ABRUPT AND GRADUAL SCENE TRANSITION DETECTION IN MPEG-4 COMPRESSED VIDEO SEQUENCES USING TEXTURE AND MACROBLOCK INFORMATION

W.A.C. Fernando, K.K. Loo

Department of Electronic and Computer Engineering, Brunel University,
Kingston Lane, Uxbridge, Middlesex, UB8 3PH, UK
Email: Anil.Fernando@brunel.ac.uk, Jonathan.Loo@brunel.ac.uk

ABSTRACT

In video indexing, video scene transition detection is an essential step to segment the video. Current techniques used for scene change detection tend to suffer from a major limitation as most of them cannot identify scene transitions in the compressed domain. Since most videos are expected to be stored in the compressed domain, scene transition detection in this domain is highly desirable. In this paper, we propose an algorithm for real-time detection of abrupt scene transitions and fade-in/fade-out scene transitions using texture and macroblock (MB) information in the MPEG-4 compressed video. MPEG-4 video stream is partially decoded where texture and MB information are extracted. MB information is used to identity the abrupt scene transitions whereas MB information together with the texture difference between frames are used to identify the boundaries of fading transitions. The performance of the proposed algorithm is measured in terms of precision and recall values. The experimental results show that the proposed algorithm can identify fade and abrupt scene transitions with a mean recall and precision values equal to 80.76% and 85.71%, respectively.

1. INTRODUCTION

With the development of various multimedia compression standards linked with significant increase in desktop computer performance and storage, the widespread exchange of multimedia information is becoming a reality. Audio-visual information is becoming available in digital form in various places around the world. As more and more of this information appears, finding the desired information becomes increasingly difficult. Currently, solutions do exist to allow search for textual information where many text-based search engines are available on the World Wide Web. However, the search for information based on the audio-visual content is difficult as there is no general standardized description for this material. Up to date, MPEG has set a standard, called "Multimedia Content Description Interface (MPEG-7)" that could extend the limited search capabilities to allow efficient retrieval of multimedia information [1,2,3]. It is envision that this information will be associated with the content itself, to allow fast and efficient searching for material of interest.

Today, the major bottleneck preventing the widespread use of digital video is the rapid retrieval of desired information based on the content from a huge database. A reliable way to solve this issue is to index the video sequence properly using a suitable descriptor, thus enabling fast access to the video clips stored in a multimedia database. Video images contain a wider range of primitive data types, require large storage, and can take hours to review, while the comparable process for still images takes seconds at most. Hence the process of organizing video for retrieval is, in some ways, akin to that of abstracting and indexing long text documents. Video is made up of a number of distinct scenes, each of which can be further broken down into individual shots depicting a single view, conversation or action. A common way of organizing a video for retrieval is to prepare a storyboard of annotated still images (often known as key frames) representing each scene.

The most common approach applied to the content based video segmentation is shot transition detection in which a video sequence is partitioned into shots, where each video shot represents a meaningful event or a continuous sequence of action. Shot transitions can be divided into two categories: abrupt transitions and gradual transitions. Gradual transitions include camera movements: panning, tilting, zooming and video editing special effects: fade-in, fade-out, dissolving, wiping.

Abrupt transitions are very easy to detect as the two successive frames are completely uncorrelated. Gradual transitions are more difficult to detect as the differences between frames corresponding to two successive shots is substantially reduced. The number of possible gradual transitions is quite high. However, in practice most gradual transitions fall into either fade/dissolve or wipe transitions. Considerable work has been reported on detecting abrupt transitions both in uncompressed and compressed videos [4,5,6,7,8,9]. The most famous approaches include histogram difference [8], frame difference, motion vector analysis for the uncompressed video [5] and DC-components [6,9], DC-sequences [7] and number of interpolated blocks [4] for the compressed video. A substantial effort has also been devoted toward gradual scene change detection [7,8,10,11,12,13] in the uncompressed video. There are a few algorithms for gradual scene transition detection in MPEG-1/2 compressed video [13]. However, scene transition detection in MPEG-4 video is still an unsolved problem.

The motivation for this research stems from the rapid demand for video segmentation in the compressed domain. Current techniques tend to suffer from a number of major drawbacks. The objective of this work is to overcome some of the limitations of the existing schemes. Earlier discussion

0-7803-8554-3/04/$20.00 ©2004 IEEE.

revealed that searching for information based on its audio-visual content from a huge video data base is a difficult task. This is more complicated if it is done in the compressed domain itself. As explained earlier, the ability to identify shot transitions automatically is the first step towards automatic video indexing or video storyboard browsing. Therefore the main objective of this work is to develop an algorithm for abrupt and fading scene transition detection (STD) in the MPEG-4 compressed video sequences.

The rest of the paper is structured as follows. Section 2 presents the proposed algorithm. Section 3 presents some experimental results. Finally, the conclusion of this research work is drawn in section 4.

2. PROPOSED ALGORITHM

2.1 Scene Change Detection Algorithm

In this paper we present a novel algorithm to identify fade-in and fade-out transitions in the MPEG-4 domain. The most important feature of this algorithm is that full decompression of the encoded sequence is not required.

The proposed scene change detection algorithm is based on analyzing the texture and MB information in the MPEG-4 bit stream. Compressed video bit stream is partially decompressed to obtain features such as motion, texture and shape information. Texture and MB information are used to identify the start and the end frames of fade-in/fade-out transitions in the MPEG-4 compressed domain.

2.1.1 Start of Fade-in/End of Fade-out Detection

Start of a fade-in transition and the end of a fade-out transition has similar properties of an abrupt scene transition. Therefore, algorithms used to detect such a transition can easily be adapted for this purpose. The algorithm described in [4] was adapted for detecting these transitions. By using this approach, the shot boundaries can be detected easily.

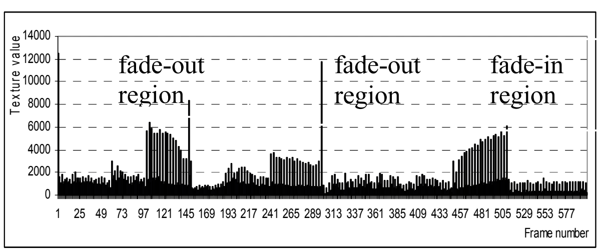

Figure 1. Texture value of the MPEG-4 compressed video sequence

The exact frame number for the abrupt scene transition is kept in an array called *Fs* for further processing. This information, together with start of fade-out/end of fade-in information, is used to detect the boundaries of the fade transition either on the fly in online scene transitions detection or at the end in off-line transitions detection.

2.1.2 Start of Fade-out/End of Fade-in Detection

Detection of the start of fade-out and the end of fade-in is the most complicated part in the fading region identification because there is a considerable correlation between consecutive frames at this point. However, as shown in Figure 1, the texture value of P-frames has a considerable contrast at the start of fade-out and at the end of fade-in scene transition. We can make use of this property to identify the start of fade out and the end of face in transition. Hence, selection of a proper threshold is the main factor of successful identification of such transitions. If the texture value of a P-frame is higher than the selected threshold, then this frame number is kept in an array called *F* as a candidate for the start-of-fade-in or end-of-fade-out.

2.1.3 Identification of the type of the transition

In this paper, the length of the fade transition is assumed to be limited to 75 frames [7]. In online scene transition detection, when a start of fade-out/end of fade-in is detected, the algorithm checks for a previously stored abrupt transition in the array *Fs*. If there is a record of such a transition within the limit of 75 frames, then the transition type is concluded to be a fade-in. The frame number of the abrupt transition is considered to be the start of fade-in transition.

On the other hand, if an abrupt scene transition is detected, then it checks for a start of fade-out in the array *F*. If such a record exists and if it is within the last 75 frames, then it can be concluded as a fade-out transition and the earliest of the detected start of fade-out/end of fade-in is considered to be the start of the fade-out transition.

If there is no match for the start of fade-out/end of fade-in for an abrupt transition within 75 frames, it can be concluded that there is no fading transitions. Complete algorithm is presented in Figure 2.

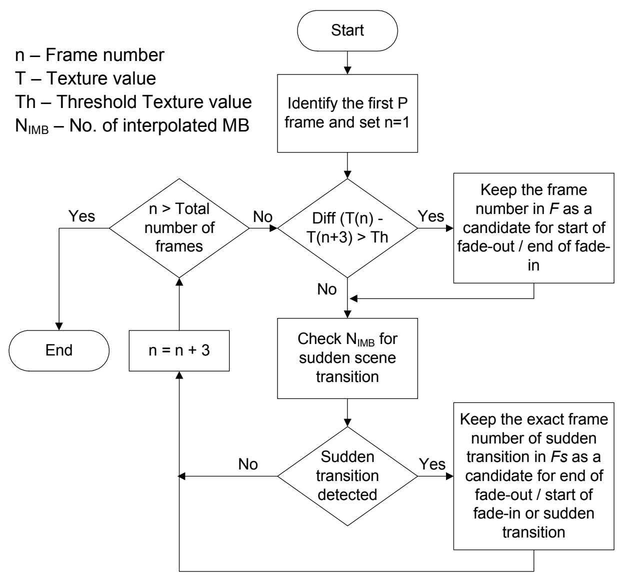

Figure 2. Flowchart of the proposed fading transition detection algorithm

Key steps for fading region identification can be described as follows:

1. Detected abrupt transition points are stored in the array, *Fs*
2. Detected start frame of fade-out and end of fade-in transitions are stored in the array *F*.
3. If an abrupt transition is detected, array *F* is searched for a matching start of fade-out candidate. If there is such a transition within the limit of 75 frames, fade-out transition is identified.
4. If a start frame of fade-out/end of fade-in transitions is detected, array *Fs* is searched for a matching abrupt transition. If there is such a transition within the limit of 75 frames, fade-in transition is identified.
5. If any abrupt transition fails to be matched with a start frame of fade-out/end of fade-in as described in step 3 and 4, an abrupt transition is identified.

3. SIMULATION RESULTS

To evaluate the performance of the proposed scene change detection method, initially, we have tested the algorithm with an image database of 10175 frames with 28 fade scene transitions. In this proposed scheme, we assumed the followings:

- Maximum range of fading transition is 75 frames.
- Minimum range of fading transition is 5 frames.
- Minimum distance between two transitions is 20 frames.

These assumptions are based on the fact that most of the gradual scene transitions are concentrated around 30-50 frames [4].

3.1 Experimental Results with Noiseless Video Sequences

Figure 3 shows measured texture value of each frame for the noiseless video sequence. In the proposed algorithm, difference of the texture values between P-frames are used to detect transitions. Figure 4 shows the difference of texture values between adjacent P-frames. Therefore, it is clear that it is possible to identify the shot boundaries from this information. Complete experimental results for the noiseless test video sequence is tabulated in Table 1.

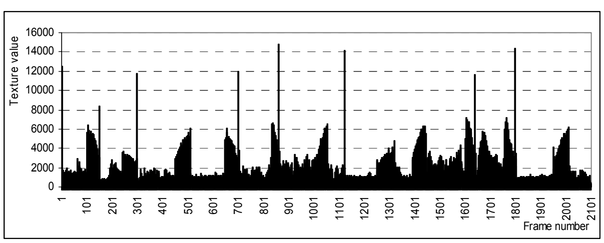

Figure 3. Texture information of frames in the MPEG-4 compressed domain

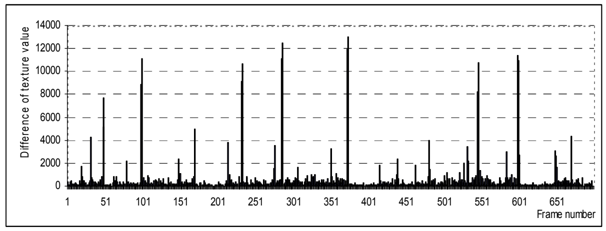

Figure 4. The difference of texture values between P-frames

From Table 1, precision and recall values for the shot detection algorithm can be calculated. There are no any missed or false detections occurred, and precision and recall are both 100% for the noiseless video sequences.

Table 1. Simulation results of the proposed SCD algorithm

Name	Frame No	Actual fading range		Fade type	Detected fading range		Error frame	
		Start	End		Start	End	Start	End
Foreman	1-150	101	150	fade-out	100	150	1	0
Suzie	151-300	241	300	fade-out	241	299	0	1
Silent	301-450	-	-	no fade	-	-	-	-
Container	451-700	451	510	fade-in	450	511	1	1
		646	700	fade-out	646	701	0	1
Carphone	701-860	831	860	fade-out	832	860	1	0
Foreman	861-1000	-	-	no fade	-	-	-	-
Silent	1001-1120	1001	1055	fade-in	1000	1121	1	1
Container	1121-1250	-	-	no fade	-	-	-	-
Suzie	1251-1390	1251	1320	fade-in	1251	1321	0	1
Foreman	1391-1640	1391	1445	fade-in	1392	1447	1	2
		1601	1640	fade-out	1601	1640	0	0
Carphone	1641-1800	1756	1800	fade-out	1756	1802	0	2
Container	1801-1950	-	-	no fade	-	-	-	-
Silent	1951-2100	1951	2010	fade-in	1950	2011	1	1
Total	2100			11			6	10

3.2 Experimental Results with Noisy Video Sequences

We considered a set of four noisy video sequences of total 8075 frames (about 323 seconds) to test the proposed algorithm. These sequences contain several fading and abrupt transitions. Four video sequences '*Spiderman*', '*Mel Gibson*', '*Star Wars*' and '*Orange SMS*' are encoded using an MPEG-4 encoder. Table 2 summarizes the performance of the proposed scene change detection algorithm with these sequences. In this experiment, four transitions were not detected. This is due to the number of interpolated MBs of B-frames exceeds the given threshold, during the transitions. We tested a video data base which contained more than 5 million frames and observed similar performance.

Table 2. Performance calculation of the simulation for MPEG-4 compressed video

Name of test video	No. of frames	Genre	No. of fades	Detected fading transitions			Recall [%]	Precision [%]
				Correct	Miss	False		
Test Sequence	2100	Noise-less test video	11	11	0	0	100.00	100.00
Spiderman	4000	movie	5	3	2	1	60.00	75.00
Mel Gibson	3025	movie	7	5	2	3	57.14	66.67
Star Wars	750	Movie trailer	2	2	0	0	100.00	100.00
Orange SMS	300	TV program	3	3	0	1	66.67	66.67
Total	10175		28	24	4	5	85.71	80.76

3.3 Performance Comparison of the Compressed and the Uncompressed Domain Scene Change Detection

To verify the effectiveness of the proposed algorithm of shot detection, we compared the results of simulations in the compressed and uncompressed domains. In Table 3 the performance of the proposed algorithm and the histogram based algorithm [9] for shot detection are compared.

Table 3. Performance comparison on compressed and uncompressed video shot detection

Parameter	Uncompressed domain shot detection	Compressed domain shot detection
Speed [fps]	1.388	46.67
Precision [%]	77.14	80.76
Recall [%]	96.42	85.71

Precision of our proposed algorithm is higher than uncompressed domain shot detection method. This is because we set two conditions to identify scene change boundaries and it makes less false detections.

Comparing the performance of two algorithms, we summarize the following main advantages:

1. Simple: The proposed algorithm uses features of the video, which can be easily extracted by partially decoding of the encoded bitstream. Total number of parameters needed to implement this algorithm is four.
2. Fast operation: The algorithm can run with a speed of 46.67 frames/second.
3. Real-time applications: The proposed algorithm can be used directly with partially decoded MPEG-4 sequences.

4. CONCLUSIONS

In this research work, we proposed a shot boundary detection algorithm for MPEG-4 compressed video sequences. We used texture and macroblock information from the MPEG-4 compressed domain and properties of fading transitions to identify shots. Experimental results show that the proposed algorithm can detect abrupt and fading scene transitions quite satisfactorily. The performance of the algorithm is measured in terms of hit rate, number of false hits, and miss rate for fades over a set of video sequences. In our experiments, we obtained a recall value of 85.76% and a precision value of 80.71%. Moreover, our algorithm is able to classify fade-ins, fade-outs and cuts. The main cause of false detections of fades in our technique was due to the noise and flashlights. Reason of miss detections is commonly due to the correlation between P-frames in the shot boundary. Unlike in other algorithms, this proposed scheme is simple and computationally inexpensive. However, future work is required to extend this work to identify dissolving and wiping transitions in the MPEG-4 compressed video. Furthermore, to detect gradual transitions, motion, shape and luminance information may also be used. By analyzing these features we can increase shot detection efficiency.

5. REFERENCES

[1] A.B. Benitez, S. Paek,, S.F. Chang, A. Puri, Q. Huang, J.R. Smith, C.S. Li, L.D. Bergman, and C.N. Judice, "Object-based multimedia content description schemes and applications for MPEG-7," *Signal Processing: Image Communication*, Vol. 16, No.1, pp. 235-269, 2000.

[2] J. Hunter, and F. Nack, , "Overview of the MPEG-7 Description Definition Language (DDL) proposals," *Signal Processing: Image Communication*, Vol. 16, No. 1, pp. 271-293, 2000.

[3] ISO/IEC JTC1/SC29/WG11 N3445, "Overview of the MPEG-7 Standard," Geneva, May/June 2000.

[4] W.A.C. Fernando, C.N. Canagarajah, D.R. Bull, "Sudden Scene Change Detection in MPEG-2 Video Sequences", *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pp. 259-264, 1999.

[5] W.A.C. Fernando, C.N. Canagarajah, D.R. Bull, "Video Segmentation and Classification for Content Based Storage and Retrieval Using Motion Vectors", *Storage and Retrieval for Image and Video Databases VII*, pp. 687-698, 1999.

[6] J. Meng, Y. Juan, S.F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence", *Proceedings of IS&T/SPIE*, Vol. 2419, February 1995.

[7] B.L. Yeo, B. Liu, "Rapid Scene Analysis on Compressed Video", *IEEE Transactions on Circuits and Systems for video technology*, Vol. 5, No. 6, pp. 533-544, December 1995.

[8] Yu, H., Wolf, W. "A Multi-resolution Video Segmentation Scheme for Wipe Transition Identification", *Proceedings of ICASSP*, pp. 2965-2968, 1998.

[9] H.J. Zhang, "Automatic Partitioning of Full-Motion Video", *ACM/Spinger Multimedia Systems*, Vol.1, No.1, pp. 10-28, 1993.

[10] A.M. Alattar, "Detecting Fade Regions in Uncompressed Video Sequences", *Proceedings of ICASSP*, pp. 3025-3028, 1997.

[11] A.M. Alattar, "Wipe Scene Change Detector For Segmenting Uncompressed Video Sequences", *Proceedings of ISCAS*, pp. 249-252, 1998.

[12] W.A.C. Fernando, C.N. Canagarajah, D. R. Bull, "Automatic Detection of Fade-in and Fade-out in Video Sequences", *Proceedings of ISCAS, Volume IV-Image and Video Processing, Multimedia, and Communications*, pp.255-258, 1999.

[13] W.A.C. Fernando, C.N. Canagarajah, D. R. Bull, "A Unified Approach to Scene Change Detection in uncompressed and Compressed Video," *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 3, pp. 769-779, 2000.