A Unified Shot Boundary Detection Framework Based on Graph Partition Model

Jinhui Yuan*
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, P. R. China
yuan-jh03@mails.tsinghua.edu.cn

Jianmin Li Fuzong Lin Bo Zhang
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, P. R. China
{lijianmin,linfz,dcszb}@mail.tsinghua.edu.cn

ABSTRACT

In this paper, we propose a unified shot boundary detection framework by extending the previous work of graph partition model with temporal constraints. To detect both the abrupt transitions (CUTs) and gradual transitions (GTs, excluding fade out/in) in a unified way, we incorporate temporal multi-resolution analysis into the model. Furthermore, instead of ad-hoc thresholding scheme, we construct a novel kind of feature to characterize shot transitions and employ support vector machine (SVM) with active leaning strategy to classify boundaries and non-boundaries. Extensive experiments have been carried out on the platform of TRECVID benchmark. The experimental results show that the proposed framework outperforms some others and achieves satisfactory results.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*abstracting methods*

General Terms

algorithms, management

Keywords

graph partition, temporal multi-resolution, active learning

1. INTRODUCTION

Shot boundary detection is a prerequisite step of content based video retrieval. A large number of methods have been proposed to perform shot boundary detection. In this paper, we propose a novel shot boundary detection framework which shows superiority to the existing methods when evaluated on the platform of TRECVID benchmark.

In the previous work [4], we propose a graph partition model with temporal constraints to perform temporal data segmentation. The model tries to minimize the association between the two subgraphs while maximize the association within each subgraph with the following criterion:

$$Mcut(A,B) = \frac{cut(A,B)}{assoc(A)} + \frac{cut(A,B)}{assoc(B)}. \quad (1)$$

For temporal data, the optimal segmentation can be got at some i with the minimal score ($i \in \{1, \cdots, N-1\}$):

$$score(i) = Mcut(\{1,2,\cdots,i\}, \{i+1, i+2, \cdots, N\}). \quad (2)$$

In this paper we apply and extend the previous work to fulfil temporal video segmentation. We incorporate a novel temporal multi-resolution analysis algorithm into the model thus the CUTs and GTs can be detected in a unified way. In addition, with a novel constructed feature, we employ SVM with active learning strategy to classify boundaries and non-boundaries.

The remainder of the paper is organized as follows. Section 2 introduces how to employ the graph partition model to perform temporal video segmentation. Section 3 presents the temporal multi-resolution method. Section 4 introduces the support vector machine framework with active learning strategy. Section 5 evaluates the algorithms on the platform of TRECVID benchmark. And finally we conclude the paper in Section 6.

2. TEMPORAL VIDEO SEGMENTATION WITH GRAPH PARTITION MODEL

Given a video sequence, we treat each frame as a node and link each other with an edge, and a weighted graph $G(V,E)$ can be constructed. Thus, the shot boundary detection is formulated as a graph partition problem.

2.1 How to define the edge weight w_{ij}?

Let H^i be a k bins color histogram of the i-th frame and adopt histogram intersection method to measure the similarity, a reasonable definition of w_{ij} is:

$$w_{ij} = \sum_k \frac{min(H_k^i, H_k^j)}{H_k^i} \times \begin{cases} e^{\frac{-\|i-j\|_2^2}{\sigma^2}} & \text{if } |i-j| < \frac{r}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where σ is a factor reflecting the similarity decaying with the temporal interval increasing, and r denotes the maxi-

*Supported by National Natural Science Foundation of China (60135010, 60321002) and Chinese National Key Foundation Research & Development Plan (2004CB318108).

MM'05, November 6–11, 2005, Singapore.

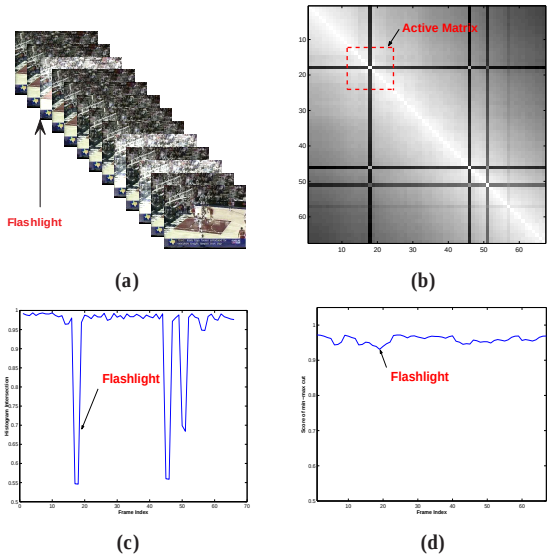

Figure 1: (a): A video sequence with three flashlights occurring. (b): The visualization of the corresponding similarity matrix, in which the red rectangle indicates the range of *active matrix*. (c): The curve obtained by the comparison between the successive frames. (d): The curve obtained according to Equation 4.

mum range in which the frames are considered to influence each other. Therefore, the calculation is restricted in a $r \times r$ sub-matrix, which we call *active matrix*. Consequently, the Equation 2 can be simplified as:

$$score(i) = Mcut(\{i-\frac{r}{2}, \cdots, i\}, \{i+1, \cdots, i+\frac{r}{2}\}). \quad (4)$$

2.2 The Algorithm

In summary, the video temporal segmentation algorithm consists of the following steps:

Step 1. Given a video sequence, treat each frame as a node and link each other by an edge, to construct a weighted graph $G(V, E)$.

Step 2. Compute w_{ij} according to Equation 3, obtaining a similarity matrix $\mathbf{W}$.

Step 3. Calculate scores of the N–1 feasible cuts according to Equation 4.

Step 4. Select feasible cuts whose scores are the local minima of the corresponding neighborhoods within a radius of $\frac{r}{2}$.

Step 5. Declare the cuts whose scores are below a predefined threshold as CUTs.

2.3 Analysis of the Algorithm

One prominent advantage of the approach is the robustness. Instead of pair-wise comparison by traditional novelty detection, the graph partition method performs boundary detection by considering the feature variation in a local neighborhood. As shown in Figure 1, there are three flashlights occurring during the video sequence. In the method

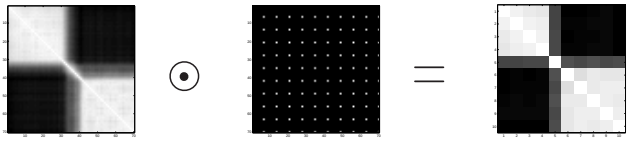

Figure 2: The three matrices are $\mathbf{W}_i^\delta$, $\mathbf{M}^\delta$, $\mathbf{V}_i^r$ respectively, where $r = 10$ and $\delta = 7$. The operation "$\odot$" means the Hadamard multiplication. The white blocks in $\mathbf{M}^\delta$ represent entries 1. The first two matrices are square ones of width 70, while the last one is square matrix of width 10.

based on pair-wise comparison, the three corresponding sharp valleys are usually considered as boundaries of shots. Consequently, many false alarms are caused. While in the proposed approach, the strong connectivity among the frames before and after the flashlight frame makes it unlikely to separate the sequence to two parts. The method is robust to various abrupt illumination change, and thus no specific flashlight detector is needed. The experiment in Section 5 will further confirm this analysis.

3. TEMPORAL MULTI-RESOLUTION ANALYSIS

By observing the patterns on the similarity matrix, we can find that there is a clear "chessboard" pattern for CUT boundary, while for GT boundary, it may yield a blurry pattern on the similarity matrix. Conversely, noticing that, at a lower resolution, whatever the length of the GT is, there will always be a "chessboard" pattern clear enough. To detect CUTs and GTs simultaneously, we re-define the score of the i-th feasible cut as follows:

$$score(i, \delta) = Mcut(\{i-\frac{r}{2}\times\delta, \cdots, i-\delta, i\}, \{i+\delta, \cdots, i+\frac{r}{2}\times\delta\}). \quad (5)$$

where δ is the sampling rate of the frames, $\delta \in \{1, 2, \cdots\}$. Equation 5 means that when calculating the score of the i-th feasible cut, instead of involving all the frames in a neighborhood of $\{i-\frac{r}{2}, \cdots, i+\frac{r}{2}\}$, the algorithm only samples every δ frames in a larger range of $\{i-\frac{r}{2}\times\delta, \cdots, i+\frac{r}{2}\times\delta\}$. With the δ varying, multiple temporal resolution graphs can be constructed.

To facilitate the computation, we define a square *selective matrix* $\mathbf{M}^\delta$ of width $r \times \delta$, in which entry 0 indicating the the corresponding frame is not sampled, 1 representing the frame sampled. Let $\mathbf{W}_i^\delta$ be the square sub-matrix of $\mathbf{W}$, centering at i and with the width $r \times \delta$. To calculate $score(i)$ at the resolution δ, the algorithm just performs Hadamard multiplication of $\mathbf{W}_i^\delta$ and $\mathbf{M}^\delta$, and thus results in a new matrix $\mathbf{V}_i^\delta$, in which the entries corresponding to the non-sampled frames equal 0. Furthermore, ignoring the 0 entries, $\mathbf{V}_i^\delta$ can be restricted an equivalent but smaller square matrix $\mathbf{V}_i^r$ of width r. The above process is depicted in Figure 2.

4. SUPPORT VECTOR MACHINE ACTIVE LEARNING

On the curve of scores, each boundary corresponds to a sufficiently small local minimum. However, not every local minimum is a shot boundary. Only by evaluating the magnitudes of the local minima can not successfully distin-

guish boundaries and non-boundaries. In this paper, we propose to employ SVM with active learning strategy to classify boundaries and non-boundaries according to the shapes of local minima.

4.1 Feature Construction

Formally, let S_i^δ denote $score(i, \delta)$, defined by Equation 5. Then the feature, which characterizes the shape of the valley centering at i on the δ resolution curve, is define as:

$$f_i^\delta = [S^\delta_{i-\frac{r}{2}\times\delta}, S^\delta_{i-(\frac{r}{2}-1)\times\delta}, \cdots, S_i^\delta, \cdots, S^\delta_{i+(\frac{r}{2}-1)\times\delta}, S^\delta_{i+\frac{r}{2}\times\delta}] \quad (6)$$

Obviously, for each δ, the f_δ is a $(r+1)$ dimension feature. To make full use of the information across different resolutions, i.e. $\delta \in \{1, 3, 5, 7, 9\}$, we concatenate those features to construct a new feature vector as:

$$F_i = [f_i^1, f_i^3, f_i^5, f_i^7, f_i^9] \quad (7)$$

Given $r = 10$, F_i is a 55 dimension feature including the the i-th feasible cut's shape information at different resolutions.

4.2 Active Learning Strategy

Manually labeling the local minima as boundaries or non-boundaries to obtain a training set is a tedious job. Similar to the idea of [3], one heuristic active learning criterion is employed. We assume that the examples difficult for thresholding method to clearly classify lie near the dividing hyperplane of the SVM. We firstly collect all the valleys which are under a specified threshold θ from all the available local minima. These valleys collected, including real boundaries and various false alarms, constitute the training set. Then the real boundaries are labeled as positive examples, and the false alarms are labeled as negative examples. Here the threshold θ is set low enough to guarantee almost all of the boundaries are collected. Meanwhile, with the threshold θ, we can remove a lot of local minima which will not influence the position of the SVM's hyperplane.

5. EXPERIMENTS

In this section, we evaluate the proposed framework on the platform of TRECVID benchmark [1]. Several comparison experiments have been designed to justify the ideas of this paper.

5.1 Experimental Setup

All the 2003 and 2004 TRECVID test collections for the task of shot boundary detection (SBD) are adopted. So far we have not considered incorporating the fade out/in (FOI) detection in the framework. Therefore, we re-edit the video collections by transforming FOIs into CUTs. We call the video collection without FOIs as "**DATA_NO_FOI**". On the other hand, in order to focus on the comparison of the characteristics of different algorithms, we apply the algorithms to fulfil CUT detection instead of boundary detection in several experiments. Thus, we create another video collection without GTs by re-editing the GTs into CUTs. This collection is called as "**DATA_NO_GT**". The algorithms evaluated on the this data set are all single resolution implementations, i.e. $\delta = 1$. Similar to other information retrieval task, the performance is evaluated by *recall* and *precision* criteria. To rank performance of different algorithms, F_1 measure, a harmonic average of *recall* and *precision* is adopted.

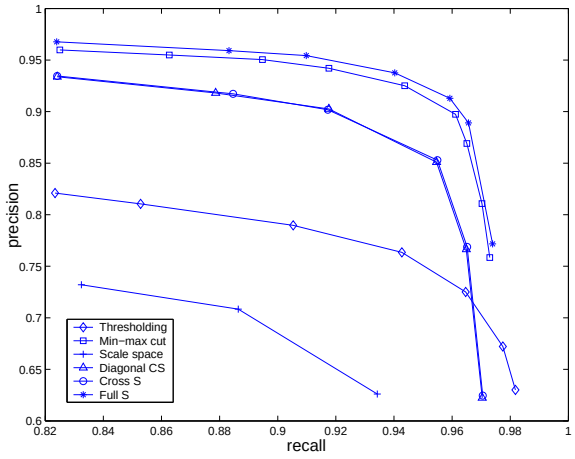

Figure 3: Performance comparison with other related algorithms.

5.2 Performance Comparison

5.2.1 Graph Partition Model

To show the effectiveness of the graph partition model, we implement five other related approaches and evaluate them on the "**DATA_NO_GT**". They are:

Thresholding	Directly compare the difference of successive frames.
Min-max cut	The algorithm proposed in Section 2.2.
Scale space	Kernel correlation depicted in [2].
Diagonal CS	Kernel correlation depicted in [2].
Cross S	Kernel correlation depicted in [2].
Full S	Kernel correlation depicted in [2].

As the Figure 3 shows, all the five approaches via multi-pair comparison outperform the pair-wise comparison method "**Thresholding**". Both the "**Diagonal CS**" and the "**Cross S**" kernels emphasize the dissimilarity between the different shots, and they performs almost the same. The "**Full S**" somehow outperforms the proposed "**Min-max cut**" method and both of them perform best. This is not surprising, since they both consider the similarity between different shots and within the the same shot. In fact, the "**Full S**" is equivalent to an alternative definition of min-max cut:

$$Mcut(A, B) = assoca(A) + assoca(B) - 2 \times cut(A, B) \quad (8)$$

Note that our experimental result is inconsistent with that of Cooper [2], in which the author claims that the "**Full S**" performs worst. However, we believe that our results are more reliable. Firstly, we evaulate them on a more delicate data set of "**DATA_NO_GT**". Secondly, we design a more straightforward experimental setup, in which a simple histogram feature and thresholding method are adopted, while Cooper has employed multi-scale features and KNN to classify CUTs and non-CUTs.

5.2.2 Temporal Multi-resolution Analysis

To examine the effectiveness of temporal multi-resolution analysis, we implemented four different methods and evalated

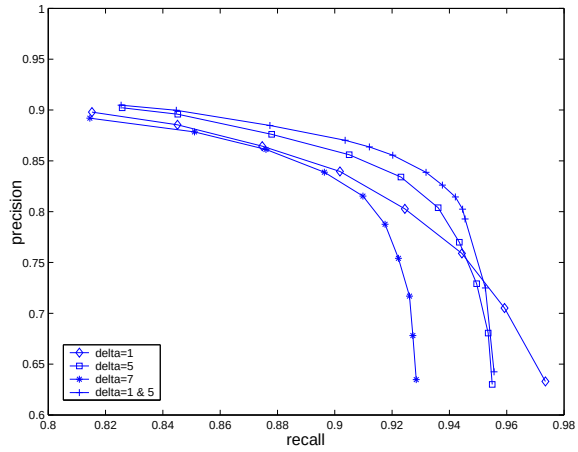

Figure 4: Evaluation of temporal multi-resolution analysis.

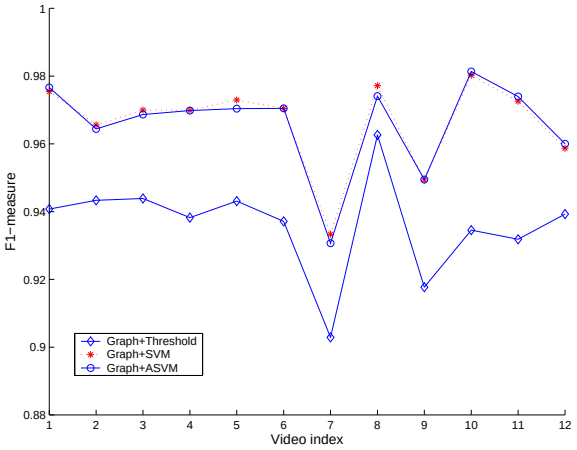

Figure 5: Support vector machine with active learning strategy.

them on "**DATA_NO_FOI**":

delta=1	$\delta = 1$ for the algorithm of Section 2.2.
delta=5	$\delta = 5$ for the algorithm of Section 2.2.
delta=7	$\delta = 7$ for the algorithm of Section 2.2.
delta=1 & 5	Multi-resolution analysis, combining the $\delta = 1$ and $\delta = 5$ for the algorithm of Section 2.2.

As shown in Figure 4, **delta=5** outperforms **delta=1**. That is because at high resolution, most of the long GTs are missed. However, **delta=7**, which is the lowest resolution, performs worse than the other two higher resolution methods. Then why the lowest resolution performs worst? We have examined the detection result of **delta=7** and find that, although the lower resolution helps to detect long GTs, it also suffers the enlargement of disturbances of motion. Since we have not incorporate the post processing with motion filtering, the **delta=7** performs worst. Finally, the **delta=1 & 5**, with the multi-resolution analysis, performs best. We can expect that, if we import the analysis of motion activity, the performance of the proposed method will be further improved.

5.2.3 SVM with Active Learning Strategy

The experiments are performed on "**DATA_NO_GT**" and a single resolution feature of $\delta \in \{1\}$ is adopted. Two of the 2003 test collections are chosen for training and all the twelve videos of 2004 test collections are used for testing. RBF(radial basis function) is adopted as the kernel function. $C = 400$ and $\sigma = 1.0$ are selected after a cross validation process. For the "**Graph+Threshold**" method, the best result among various threshold settings is chosen to be compared.

Graph+Threshold	Algorithm of Section 2.2.
Graph+SVM	SVM Trained with randomly selected samples.
Graph+ASVM	SVM Trained with active learning strategy.

As shown in Figure 5, both the SVM methods have outperformed the thresholding method. The performance of "**Graph+ASVM**" is comparable to that of "**Graph+SVM**", while the size of the training set for active learning is just about $\frac{1}{6}$ of that of "**Graph+SVM**", 1090 for the former and 6948 for the latter.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we extend our previous work and design a unified shot boundary detection framework based on graph partition model. Extensive experiments on the benchmark of TRECVID have been performed to justify the proposed ideas. Nevertheless, several problems remains for further research. For example, we have not incorporated the detection of FOIs into the framework. With multi-resolution analysis, we have to design methods to effectively reduce the disturbances of motion. How to effectively make use of information across different resolutions is also an important problem to solve. Despite the remaining problems above, we believe that the proposed framework may be a promising method for nearly perfect temporal segmentation of videos.

7. REFERENCES

[1] Trec video retrieval evaluation, http://www-nlpir.nist.gov/projects/trecvid/.

[2] M. Cooper. Video segmentation combining similarity analysis and classification. In *ACM Multimedia*, October 2004.

[3] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning*, pages 839–846. Morgan Kaufmann, San Francisco, CA, 2000.

[4] J. Yuan, B. Zhang, and F. Lin. Graph partition model for robust temporal data segmentation. In *Proc. of PAKDD, LNAI*, volume 3518, pages 758–763, 2005.