# Rushes Video Summarization using Audio-Visual Information and Sequence Alignment

Yang Liu[1], Yan Liu[1], Tongwei Ren[1, 2], Keith C. C. Chan[1]
[1]Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210093, P. R. China
{csygliu, csyliu, cstwren, cskcchan}@comp.polyu.edu.hk

## ABSTRACT

This paper describes our system and methodologies for the BBC rushes video summarization task of TRECVID 2008. The procedure of the system is composed of three major steps: shot detection, irrelevant and repetitive subshot removal, and final summary generation. First, we segment the original rushes video into subshots according to the difference and accumulative difference of local color histogram between consecutive frames. Second, we recognize the irrelevant subshots, such as subshots of color bar, pure gray frames, and clapper board. We propose a novel video sequence alignment algorithm to detect repetitive subshots. After removing the irrelevant and repetitive subshots, we generate the final summary using the remaining informative and representative subshots. The evaluation from TRECVID 2008 shows that our system can generate good video summaries.

## Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding –*Video analysis*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing –*Abstracting methods*

## General Terms

Algorithms, Experimentation

## Keywords

Rushes Editing, TRECVID, Video Summarization

## 1. INTRODUCTION

With the rapid development of multimedia technologies, and significant increase of the volume of digital video, the video summarization/abstracting technologies attract more and more attention.

An efficient summary can provide a quick and comprehensive overview of original video to users. Many works have been done in news, sports, and instructional video summarization [1]-[3]. However, how to generate a summary from unedited rushes, which contain lots of irrelevant (such as color bar) and highly redundant (such as retake) contents, is still a challenging problem.

NIST has organized a summarization task in TRECVID on BBC rushes videos. P. Over *et. al.* provide state-of-the-art introduction to the research background, problem statement, data preparation and result evaluation in [4]. Compared to the task in TRECVID 2007, a remarkable difference in this year is that the permitted duration of each summary is shortened from 4% to at most 2% of the original video. This constraint influences many aspects in summarization and makes the task more challenging.

According to the requirement in TRECVID 2008, we developed a new system for the summarization task. The basic framework is based on our work in last year and some more effective techniques are integrated into the system.

Figure 1 shows the framework of our system. The input is the unedited BBC rushes video, and the output is the final summary. The entire procedure is composed of three important steps: subshot boundary detection, irrelevant and repetitive subshot removal and final summary generation.

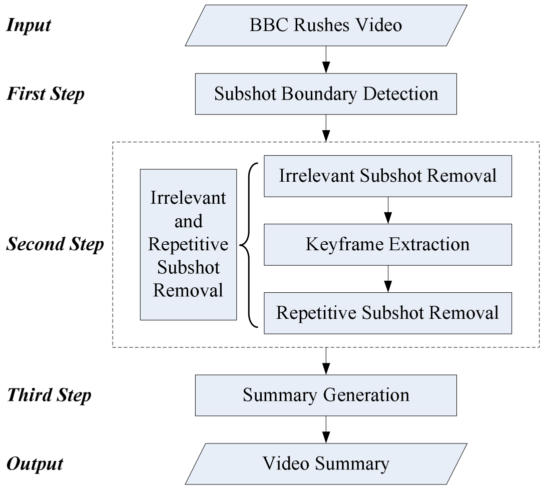

Input
BBC Rushes Video
First Step
Subshot Boundary Detection
Second Step
Irrelevant and Repetitive Subshot Removal
Irrelevant Subshot Removal
Keyframe Extraction
Repetitive Subshot Removal
Third Step
Summary Generation
Output
Video Summary

**Figure 1. Framework for BBC rushes summarization system**

First, local color histogram is computed for each frame in the original rushes, and the video is segmented into a set of subshots based on the difference and accumulative difference between frames on local color histogram. Second, the irrelevant subshots are detected by two audio-video features, sum of gradient in vertical direction and transition of sound energy. In each remaining subshot, the keyframes are extracted, and the repetitive

*TVS'08*, October 31, 2008, Vancouver, British Columbia, Canada.
Copyright 2008 ACM 978-1-60558-309-9/08/10...$5.00.

subshots are detected based on keyframe sequence alignment. After removing irrelevant and repetitive subshots, the final summary is generated by extending the keyframes in the retained subshots with uniform rate.

The rest of this paper is organized as follows: Section 2 describes the details of subshot boundary detection. Section 3 presents the technologies used to remove the irrelevant and repetitive subshots. The method of final summary generation is given in Section 4. The evaluation of our system is analyzed and discussed in Section 5. We close the paper with conclusions and further work.

## 2. SHOT BOUNDARY DETECTION

We select local color histogram as the feature in subshot boundary detection, because of its good performance in color and motion information description and ease in extraction. Each video frame is divided into 4*4 sub-images with same size and shapes. For each sub-image, 16 bins color histogram on HSV color model is extracted according to MPEG-7 [10]. Therefore, each frame is represented by a 256 bins feature vector. Figure 2 shows the feature extraction result of a frame. Figure 2 (a) is the video frame, (b) is the 16 bins histogram of the sub-image in top right corner, (c) is the 256 bins histogram of the whole frame.

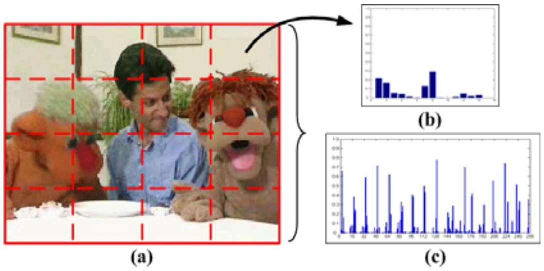


**Figure 2: Local color histogram feature extraction**

Then we use the shot detection approach in [5] to segment the raw rushes data. Though the approach was proposed for shot detection, video subshots can also be detected because the local color histogram is sensitive to the color change and motion information. We compute the difference between frame *i* and frame *j* on local color histogram as follows:

$$D(f_i, f_j) = C_1 \sum_{n=1}^{N} (C_2 \sqrt{\sum_{k=1}^{K} (h_j(n,k) - h_i(n,k))^2}\,) \tag{1}$$

where *n* denotes the *n*th sub-image in frame *i*, *k* denotes the *k*th bin in the histogram of this sub-image, and $h_i(n,k)$ is the value of this bin. *K* is the bin number of color histogram of each sub-image (*K*=16); *N* is the sub-image number of each frame (*N*=16). $C_1$, $C_2$ are two constants for normalization ($C_1$=1/16 and $C_2$=1/$\sqrt{2}$).

The accumulative difference between frame *i* and frame *j* is computed as follows:

$$AD(f_i, f_j) = \sum_{k=i}^{j-1} D(f_k, f_{k+1}) \tag{2}$$

As mentioned in [5], two thresholds, $T_D$ and $T_{AD}$, are predefined for the difference and accumulative difference in subshot boundary detection. When $D > T_D$ or $AD > T_{AD}$, the subshot boundary is considered to exist.

## 3. IRRELEVANT AND REPETITIVE SUBSHOT REMOVAL

After segmentation, the raw rushes video is partitioned into a set of subshots. Some subshots are irrelevant to final summary since they cannot provide representative information or impress users. Among the informative subshots, repetitive content, such as retake, is also common in rushes video. Both irrelevant and repetitive subshots may badly influence the quality of final summary, so they should be detected and removed.

### 3.1 Noise Detection and Removal

In general, there are two kinds of irrelevant subshots in rushes video. One is very short subshots which cannot impress the users. The other is subshots without informative contents, such as color bars, pure color frames, and clapper boards. All test rushes videos in TRECVID 2008 include at least one kind of irrelevant subshots. The percents of irrelevant subshots in duration to the entire videos vary from 2% to 33%.

Figure 3 illustrate the irrelevant subshot detection procedure. First the subshots without enough duration (in our system the permitted duration is no less than 10 frames) are removed. Then two features are extracted from visual and audio views respectively. Finally, the relevant and irrelevant video subshots are classified based on the extracted features.

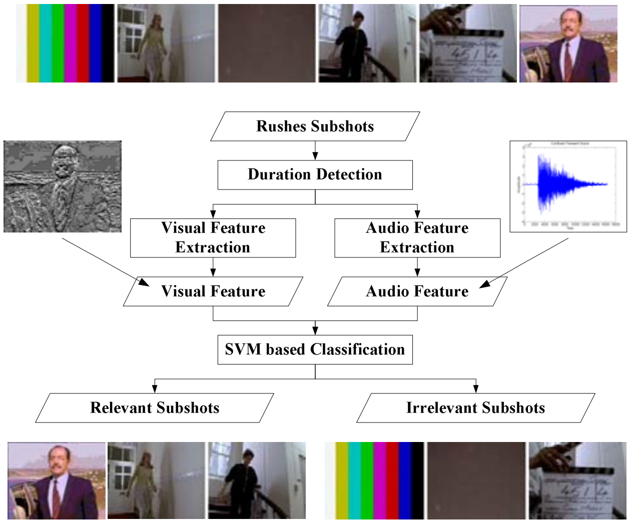


**Figure 3: Flowchart of irrelevant video subshot detection**

#### *3.1.1 Color bar and pure color frame detection*

Color bar and pure color frames are two kinds of irrelevant subshots. Considering their consistence of visual information in vertical direction, we use the sum of gradients on gray scale in vertical direction as the feature:

$$G = \sum_{i=1}^{M} \sum_{j=1}^{N-1} | I(i, j+1) - I(i, j) | \tag{3}$$

where $M$ and $N$ are pixel numbers of each frame in horizontal and vertical directions, respectively.

Figure 4 shows some examples of the gradient in vertical direction on the gray scale (white denotes that the gradient value is 1 and black denotes that the gradient value is 0). Normal frames usually have relatively large gradient values in various places of the image, but the gradient values in color bar or pure color frame are usually very small. A Support Vector Machine (SVM) classifier [9] is used to classify the frames in a subshot. For each subshot, if more than 80% of frames are detected as color bar or pure color frame, it will be considered as an irrelevant subshot.

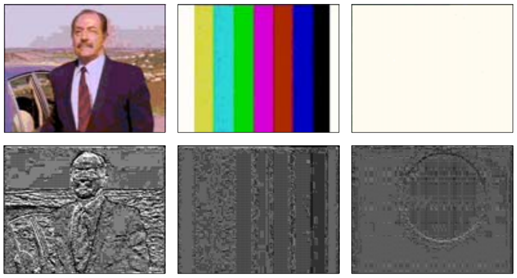


**Figure 4: Sum of gradients on gray scale in vertical direction**

### *3.1.2 Clapper board detection*

In video or film production, the clapper board is a device used to synchronize video and audio, and identify the takes. It occurs far more frequently than color bars and pure color frames. Unfortunately, recognizing the video subshots with clapper board is relatively hard because of the variety of the boards' modality and its visual similarity with the normal scene. In our system, an audio-based method [6] is used to detect clapper board from segmented subshots, and the flowchart is described in Figure 5.

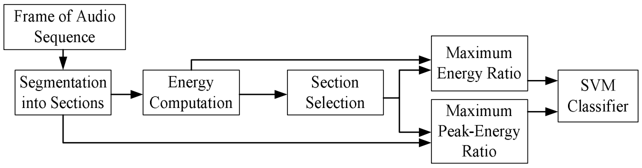
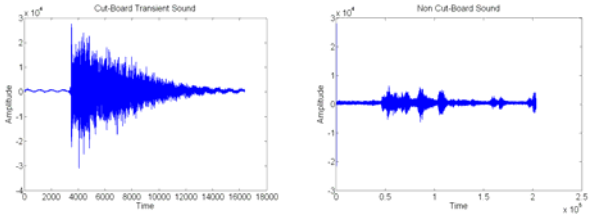


**Figure 5: Flowchart of clapper board detection**

The method is based on an important and unique characteristic lies in the usage of clapper board that the acoustical energy is greatly increased after the action of "knock" (Figure 6).



**(a) Clapper Board** **(b) Non-Clapper-board**

**Figure 6: Audio features of clapper board transient sound and non-clapper-board sound**

To detect the transient sound of clapper board, the sample points are set uniformly with predefined distance in each subshot. Then a slide window including $N$ sampling points is used, and the sampling points are further divided into $M$ parts uniformly ($M$=32 in our experiment). Supposed the sampling points are $x_1, x_2, \ldots, x_N$, and each part in the slide window can be represented as:

$$B_i = \{x_{N_k+1}, x_{N_k+2}, \ldots., x_{N_k+M}\}, \quad N_k = \frac{kN}{M}, \quad k = 0,1,2,\ldots,M-1 \qquad \textbf{(4)}$$

The energy of each part and the whole slide window are computed:

$$E(B_i) = \sum_{x_n \in B_i} x_n^2 \ , \quad E_{total} = \sum_{k=1}^{M} E(B_i) \qquad \textbf{(5)}$$

Two features are extracted based on the energy [8] as follows and the final decision is made by an SVM classifier [9]:

$$\begin{aligned} F_1 &= \max(E(B_i)/E(B_{i-1}), E(B_i)/E(B_{i-2})) \\ F_2 &= \max_{x_n \in B_i}(x_n^2)/E(B_{i-1}) \end{aligned} \qquad \textbf{(6)}$$

The slide window is moved in the detection. Once the decision indicated the presence of clapper board transient, the subshot is determined as an irrelevant subshot with clapper board; if no clapper board transient sound is detected in final, the subshot is determined as one without clapper board.

## 3.2 Keyframe Extraction

After removed irrelevant subshots, all remaining video subshots are considered to be relevant to the final summary. We extract keyframes from the retained subshots.

There have been many keyframe extraction algorithms. In our system, the purpose of keyframe extraction is to describe various states in the vision perception level. So we use an unsupervised clustering algorithm on the local color histogram feature.

First, we compute the "stability" of each frame as follows:

$$S(f_i) = 1 - (D(f_{i-1}, f_i) + Diff(f_i, f_{i+1}))/2 \qquad \textbf{(7)}$$

The frames with local maximum stabilities are selected as the candidate keyframes. Considering two adjacent keyframes $f_i$ and $f_j$, if the difference of any two frames between $f_i$ and $f_j$ (including $f_i$ and $f_j$) on local color histogram feature is less than a predefined threshold $T_S$, the two keyframes are combined as one and the middle frame between them is selected as the new candidate keyframe. The procedure is iterated until no keyframe can be combined, and the computational complexity is $O(M*N^2)$, here $M$, $N$ are the number of subshots and keyframes in each subshot.

Then the frames between any two keyframes $f_i$ and $f_j$ are classified into two classes. The best partition position $p$ is selected to make the sum of differences between each frame to their corresponding keyframe is minimal:

$$p = \arg\min\left(\sum_{k=i+1}^{p} D(f_i, f_k) + \sum_{k=p+1}^{j-1} D(f_k, f_j)\right) \qquad \textbf{(8)}$$

where $f_i$ and $f_j$ are two keyframes, and $f_k$ is a frame between them.

For each cluster, we select the frame which is most similar to the clustering center as the new candidate frame. After several iterations, the final keyframes are generated, and each subshot can be described by a keyframe sequence.

## 3.3 Alignment based Retake Detection

Different with edited video, rushes video usually contains much redundant information because of the repetitive shooting with same contents, which are called "retakes". Obviously, for the same content, only one take should be retained in the final summary and other retakes should be removed.

Since each subshot can be represented as a keyframe sequence, we use a well-known Needleman-Wunsch algorithm [11] to align the keyframe sequences of two adjacent subshots $s_m$ and $s_n$, and determine whether they are matched or partly matched. After alignment, some keyframe pairs in the two subshots are matched, and the similarities between all matched pairs of keyframes are summed. Two scores are computed as follows:

$$score_m = \frac{1}{N_m} \sum_{k=1}^{N_{match}} (1 - D(f_i, f_j))$$
$$score_n = \frac{1}{N_n} \sum_{k=1}^{N_{match}} (1 - D(f_i, f_j)) \qquad \textbf{(9)}$$

where $f_i$ and $f_j$ are a pair of keyframes which are matched, $N_{match}$ is the number of matched keyframes pairs, $N_m$, $N_n$ are the total keyframe numbers in $s_m$ and $s_n$, respectively.

Suppose $N_m \leqq N_n$ and $score_m \geqq score_n$. For a predefined threshold $T_M$, if $score_n$ is larger than $T_M$, it means that $s_m$ and $s_n$ are matched; if $score_m$ is larger than $T_M$, but $score_n$ is not, it means that $s_m$ and $s_n$ are partly matched and $s_m$ is a part retake of $s_n$; if both $score_m$ and $score_n$ are less than $T_M$, it means that $s_m$ is different to $s_n$.

Based on the definition of "match", the repetitive subshots are detected. First, we cluster the subshots as following:

***Step1***: mark each subshot as "unprocessed";
***Step2***: select the first unprocessed subshot and create a new cluster as "current cluster", if no unprocessed subshot is found, go to Step 5;
***Step3***: for each unprocessed subshot behind the selected subshot, detect whether it is matched or partly matched with all the subshots in current cluster; if so, add it into current cluster and mark it as "processed";
***Step4***: repeat Step 2 and Step 3;
***Step5***: finish.

Using above algorithm, we can establish relativities among subshots as shown in Figure 7. The subshots connected with same color arcs are in the same cluster.

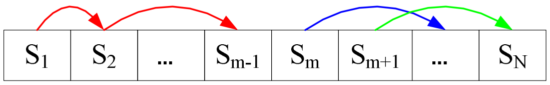


**Figure 7: Subshot clustering result**

Based on the result of subshot clustering, we detect the retakes in rushes. Figure 8 shows an example of retake detection. The video sequence contains 10 subshots (marked as 1, 2, …, 10). The matrix denotes the subshot clustering result, that is, if two subshots are in the same cluster, the color of corresponding block is white; otherwise it is gray. First, the subshots (subshot 1, 3, 6) in the same cluster with the first subshot (subshot 1) are marked (with red boxes). So the original video sequence can be divided into three sub-sequences: {1, 2}, {3, 4, 5} and {6, 7, 8, 9, 10}. Compare each sub-sequence to its successive one, if the current sub-sequence is equal to the whole or the fore part of the successive one, remove it; if the successive sub-sequence is a part of current one, remove the successive sub-sequence and compare the current sub-sequence to the next one if exist; otherwise, retain current sub-sequence and remove the repetitive part in the successive one. Repeat the procedure in the unprocessed part in the last sub-sequence (separated from processed part with blue lines) till all subshots are processed. The method can obtain the largest information inclusion and keep the subshot order as the original video.

In this example, compare sub-sequence {1, 2} with {3, 4, 5} first, and find {1, 2} is equal to the fore part ({3, 4}) of {3, 4, 5}, so remove {1, 2}; then compare {3, 4, 5} to {6, 7, 8, 9, 10} and find they have repetitive parts, so retain {3, 4, 5} and remove {6, 7} in {6,7,8,9,10}. In the unprocessed part ({8, 9, 10}) of last sub-sequence, repeat the above procedure. Finally, subshot 3, 4, 5, 9, 10 are retained.

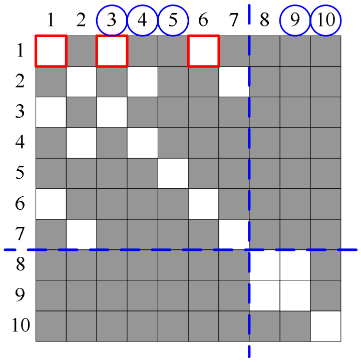


**Figure 8: Example of retake detection**

# 4. FINAL SUMMARY GENERATION

After irrelevant and redundant content removal, the remaining subshots are representative and informative. We generate the final summary based on the extension of keyframes in the retained subshots.

According to the research in [7], a scene needs 3.25 seconds to completely analyze by a normal person. However, considering the special review approach in evaluation (the reviewers can use "Pause" button), the minimum duration of a video clip can be shortened. In our system, the duration of a video clip is determined as being no less than one second (25 frames) and may be longer if permission:

$$r = \max(r_0, N_{total} / N_{kf}) \qquad \textbf{(10)}$$

where $r_0$ is the rate of original rushes video, $N_{total}$ is the total number of original rushes and $N_{kf}$ is the number of keyframe in remaining subshots. Experiments show that the extension rate is usually enough for summary browsing. And an additional advantage of uniform rate extension is that the summary result usually has a pleasant rhythm.

## 5. RESULTS

There are eight evaluation criterions for TRECVID 2008 rushes summarization task: ***DU*** - duration of the summary (secs.); ***XD*** - difference between target and actual summary size (target-actual) (secs.); ***TT*** - total time spent judging the inclusions (secs.); ***VT*** - total video play time (versus pause) judging the inclusions (secs.); ***IN*** - fraction of inclusions found in the summary (0 - 1); ***JU*** - Summary contained lots of junk: 1 strongly agree - 5 (best) strongly disagree; ***RE*** - Summary contained lots of duplicate video: 1 strongly agree - 5 (best) strongly disagree; ***TE*** - Summary had a pleasant tempo/rhythm: 1 strongly disagree - 5 (best) strongly agree. For DU, TT, and VT scores, the lower scores are, the better performance is for the summary length. For XD, IN, JU, RE, and TE scores, the higher scores are, the better performance is for summary content.

Table 1 shows the evaluation results of our system on these eight criterions. For all eight criterions, our results are better than the mean and median results of 43 submissions. It means that our summaries contain the main contents of the original rushes video with acceptable duration. More importantly, our JU score ranks 4th in all submissions. It means that the algorithms for irrelevant content detection in our system are very effective. The TE score of our system ranks 9th in all submissions. This means that our summaries have a pleasant rhythm and are easy to understand.

**Table 1. Our results on TRECVID 2008 rushes summarization task**

| Criterions | DU | XD | TT | VT | IN | JU | RE | TE |
|---|---|---|---|---|---|---|---|---|
| Baseline | 31.31 | 0.40 | 59.59 | 31.36 | **0.83** | 2.66 | 2.02 | 1.44 |
| Mean of 43 submissions | 27.10 | 4.60 | 41.20 | 29.40 | 0.44 | 3.16 | 3.27 | 2.73 |
| Median of 43 submissions | 28.11 | 3.60 | 41.41 | 30.27 | 0.45 | 3.11 | 3.37 | 2.80 |
| **Our results** | **26.11** | **5.60** | **37.70** | **28.32** | 0.47 | **3.56** | **3.48** | **3.21** |

Table 2 compares our results in TRECVID 2007 and TRECVID 2008 rushes summarization task. The DU, TT, VT, and IN results are improved in this year. However, the RE result is worse than it in last year. EA, which means easy to understand the summary (1 strongly disagree - 5 strongly agree), is a criterion used in last year. In this year, another similar criterion, TE, is used to measure the understandable extent of the summary. In this criterion, our current system also achieve enhancement compared to last year. In another criterion, XD, this year's performance looks worse than last year's. But actually, this criterion is computed by *target summary size - actual summary size*. The *target summary size* of last year is much larger than that of this year. This is the reason why we have a larger XD value this year.

It is worth to note that all the improvements are achieved based on the more strict duration constraint in this year. This fact proves the effectiveness of our current system.

**Table 2. Comparison between our results in TRECVID 2007 and TRECVID 2008**

| Criterions | DU | XD | TT | VT | IN | RE | TE/EA |
|---|---|---|---|---|---|---|---|
| Our results in TRECVID 07 | 31.37 | 28.50 | 62.17 | 33.33 | 0.39 | **3.83** | 3.12 |
| Our results in TRECVID 08 | **26.11** | 5.60 | **37.70** | **28.32** | **0.47** | 3.48 | **3.21** |

## 6. CONCLUSION AND FUTURE WORK

In this paper, we describe our system designed for BBC rushes summarization task of TRECVID 2008. We utilize different technologies to generate the summary, such as color histogram based shot detection, visual and audio based irrelevant content detection, and alignment based retake detection. Evaluation results distributed by TRECVID demonstrate that our summaries achieve better performance on all eight evaluation criterions compared with the mean and the median of all submissions, especially for the conciseness and pleasant rhythm.

Future work will be explored from the following three aspects. First, we want to explore how to select more suitable metric for keyframe sequence alignment because it significantly influences the performance of repetitive content detection. Second, how to make the automated content selection consistent with human's attention and interesting is still a promising topic and worthy for further investigation. Third, we intend to test our system on some other video datasets, and extend current system to a more general framework for video summarization.

## 7. REFERENCES

[1] Takao, S., Haru, T., and Ariki, Y. Summarization of News Speech with Unknown Topic Boundary. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. pp: 615-618, 2001.

[2] Ekin, A., Tekalp, A.M., and Mehrotra, R. Automatic Soccer Video Analysis and Summarization. *IEEE Transactions on Image Processing*. vol. 12, no. 7, pp. 796–807, July 2003.

[3] Choudary, C. and Liu. T.-C. Summarization of Visual Content in Instructional Videos. *IEEE Transactions on Multimedia*. vol. 9, no. 7, pp. 1443-1455, Nov. 2007.

[4] Over, P., Smeaton, A.F., and Kelly, P. The TRECVID 2008 BBC rushes summarization evaluation. In *Proceedings of the International Workshop on TRECVID Video Summarization* (*TVS'08*), Vancouver, British Columbia, Canada, October 31, 2008, ACM Press, New York, NY, pp. 1-20, 2008.

[5] Zhang, H.J., Kankanhalli, A., and Smoliar, S.W. Automatic Partitioning of Full-Motion Video. *Journal of Multimedia Systems* 1, 10-28. 1993.

[6] Liu, Y., Liu, Y., and Zhang, Y. The Hong Kong Polytechnic University at TRECVID 2007 BBC Rushes Summarization. In *Proceedings of the International Workshop on TRECVID Video Summarization* (*TVS'07*), Augsburg, Bavaria, Germany, September 28, 2007, ACM Press, New York, NY, 2007.

[7] Pfeiffer, S., Lienhart, R., Fischer, S., and Effelsberg, W. Abstracting Digital Movies Automatically. *Journal of Visual Communication and Image Representation*, 7, 4, pp: 345-353. Dec. 1996.

[8] Yan, J. X, Dou, W. B., and Dong, Z. W. Time-Domain Detection Method of Transient Signals in Audio Coding. *Journal of Electronics& Information Technology*, vol. 28, no. 2, Feb. 2006.

[9] Chang, C.C. and Lin, C.J. LIBSVM: a library for support vector machines, 2001.

[10] Manjunath, B. S., Ohm, J.R., Vasudevan, V.V., and Yamada, A. Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 703-715, 2001.

[11] Needleman, S. and Wunsch, C. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443-453, 1970.