VideoGraph: A New Tool for Video Mining and Classification [*]

Jia-Yu Pan, Christos Faloutsos
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
{jypan, christos}@cs.cmu.edu

ABSTRACT

This paper introduces *VideoGraph*, a new tool for video mining and visualizing the structure of the plot of a video sequence. The main idea is to "stitch" together similar scenes which are apart in time. We give a fast algorithm to do stitching and we show case studies, where our approach (a) gives good features for classification (91% accuracy), and (b) results in *VideoGraph*s which reveal the logical structure of the plot of the video clips.

1. INTRODUCTION

In this paper, we focus on automatically determining and visualizing the 'story-plot' of a video clip. This helps us distinguish between different video types, e.g., news stories versus commercials. The problem is: given a video clip, determine the evolution of its story-plot. For example, do similar scenes alternate, as is the case of dialogue?

2. PROPOSED METHOD

The main idea behind our approach is to spot shots that tend to occur repeatedly. For example, in an interview, shots of the interviewer and the interviewee will occur often, and intermixed. An important concept we introduce is the *shot-group*. Recall that a *shot* is a set of consecutive similar frames.

Definition 1. A *shot-group* is a set of shots, that are similar according to our similarity function.

Notice that shots of a shot-group may or may not be consecutive in time. Our experience showed that we should mainly consider shot-groups that consist of many shots. Specifically, we make the following distinctions:

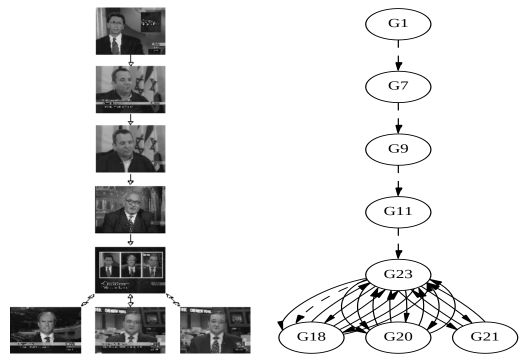

Figure 1: The VideoGraph of a news story. An alternating starlike structure is shown, which corresponds to an interview in the latter part of the story.

A shot-group is called *persistent* if it contains multiple, consecutive shots; while a shot-group that contains multiple, non-consecutive shots, is called *recurrent*. A shot-group that is both persistent and recurrent is called a *basic* shot-group; it is exactly the type of shot-group that we use to construct a VideoGraph. Intuitively, basic shot-groups are the ones that contain shots that occur often, and therefore should be helpful in revealing the plot evolution. This is achieved through the VideoGraph:

Definition 2. The VideoGraph of a video clip is a directed graph, where every node correponds to a shot-group, and where edges indicate temporal succession.[1]

Figure 1 gives a news clip and its VideoGraph. Notice the pronounced star-like pattern, which, it turns out, corresponds to an interview. The graph contains only the basic shot-groups, namely G1, G7, G9, G11, G23, G18, G20, G21. Notice the heavy traffic between shot-groups G23, G18, G20 and G21, with G23 being the center of the 'star'-shape. In the same Figure 1 we also show the key frames from each shot-group. The key frame of a shot-group was randomly selected among the frames of the participating shots. Notice again that the 'center' shot-group, G23, indeed is the recurring shot of the interview, which contains shots when the

[*] This material is based on work supported by the National Science Foundation under Cooperative Agreement No. IRI-9817496.

JCDL'01, June 24-28, 2001, Roanoke, Virginia, USA.

[1] Dashed edges indicate that some non-basic shot-groups have been deleted along the way.

reporter proposed questions to the invited speakers. The shot-groups which interact with G23 heavily, i.e. G18, G20, and G21, contains exactly the shots of the invited speakers replying to the reporter's questions.

Next we describe the algorithm to generate the VideoGraph for a video clip. First, use any off-the-shelf shot detection technique to break the video into shots. Next, use our 'stitching' algorithm (described next) to group similar shots into shot-groups. Each shot-group is assigned a label. Then, collect statistics (i.e. count of persistent shot-groups, count of recurrent ones, e.t.c). Finally, keep only the shot-groups that are both persistent and recurrent; consider the edges among them, indicating temporal succession; and use an off-the-shelf graph-layout tool (e.g., *dot*[2]to draw the resulting graph. This is exactly the VideoGraph for this video clip.

There are two issues left to discuss: (a) which statistics to keep and (b) how the 'stitching' algorithm works. For the first question, we propose the following statistics: the number of I-frames, the number of shots, and the number of shot-groups in the video clip; the number of persistent, recurrent and basic shot-groups; the count of shots in each of the three classess of shot-groups; the average number of shots per shot-group; and the percentage of shots that are within persistent/recurrent shot-groups.

The second issue to discuss is our *video stitching algorithm ('VS')*, and specifically, how it decides whether two shots should be grouped together or not. Due to lack of space, we just give the main idea: For each I-frame, we used the DCT-coefficients of the macroblocks (MB), and, specifically, the first m=20 attributes, after using a dimensionality reduction method (*FastMap*[1]). Thus, each shot is a 'cloud' of m-dimensional points. We group together two such 'clouds', if the density of the union is comparable to the original, individual densities.

3. EXPERIMENTS - RESULTS

We used MPEG-1 video clips segmented from CNN news reports. Each clip is either a news story or a continuous series of commercials. We chose the technique from [2] to do scene shot detection.

Story-plot visualization Figure 2 shows the VideoGraphs for several news stories (upper part)and commercials (lower part).It is clear that most of the commercial clips have fewer persistent and recurrent shot-groups and much fewer basics, than news stories do. Consequently, VideoGraphs of commercial clips are more likely to be an empty graph or a graph of one or two nodes (Figure 2). This is due to the fact that commercials rarely have a 'plot' being, instead, a collection of shots are not revisited.

Classification Here we illustrate that the features we extracted capture the logical structure of a video clip quite well and therefore are promising for video classification. We conducted a classification experiment based on 68 news stories and 33 commercials, whose total size is about 2.5 gigabyte. Each clip is represented by 20 features (count of recurrent shot-groups, count of persistent shot-groups, e.t.c.) We conducted 5-fold cross-validation using an ID3 classifier and we achieved classification accuracy of 91% (plus or minus 0.16 percentage points, for the 95% confidence interval). We used an off-the-shelf package, $\mathcal{MLC}$++[3], for the classification.

[2]Available at http://www.research.att.com/sw/tools/graphviz/

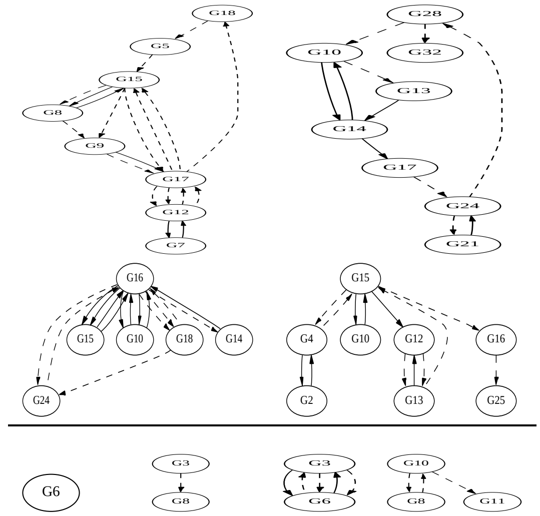

Figure 2: VideoGraphs of news stories (upper part) and commercials (lower part). Note the more complicated structure of news stories.

4. CONCLUSIONS

We presented the *VideoGraph*, a new tool for video mining and story plot visualization. The heart of our approach is to group similar shots together, even if they are not consecutive. We give algorithms that "stitch" similar shots into shot-groups, and automatically derive VideoGraphs. We also show how to derive features (e.g. number of recurrent shot-groups, etc.) for video mining and classification. In a case study with news and commercials, our proposed features achieved 91% classification accuracy, without even using the audio information. VideoGraphs can also be used as video representatives for efficient browsing in digital video libraries such as Informedia [4], and are useful on keyframe selection.

5. REFERENCES

[1] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *Proceedings of the ACM SIGMOD Conference*, pages 163–174, 1995.

[2] V. Kobla, D. Doermann, and C. Faloutsos. Videotrails: Representing and visualizing structure in video sequences. *Proceeding of the Fifth ACM International Multimedia Conference*, pages 335–346, November 1997.

[3] R. Kohavi, D. Sommerfield, and J. Dougherty. Data mining using MLC++: A machine learning library in C++. *Tools with Artificial Intelligence*, `http://www.sgi.com/Technology/mlc`.

[4] H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons learned from the creation and deployment of a terabyte digital video library. *IEEE Computer*, 32(2):66–73, February 1999.