Dokuz Eylül University Video Shot Boundary Detection at TRECVID 2006

Taner Danisman, Adil Alpkocak

Computer Engineering Department
Dokuz Eylül University, Tinaztepe Campus, 35160 Buca, Izmir, Turkey
{taner,alpkocak@cs.deu.edu.tr}

Abstract

In this study we described our participation in the NIST TRECVID 2006 evaluation for Shot Boundary Determination task with 10 submissions. We have used a method based on histogram differences for cut and gradual transition detection. Both the details of our approach used in the SBD task submissions and effects of using different threshold & skip frame intervals on evaluation results are presented.

Keywords: Shot Detection, Cut/Gradual Transition Detection

1. Introduction

Indexing and retrieval of video becomes more and more important as the size of video data increases. In addition to the video by TV broadcasters, nowadays all low cost modern digital still cameras, even cellular phones have the capability of capturing and storing moving pictures. As a result of wide spread use of these devices, there is an emerging need for efficient accessing those huge amount of video data produced.

The importance of shots in video indexing and retrieval similar like importance of words in textual indexing and retrieval methods, thus shot locations and types gives important clues about the video itself. There are many methods in literature to find shot regions [1][2][3][4][5][6][7].

This year we have participated in TRECVID for the first time for shot boundary determination task. Therefore we use simple approaches (*histogram differences*) with a small temporal modification on processing style as a starting point and we have submitted 10 runs each using the same system with different parameters applied.

Section two presents detail of our algorithm for shot boundary determination task including hard cut and cross-fade detections, then we described the evaluations of our submissions in terms of system parameters (simply thresholds, T_C, T_G and T_{SFI}) and finally conclusions are presented.

2. Shot Boundary Determination Task

SBD is a process to identify the boundaries of shots from a sequence of video frames, where a shot is the smallest meaningful part of video. In video processing, SBD appears at the very early phase of the video processing. Furthermore, SBD is a must for efficient access of desired information in huge amount of video resources which requires a set of difficult and usually CPU intensive tasks.

Our approach for SBD task is based on color histogram differences in RGB color space for both cut and gradual transition detection. This method uses a threshold value for cut detection and a *skip frame interval* value for eliminating consecutive frames that have much redundant information (*TRECVID dataset and test videos are all MPEG files with 29,97fps ratio*) for faster processing. Instead of processing all consecutive frames we skipped frames by a factor of *skip frame interval* value. The values used in our experiments are changed from 1 to 5 frames and it is clear that it dramatically reduces the computation time with a relatively small change in the precision.

Histograms are one of the most commonly used methods to detect shot boundaries within video data. In our approach, histograms of pixels in RGB color space for each frame are used to detect shot boundaries. First, we have quantized RGB color space into 27 equal sub-spaces by dividing each axis of color space into three equal parts. In another say, each sub-space is a cube with a size of 85. Shortly, this has resulted a 27 bin feature vector which is easy to compute and process further. More formally, the histogram, H, can be defined as follows: where R_i, G_i, B_i, represents the ith pixel values for red green and blue channels, respectively.

$(\forall R_i, \forall G_i, \forall B_i)(r,g,b \ni \{1,2,3\}, p = 255 \div 3)$ and $H(r,g,b)=H(r,g,b)+1$

where $(r=R_i \div p) \& (g=G_i \div p) \& \ (b=B_i \div p)$

Finally, computing the color histogram forms a feature vector with a size of 27. Then, Euclidean distances of histograms belonging to two successive frames are calculated for the cut and gradual transition detection. If any frame by frame difference is greater then the maximum allowed threshold value, T_C, then a cut is said to be detected. Formally, if there are n frames with a size of M×N and let $H_i(j)$ be the histogram value of jth bin of the ith frame. Then the difference between the ith and $(i+1)$th frame can be defined as follows:

$$D_i = \sqrt{\sum_{j=1}^{m} \left(H_i(j) - H_{i+1}(j)\right)^2}$$

If the D_i value (Euclidean distance) of these histograms exceeds the threshold T_C then a cut is detected. It assumes that the background information does not change either so frequently or strongly among the boundaries of a shot region. Assuming that the number of pixel belongs to background is dominating.

Our algorithm is based on the color histogram differences of two frames. It has three different thresholds. One for detection of abrupt changes T_C (for Cut Detection), the other is for detection of gradual transitions especially cross-fade effect T_G (for Gradual Transition Detection) and the last one T_{SFI} is to decrease temporal redundancy in video frames.

Consecutive frames in video usually have similar spatial information. If we skip only one frame then total number of comparisons is divided in half. Therefore we use skip frame interval threshold T_{SFI} for better computing performance. We set the T_{SFI} value to 1 and 5 in our experiments. But our algorithm does not simply compare every T_{SFI}th frames. It can compare frames in both directions (backward and foreword) for better frame accuracy. Whenever it finds a distance that exceeds the T_C value it turns back to the next frame of the previously compared frame and then continues its step by step comparison without considering T_{SFI} value. Pseudo-code is as follow;

```
While frames exists
   Compute Euclidean distance of current frame $F_i$ to target frame $F_{i+T_{SFI}}$
   If distance exceeds $T_C$ then
      If it's a first time shot then set target frame to $F_{i+1-T_{SFI}}$
      Else
            set shot.startFrame=i , shot.endFrame= TargetFrame
            If TargetFrame-i smaller than $T_G$ then mark this shot as 'SubGradual'
            else mark this shot as 'Cut'
            Set current frame with target frame
            Add $T_{SFI}$ to target frame
   Else
      If it is not a first time shot then add $T_{SFI}$ to target frame
      Else increase the target frame.
```

A good threshold value T_C should be selected for best accuracy on hard cuts and skip frame interval must be long enough to get better computing performance by eliminating the temporal dimension of video.

Gradual transitions are detected on a second pass by computing the length of the consecutive cuts. We have used a second threshold that holds the minimum number of frames that a shot holds. If a shot region has frames less then the threshold T_G then it is marked as a gradual transition. The value of T_G is fixed to 10 frames in our experiments.

It is clear that selection of threshold values T_C and T_G critically effects the number of shots can be detected by the system. If T_C has a lower value, then number of detected shot increases, otherwise decreases. If T_G has a lower value then number of detected gradual transitions decreases, otherwise it increases.

T_G is the minimum number of frames that a shot should holds. If a shot has less then T_G number of frames then it is marked as a SubGradual. Another say, shots that has short durations (compared with T_G value) are set as gradual transitions. Minimum and maximum boundaries of T_G can be set between 10-60 frames which are the most commonly used average gradual cross-fade transition length. Its value is linearly proportional with the number of shots detected as cross-fade. Smaller T_G value also decreases the number of detected cross-fades. We can define the gradual cross-fades more formally;

Let T_G is the threshold value for detecting the cross-fade effect.

S_i represents the ith shot L_{S_i} and represents the shot length in terms of # of frames.

S_{CF_j} is a sub cross-fade region whose $L_{Si} \leq T_G$ and $CF_{i,j}$ is the cross-fade between the ith and jth frames then

$$S_{CF_i} = \begin{cases} Cut & L_{S_i} \rangle T_G \\ SubGradual & L_{S_i} \leq T_G \end{cases} \text{ and } CF_{i,j} = \begin{cases} Gradual & (\exists i, \exists j) S_{CF_i}, S_{CF_j} \in \{SubGradual\}, j = i+1 \\ null & else \end{cases}$$

Figure 1 shows cross-fade detection example. In the first line there is a fade out, last line shows a fade-in effect and they produce the middle line the actual dissolve effect.

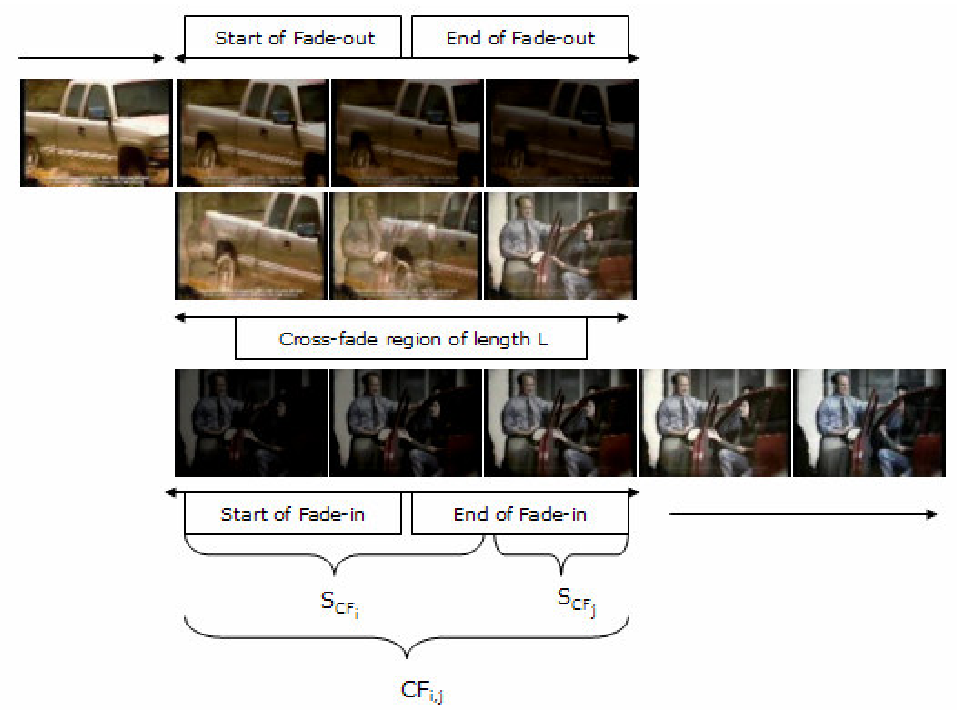

Figure 1. Example cross-fade detection

3. Run Overview and Evaluations

We have used TRECVID 2006 dataset and master shot reference provided by [8] in our experiments and we have submitted 10 runs on test set each having different thresholds for frame by frame difference value T_C and skip frame interval value T_{SFI}. Each sysId has a self-descriptive naming convention. The first number in the sysId is the threshold value, T_C, and the last one is the T_{SFI} value.

Table 1 and Table 2 shows the parameters used in each submission and the TRECVID 2006 shot boundary determination results of DEU, respectively.

Table 1. Summary of each DEU run

System ID (runID)	T_C	T_G	T_{SFI}
EU_1150_1	1150	10	1
EU_1150_5	1150	10	5
EU_1250_1	1250	10	1
EU_1250_5	1250	10	5
EU_550_1	550	10	1
EU_550_5	550	10	5
EU_750_1	750	10	1
EU_750_5	750	10	5
EU_950_1	950	10	1
EU_950_5	950	10	5

Table 2. TRECVID 2006 shot boundary determination results

	ALL		CUTS		GRADUAL			
							Frame	
sysid	Recall	Prec	Recall	Prec	Recall	Prec	Recall	Prec
EU_1150_1	0.273	0.160	0.324	0.147	0.138	0.383	0.397	0.771
EU_1150_5	0.267	0.168	0.320	0.154	0.123	0.423	0.378	0.790
EU_1250_1	0.260	0.167	0.307	0.152	0.131	0.419	0.393	0.763
EU_1250_5	0.249	0.171	0.298	0.157	0.115	0.463	0.364	0.771
EU_550_1	0.425	0.143	0.446	0.122	0.367	0.322	0.523	0.729
EU_550_5	0.422	0.147	0.445	0.126	0.359	0.343	0.502	0.756
EU_750_1	0.372	0.153	0.414	0.136	0.260	0.330	0.471	0.768
EU_750_5	0.362	0.156	0.405	0.139	0.244	0.354	0.441	0.806
EU_950_1	0.329	0.163	0.377	0.147	0.199	0.354	0.405	0.795
EU_950_5	0.319	0.167	0.367	0.151	0.190	0.387	0.395	0.817

According to the results we gained a low performance on both gradual and cut detection. Although gradual transition detection problem is more complex than cut detection problem, precision of gradual detection is quite better than precision on cuts because of the low T_G value.

Results showed that selecting a low T_C value leads to better cut detection result as in case **EU_550_1** and **EU_550_5**. But at the same time it makes the lowest frame precision value on gradual transition detection.

We made our experiments on a single Pentium IV 3Ghz processor having 1GB of memory. Our timing results shows that *skip frame interval* T_{SFI} value decreases the total processing time as shown in Table 3. However the most significant time belongs to decoding phase. Effect of T_{SFI} dominates on feature extraction time. In this table; *decoding time* refers to the time takes to decode each mpeg file to corresponding jpeg files. The *segmentation Time* is the time to compute feature vector and time to find transition type. And *total run time* is the sum of the Decoding Time and Segmentation time.

Table 3. Timing results in seconds for TRECVID 2006 test set.

Run Id	Decode Time (A)	Segmentation Time			Total Run Time A+B+C
		Feature Extraction Time (B)	Transition Detection Time (C)	Total Segmentation Time (B+C)	
EU_550_1	20274.73	15479.93	286.82	15766.75	36041.48
EU_750_1	20274.73	15479.92	276.25	15756.17	36030.90
EU_950_1	20274.73	15479.92	274.12	15754.04	36028.78
EU_1150_1	20274.73	15479.92	269.21	15749.13	36023.87
EU_1250_1	20274.73	15479.92	271.89	15751.81	36026.54
EU_550_5	20274.73	3382.27	244.82	3627.09	23901.82
EU_750_5	20274.73	3336.96	238.26	3575.22	23849.95
EU_950_5	20274.73	3286.78	250.54	3537.32	23812.06
EU_1150_5	20274.73	3112.32	223.22	3335.54	23610.28
EU_1250_5	20274.73	3329.27	253.51	3582.78	23857.51

4. Conclusions

We have presented our approach to shot boundary determination problem. It is clear that use of T_{SFI} decreases the total segmentation time with a small change in detection performance. Our gradual detection performance is quite better than our cut detection performance because of the low T_{G} value.

Main problem in our model is to find a universal threshold T_{C} and T_G to address all types of video having different characteristics. Adaptive thresholds can solve this problem. Another problem in this area is varying frame numbering problem of different decoders on same video document. During our experiments frames decoded by mplayer [9] produced different frame numbers than frames in master shot reference.

Finally in our experiments decode time has the majority over others. After extracting the feature vectors, it takes about 258 seconds in average to complete the segmentation of whole test set having 13 videos and total of 597.043 frames.

5. References

1. R. Lienhart, "Reliable Transition Detection in Videos: A Survey and Practitioner's Guide", *International Journal of Image and Graphics (IJIG)*, Vol.1, No.3, pp.469-486, 2001
2. U. Gargi, R. Kasturi, and S. H. Strayer. Performance characterization of video-shot-change detection methods. *Circuits and Systems for Video Technology*, IEEE Transactions on, 10(1):1, 2000.
3. B. T. Truong, C. Dorai, and S. Venkatesh. New enhancements to cut, fade, and dissolve detection processes in video segmentation. *In Proceedings of the 8th ACM International Conference on Multimedia*, pages 219--227, 2000
4. Zhang HJ, Kankanhalli A, Smoliar SW. Automatic Partitioning of Full Motion Video. *Multimedia Systems* vol 1, 10-28, Jan 1993
5. R. Lienhart. Comparison of Automatic Shot Boundary Detection Algorithms. *SPIE Storage and Retrieval for Still Image and Video Databases VII 1999*, Vol. 3656, pp. 290-301, Jan. 1999.
6. R. Lienhart, “Reliable Dissolve Detection”, *In Storage and Retrieval for Media Databases 2001*, Proc. SPIE 4315, pp. 219-230, Jan. 2001
7. M. Albanese, A. Chianese, V. Moscato, L. Sansone: "A Formal Model for Video Shot Segmentation and its Application via Animate Vision", *Multimedia Tools and Applications*, Volume 24, N.3, pp. 253-272, December 2004
8. C. Petersohn. "Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System", *TREC Video Retrieval Evaluation Online Proceedings*, TRECVID, 2004
9. A. Gereffy, “mplayer tool,” http://www.mplayerhq.hu, May 2005, Version 1.0