AN EFFICIENT SHOT DETECTION METHOD FOR VIDEO SUMMARIZATION

Fehmi Chebil, Jun Yu and Asad Islam

Nokia Research Center, USA {fehmi.chebil, jun.yu, asad.islam}@nokia.com

ABSTRACT

We present two techniques for abrupt scene cut and gradual scene change detections in videos. The methods operate on the compressed video bitstreams thus provide a significant complexity reduction compared to spatial domain approaches. The proposed techniques are based on new measures exploiting histogram differences of the video frames and on information extracted from the compressed codestreams. We also suggest a technique for selecting key frames from the identified video scenes. Our experimental results show that these techniques lead to very good performance.

1. INTRODUCTION

The advances in video capturing, processing, displaying and sharing techniques result in a tremendous amount of video data available for users. Navigating through this content for browsing or retrieving a video sequence becomes difficult and time consuming. Abstraction or summarization of video clips as suggested in [1] helps significantly in overcoming this problem. Shot detection, i.e. the segmentation of video into separate shots, or scenes, is usually the first step for video summarization. Key frames can then be selected from these shots to generate a summary of the video data. It is therefore important to develop efficient methods for shot detection.

Conventional approaches such as those described in [2-3] perform shot detection in the pixel domain, known also as the spatial domain. Since video sequences are saved and shared in compressed form, these techniques require decoding of the video clips. For several platforms, such as the mobile, decoding videos and processing them in their raw format is very costly, and sometimes not feasible, due to the limited resources in computational power and in memory. It is therefore crucial when targeting such applications to utilize techniques that detect shots in the compressed domain.

There are several techniques in the literature that proposed compressed domain methods for scene analysis. Shen and Delp in [4] suggested a technique for scene change detection based on the color histogram difference of the reconstructed DC coefficients of the frames in the video clips. Yeo et al in [5] presented a method, which uses the color histogram difference in a sliding window of the approximated DC images of the frames to identify scene cuts and gradual scene changes.

In this paper, a temporal segmentation technique for video abstraction operating in the compressed domain is presented. In our method we use the approximated DC images of the video sequences. In addition, we utilize the information extracted from the compressed codestream about motion vectors and macro block types. In order to use our technique on all profiles and levels of H.263 and MPEG-4 standards, we do not employ any information about Bi-directional (B) frames. To efficiently detect abrupt scene cuts and gradual scene changes we define and use two distinct modified histogram differences. The developed technique yield to near optimal performance when tested on H.263 and MPEG-4 encoded video. The rest of the paper is organized as follows: In Section 2, we introduce our scene cut detection method. In section 3, the suggested gradual scene change detection technique is presented. In section 4 we propose a method for key frame selection and we present our experimental results in Section 5.

2. ABRUPT SCENE CHANGES (SCENE CUTS)

DC image refers to the image formed by only using DC coefficients of the DCT transformed original frame. Let $P(x, y)$ be a frame having a width W and a height H, its DC image is obtained by:

$$DC(i,j) = \frac{1}{64}\sum_{m=8i}^{8i+7}\sum_{n=8j}^{8j+7} P(m,n), 0 \le i \le \frac{W}{8} - 1, 0 \le j \le \frac{H}{8} - 1 \quad (1)$$

Computing the DC image for I frames is calculated by simply extracting the DC values of all the 8 by 8 DCT blocks. However, for the predicted frames, P frames, the extraction of exact DC values necessitates the full decoding of the bitstream. To avoid full decoding, the DC approximation method suggested in [5] is used to obtain DC coefficients of P frames using the DC coefficients from the previous I or P frames.

When the DC frames for every frame are retrieved, we apply our shot detection technique. The goal is to identify

the boundaries, i.e. the frames at which a change of scene occurs. There are two kinds of shot boundaries: an abrupt one and a gradual one. A Scene cut happens when the scene change takes place at one frame. A gradual scene refers to all other types of scene changes. In this paper we treat these two types separately.

To identify abrupt scene changes, we define $MHD(k,k+1)$, an RGB color histogram difference between two consecutive DC images k and $k+1$ as:

$$MHD(k,k+1) = \sum_{i=0}^{G-1} |H_k^R(i) - H_{k+1}^R(i)| + |H_k^G(i) - H_{k+1}^G(i)| + |H_k^B(i) - H_{k+1}^B(i)| \quad (2)$$

where $H_k^R(i)$, $H_k^G(i)$ and $H_k^B(i)$ denote the modified R, G, B histograms for DC image k, respectively and G refers to the number of bins for the histogram. For I frames, the histogram is computed using all the macroblocks. For P frames, only the change blocks are considered to count the histogram. We define a changed block as a block that is either coded in Intra mode or coded in Inter mode and with non-zero motion vectors. Since the compressed data in video standards represents usually the luminance and chrominance components, conversion of the DC images to the RGB color space is performed.

A sliding window is then applied on these histogram values $MHD(k,k+1)$ to detect the scene cut based on the local peaks in the window. The peak in the histogram differences has to be n times larger than any other histogram difference within the window. Let $2W-1$ be the window size, representing the number of frames. Frame k is a scene cut if $MHD(k,k+1) \geq nMHD(l,l+1)$ for $k-W+1 \leq l \leq k+W-1$ *and* $l \neq k$. This approach was suggested in [5] and a constant value of n was employed since all the blocks in the frames were used to compute the histograms, in our approach however, we only use changed blocks in the predicted frames. We, therefore, have to "normalize" our comparison ratio n. Let N_k denote the number of changed blocks in frame k, the average number of changed blocks in the window of frames of $2W-1$ around frame k is

$$AN_k = \frac{1}{2W-1} \sum_{l=k-W+1}^{l=k+W-1} N_l \quad (5)$$

The comparison ratio we utilized is $n = k\frac{AN_k}{NI_l}, k-W+1 \leq l \leq k+W-1$ *and* $l \neq k, N_l \neq 0$.

In our experiments, choosing $k=2$ leads to very good results.

To further improve the result and reduce false detections, we need to refine the set of scene cut frames by applying a validation test. Typically, if there is a scene cut between frames k and $k+1$, then most blocks in frame $k+1$ will be very different from those in frame k. In such a situation, encoders will either code frame $k+1$ as Intra or the frame will include many changed blocks. If the frame $k+1$ is P coded, then let N_{k+1} denote the number of its changed blocks and let NS be the total number of blocks in the frame. We define R_{k+1} as the ratio of changed blocks in the bitstream for any frame:

$$R_{k+1} = \frac{N_{k+1}}{NS} \quad (5)$$

We apply the following test on the potential scene cut frames:

$$\text{If} \begin{cases} R_{k+1} > T, \text{ then frame k is } \textit{true scene cut} \\ \textit{else, it} \text{ is not a } \textit{scene cut} \end{cases}, \quad (6)$$

where T is a threshold. Fernando et al [6] used such a threshold for Intra coded macro blocks to detect scene cut detections. Relying on this measure by itself makes the algorithm dependent on the behavior of the encoder used to compress the video. We found that utilizing this measure reduces false alarms significantly.

3. GRADUAL SCENE CHANGES (GSC)

$MHD(k,k+1)$ in equation (3) can be viewed as the absolute value of the first order derivative of the histogram values. We try to allocate a local maximum of this derivative that is greater than a certain threshold to detect abrupt changes.

In gradual change, we propose to utilize a form of a second derivative of the histogram difference. For detecting a gradual scene change in a video sequence starting at a frame i and lasting for a set of frames GS, we relate GS to the video's frame rate. For a 10 frames per second video for example, GS can take a value between 5 and 15. We define our measure $GSC(i)$ for detecting gradual scene changes around a frame i as:

$$GSC(i) = \frac{\sum_{j=i}^{i+GS-1} |MHD(j,j+1) - MHD(j+1,j+2)|}{EHD(i,i+GS+1)} \quad (7)$$

Where $EHD(i,i+GS+1)$ is the histogram difference between frame i and $i+GS+1$, the histograms are

computed for the entire approximated DC frames. It is used as a normalization factor for our measure. If a frame was identified as a scene cut point in the GS frames, it is not included in the computation of $GSC(i)$.

After computing $GSC(i)$, the entropic thresholding method described in [2] is applied to obtain an automatic threshold T_{GSC}. For any frame i, if $GSC(i) > T_{GSC}$, then it is a GSC frame.

Finally, post processing of the determined GSC frames is performed to merge successive short duration gradual scene changes. This is a similar procedure to the post-processing in image segmentation, in which over-segmented objects of small sizes are merged.

4. KEY FRAME SELECTION

After the temporal segmentation of the video into scenes, key frames are extracted. The procedure we applied consists of:

1. The first frame of every shot is selected as a key frame.
2. If the same shot is followed by a scene cut, then the last frame is also selected as a key frame.
3. For scenes which we identified as gradual scene changes, the first frame is selected as a potential key frame. It is validated if the following condition is satisfied:

$$i \text{ is a } \begin{cases} key\ frame, if\ EHD(i,j) > T' \\ not\ key\ frame, otherwise \end{cases} \quad (8)$$

where j is the previously identified key frame and T' is the median value of histogram differences computed for the frames between i and j.

Other methods could be also utilized to further tune the selected key frames. Such techniques may be based on features extracted such as described in [7].

5. EXPERIMENTAL RESULTS

We applied our proposed algorithms on several video sequences. The length of the videos varies from several hundreds of frames to thousands of frames. As we are targeting mobile phone based applications, the frame rates of the videos tested were 10, 15 and 20 frames/s and the resolutions were subQCIF (128x96) and QCIF (176x144). In our experiment we used the threshold T in equation (6) as 0.5 and the GS length is selected to be 10.

Table 1 gives some shot change detection results. The column "Scene Cut" shows the number of true scene cuts in each sequence. The cuts were identified by several viewers. The video sequences were shown to the observers at a rate of 1 frame per second. A video player application showing the same video in two different views with one frame delay was used to help identifying the scene cuts. "Detected Cut" gives the number of correctly detected scene cuts by the proposed algorithm. "False Cut" shows the number of incorrectly detected scene cuts using the proposed technique. "Missed Cut" means there was a scene cut but the algorithm failed to detect it. Recall and Precision are two widely utilized metrics to represent the missed detections and the false alarms. They are obtained as:

$$\text{Re}call = \frac{Detects}{Detects + MissedDetects}$$

$$\text{Pr}ecision = \frac{Detects}{Detects + FalseAlarms}$$

In [8] the performance of several shot detection methods were evaluated. The authors classified the methods as color histogram based, MPEG based and block matching based techniques. Our technique does not fall particularly in any of these categories but uses features from all of them. In detecting scene cuts, the performance of our algorithm is very close to spatial domain approaches based on the usage of the RGB color histogram difference, but with a significantly less computational complexity, as we do not decode the video. Table 2 shows the performance of an exact DC based RGB color histogram difference for the sequences listed in Table 1.

For GSC, the same notations for reporting scene cuts are utilized. The same procedure used for determining scene cuts was also employed for gradual scene changes. It is reported in [8] that most techniques fail in GSC detection. Our proposed algorithm shows a very good performance as reported in table 3.

Shot detection is used as a first step in video summarizations. The extracted key frames according to the described algorithm are encoded to form an abstract of the video. The length and the size of the desired video summary would determine the number of key frames to be include, the frame rate and the bit rate.

Besides summarization, key frame extractions helps in generating a "Table of Contents (ToC)" for videos. The

Clip	Length	Scene Cut	Detected Cut	False Cut	Missed Cut	Recall %	Precision %
1	2261	11	13	2	0	100	85
2	1100	13	14	1	0	100	93
3	428	16	16	0	0	100	100
4	3020	21	23	3	1	95	87

Table 1 Scene change detection results

Clip	Length	Scene Cut	Detected Cut	False Cut	Missed Cut	Recall %	Precision %
1	2261	11	11	1	1	91	91
2	1100	13	13	0	0	100	100
3	428	16	14	0	0	88	100
4	3020	21	21	1	2	91	91

Table 2 Scene cut detection results by exact DC

Clip	Length	GSC	Detected GSC	False GSC	Missed GSC	Recall %	Precision %
1	2261	13	14	2	1	93	87
2	1100	7	9	2	0	100	81
3	428	7	11	4	0	100	75
4	3020	21	24	3	0	100	88

Table 3 Gradual Scene change detection results

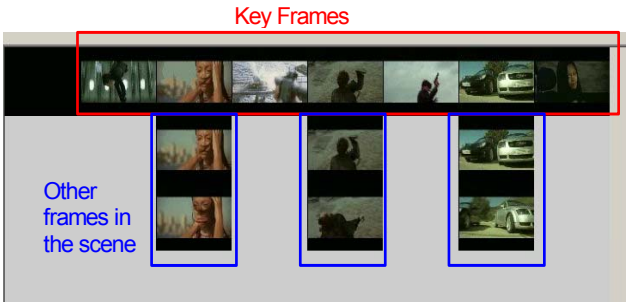

Figure 1 Table Of Content of a Video Clip

ToC allows users to navigate, browse and access the content of long videos. This could be achieved by presenting thumbnails, down-sampled resolution, of key frames as shown in figure 1.

6. CONCLUSION

In this paper, we have proposed a compressed domain method for scene cut and gradual scene change detections. The algorithm is very simple yet powerful in detecting shot changes. Among the applications benefiting from this method, summarization of video and Table of Content generation were suggested.

9. REFERENCES

[1] B. Gunsel and A. Tekalp, Content-based video abstraction, 1998 International Conference on Image Processing, 1998, ICIP 98, Proceedings. , 4-7 Oct. 1998 Pages: 128 - 132 vol.3.

[2] J. Yu and M. D. Srinath, An efficient method for scene cut detection, Pattern Recognition Letters, vol. 22, pp. 1379-1391, 2001.

[3] H. Zhang, A. Kankanhalli and S. Smoliar, Automatic partitioning of video, Multimedia Systems, vol. 1(1), pp. 10-28 (1993).

[4] K. Shen and E.J. Delp, "A fast Algorithm for Video Parsing Using MPEG Compressed Sequences", Proceedings of the IEEE International Conference on Image Processing, October 23-26, 1995, Washington, DC., pp 252-255.

[5] B. Yeo and B. Liu, Rapid Scene Analysis on Compressed Video, IEEE Trans. On CSVT, vol. 5, No. 6, Dec., 1995, p. 533-544.

[6] W.A.C Fernando, C.N Canagarajah, D.R Bull, "DFD based scene segmentation for H.263 video sequences," Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on , Volume: 4 , 30 May-2 June 1999 Pages:520 - 523 vol.4

[7] Serkan Kiranyaz, Kerem Caglar, Bogdan Cramariuc and Moncef Gabbouj, Unsupervised Scene Change Detection Techniques in Feature Domain via Clustering and Elimination, Proc. 2002 Tyrrhenian International Workshop on Digital Communications.

[8] U. Gargi, R. Kasturi, and S. Strayer, Performance characterization of video-shot-change detection methods, IEEE Transactions on CSVT, Volume: 10, Issue: 1, Feb. 2000 Pages: 1 –13.