i})$.

2.3.2. Calculate the shot activity similarity:

$$\begin{aligned} ShtActSim_{i,j} &= Attr_{cnt} \times |Act_i - Act_j| \\ Attr_{cnt} &= max(0, 1 - \frac{(b_j+e_j)/2 - (b_i+e_i)/2}{bLength}) \end{aligned}$$

where $Attr_{cnt}$ is the *temporal attraction* between the two center frames of shot i and shot j.

2.3.3. Calculate the overall shot similarity:

$$ShtSim_{i,j} = W_C * ShtClrSim_{i,j} + W_A * ShtActSim_{i,j} \tag{4}$$

where W_C and W_A are appropriate weights for color and activity measures.

2.4. Scene Structure Construction

Similar shots are grouped into a group, but even non-similar groups can be grouped into a single scene if they are semantically related. In video, even though two or more processes are developing simultaneously, they have to be displayed sequentially, one after another. This is common in movie. For example, when two people are talking to each other, even though both people contribute to the conversation, the movie switches back and forth between these two people. In this example, clearly there exist two groups, one corresponding to person A, and the other corresponding to person B. These two groups are semantically related and should be merged together into a single scene. The algorithm that collects semantically related groups into a scene is described below:

[**Main algorithm**]
Input: Video shot sequence, $S = \{shot\ 0, ..., shot\ i\}$.
Output: Video structure in terms of *scene*, *group*, and *shot*.
Procedure:

1. Initialization: assign shot 0 to group 0 and scene 0; initialize the group counter $numGrp = 1$; initialize the scene counter $numScn = 1$.
2. If S is empty, quit; otherwise get the next shot. Denote this shot as shot i.
3. Test if shot i can be merged to an existing group:
   (a) Compute the similarities between the current shot and existing groups: Call $findGrpSim()$.
   (b) Find the maximum group similarity:
   $$\begin{aligned} maxGrpSim_i &= \max_g \ GrpSim_{i,g} \\ g &= 1, ..., numGrp \end{aligned}$$
   where $GrpSim_{i,g}$ is the similarity between shot i and group g. Let the group of the maximum similarity be group g_{max}.
   (c) Test if this shot can be merged into an existing group:
   If $maxGrpSim_i > grpThd$, where $grpThd$ is a predefined threshold:
      i. Merge shot i to group g_{max}.
      ii. Update the video structure: Call $updtStrt()$.
      iii. Goto Step 2.
   otherwise:
      i. Create a new group containing a single shot i. Let this group be group j.
      ii. Set $numGrp = numGrp + 1$.
4. Test if shot i can be merged to an existing scene:
   (a) Calculate the similarities between the current shot i and existing scenes: Call $findScnSim()$.
   (b) Find the maximum scene similarity:
   $$\begin{aligned} maxScnSim_i &= \max_s \ ScnSim_{i,s} \\ s &= 1, ..., numScn \end{aligned}$$
   where $ScnSim_{i,s}$ is the similarity between shot i and scene s. Let the scene of the maximum similarity be scene s_{max}.
   (c) Test if shot i can be merged into an existing scene:
   If $maxScnSim_i > scnThd$, where $scnThd$ is a predefined threshold:
      i. Merge shot i to scene s_{max}.
   otherwise:
      i. Create a new scene containing a single shot i and a single group j.
      ii. Set $numScn = numScn + 1$.
5. Goto Step 2.

The input to the algorithm is an unstructured video stream while the output is a structured video consisting of scenes, groups, shots, and key frames.

[**findGrpSim**]
Input: Current shot and group structure.
Output: Similarity between current shot and groups.
Procedure:

1. Denote current shot as shot i.
2. Calculate the similarities between shot i and existing groups:
   $$GrpSim_{i,g} = ShtSim_{i,g_{last}} \quad , g = 1, ..., numGrp \tag{5}$$
   where g is the index for groups and g_{last} is the last (most recent) shot in group g. That is, the similarity between current shot and a group is the similarity between the current shot and the most recent shot in the group. The reason of choosing the most recent shot to represent the whole group is that all the shots in the same group are visually similar and the most recent shot has the largest *temporal attraction* to the current shot.

[**findScnSim**]
Input: Current shot, group structure and scene structure.
Output: Similarity between current shot and scenes.
Procedure:

1. Denote current shot as shot i.
2. Calculate the similarity between shot i and existing scenes:
   $$ScnSim_{i,s} = \frac{1}{numGrp_s} \sum_{g}^{numGrp_s} GrpSim_{i,g} \tag{6}$$
   where s is the index for scenes; $numGrp_s$ is the number of groups in scene s; and $GrpSim_{i,g}$ is the similarity between current shot i and g^{th} group in scene s. That is, the similarity between current shot and a scene is the average of similarities between current shot and all the groups in the scene.

[**updtStrt**]
Input: Current shot, group structure, and scene structure.
Output: An updated version of group structure and scene structure.
Procedure:

1. Denote current shot as shot i and the group having the largest similarity to shot i as group g_{max}. That is, shot i belongs to group g_{max}.
2. Define two shots *top* and *bottom*, where *top* is the second to the last shot in group g_{max} and *bottom* is the last shot in group g_{max} (i.e. current shot).

Table 1. Scene structure construction results.

movie name	frames	shots	groups	scenes
BMC	21717	133	27	5
PW	27951	186	25	7
GR	14293	84	13	6
MS	35817	195	28	12
ST	18362	77	10	6
SW	23260	180	31	21
TR	35154	329	65	21

3. For any group g, if any of its shots ($shot\ g_j$) satisfies the following condition

$$top < shot\ g_j < bottom \tag{7}$$

merge the scene that group g belongs to into the scene that group g_{max} belongs to. That is, if a scene contains a shot which is interlaced with the current scene, merge the two scenes.

3. EXPERIMENTAL RESULTS

In all the experiments reported in this section, the video streams are MPEG compressed, with the digitization rate equal to 30 frames/sec. To validate the effectiveness of the proposed approach, representatives of various movie types are tested. Specifically, *The Bridges in Madison County (BMC)* (romantic-slow), *Pretty Woman (PW)* (romantic-fast), *Grease (GR)* (music), *The Mask (MS)* (comedy), *Star Trek (ST)* (science fiction-slow), *Star War (SW)* (science fiction-fast), and *Total Recall (TR)* (action) are used in our experiments. Each video clip is about 10-20 minutes long. The experimental results are shown in Table 1.

In all the experiments, the scene structures created by the algorithm are judged by human who watches the entire video. Although *scene* is a semantic concept, relative agreement can be reached among different people. Based on the scene structure created by the algorithm, high level text descriptions can be further associated to the video structure to facilitate user's access to the video. This is illustrated in Figure 2.

Figure 2: Scene structure for BMC

In Figure 2, five scenes are created from the 21717-frame video clip (BMC). Along with the text description of each scene, representative frames are also displayed to the user. Clicking on the representative frames will start playing the video of the corresponding scene. This scene structure not only provides the user with a non-linear access to the video (in contrast to conventional linear *fast-forward* and *rewind*), but also gives the user a global "picture" of the whole story of the video. If, instead, we were using 1D array of key frames to present the video, $2 \times 133 = 266$ frames have to be presented sequentially. Because of the 1D linear display nature, even if a user can patiently browse through all the 262 frames, it is still difficult for him or her to perceive the underlying story structure.

4. CONCLUSIONS AND FUTURE WORK

This paper presents an effective approach for video scene structure construction, in which shots are grouped into semantic-related scenes. The output of the proposed algorithm provides a structured video that greatly facilitates user's access. As realized by many researchers, the construction of semantic scenes from syntactic shots is a difficult research task. No claim is made in this paper to correctly construct the scene structure fully automatically, but rather to provide a video structure analysis tool which can assist human in constructing video scene structures.

For the future work, we are currently exploring more reliable and semantic-rich features based on audio content, close-caption content and object based content.

5. REFERENCES

[1] Hisashi Aoki, Shigeyoshi Shimotsuji, and Osamu Hori. A shot classification method of selecting effective key-frames for video browsing. In *Proc. ACM Conf. on Multimedia*, 1995.

[2] Ruud M. Bolle, Boon-Lock Yeo, and Minerva M. Yeung. Video query: Beyond the keywords. Technical report, IBM Research Report, Oct 17 1996.

[3] P. O. Gresle and T. S. Huang. Gisting of video documents: A key frames selection algorithm using relative activity measure. In *The 2nd Int. Conf. on Visual Information Systems*, 1997.

[4] Wayne Wolf. Key frame selection by motion analysis. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, 1996.

[5] Minerva Yeung, Boon-Lock Yeo, and Bede Liu. Extracting story units from long programs for video browsing and navigation. In *Proc. IEEE Conf. on Multimedia Computing and Systems*, 1996.

[6] HongJiang Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems*, 1(1), 1993.