Shot boundary detection without threshold parameters

H. Koumaras,[a] G. Gardikis,[b] G. Xilouris,[a] E. Pallis,[c] and A. Kourtis[a]
[a]NCSR Demokritos
Institute of Informatics and Telecommunications
Patriarchou Gregoriou Str., 15310 Athens Greece
E-mail: koumaras@iit.demokritos.gr

[b]University of the Aegean
Department of Information and Communication Systems Engineering
Karlovassi 83200 Samos, Greece

[c]Technological Educational Institute of Crete
Department of Applied Informatics and Multimedia
Estauromenos 71500 Heraklion, Greece

Abstract. *Automatic shot boundary detection is a field, where many techniques and methods have been proposed and have claimed to perform reliably, especially for abrupt scene cut detection. However, all the proposed methods share a common drawback: the necessity of a threshold value, which is used as a reference for detecting scene changes. The determination of the appropriate value or the dynamic reestimation of this threshold parameter remains the most challenging issue for the existing shot boundary detection algorithms. We introduce a novel method for shot boundary detection of discrete cosine transform (DCT)-based and low-bit-rate encoded clips, which exploits the perceptual blockiness effect detection on each frame without using any threshold parameter, therefore minimizing the processing demands required for algorithm implementation.* © *2006 SPIE and IS&T.* [DOI: 10.1117/1.2199878]

1 Introduction

Today, a typical end-user of a multimedia system is usually overwhelmed with video collections, facing the problem of organizing them so that they are easily accessible. Thus, to enable an efficient browsing of these video anthologies, it is necessary to design techniques and methods for indexing and retrieving video data. Therefore, the issue of analyzing and automatically indexing the video content by retrieving highly representative information (e.g., shot boundaries) has been raised in the research community.

Several approaches have been proposed in the literature for automatic shot boundary detection (SBD), which can be basically classified according to the detection algorithm that each method implements.

The first group of the SBD methods exploits the variation of the color intensity histograms between consecutive frames. Based on the hypothesis that all frames that belong to the same scene are characterized by the same color histogram, then detecting a color histogram change is a metric for possible scene cut.[1] Another group of methods exploits the classification of frames based on mathematical models, like the analysis of the statistics derived from a specific pixel area along the video sequence.[2] Similarly, other methods are based on edge detection and edge comparison between successive frames,[3] while some specialized methods for MPEG-coded signals have also been proposed.[4,5]

However, all the aforementioned methods use a threshold parameter to distinguish shot boundaries and changes. Thus, a common challenge (stemming from the previously referred methods) prior to the SBD process is the selection of the appropriate threshold for identifying the level of variation, which in turn defines a shot boundary.[6] If a global threshold is used for the detection of shot boundaries over the whole video, then successful detection rate may vary up to 20% even for the same video content.[7] To improve the efficiency and eliminate this performance variation, some later works propose the use of an adaptive threshold, which can be dynamically determined based on the video content.[8,9] But even these methods require a lot of computational power to successfully estimate the appropriate threshold parameter, making their implementation a challenging issue, especially for real-time applications. Another approach uses supervised classifiers instead of thresholds.[10]

This paper introduces a novel method for SBD, which enables the quick and easy extraction of the most significant frames from a discrete cosine transform (DCT)-based encoded video, without requiring any threshold calculation. The proposed method makes use of a multimetric pixel-based algorithm, which calculates for each frame the mean pixel value differences across and at both sides of DCT block margins. Then, the normalized results indicate the magnitude of the tiling effect. The proposed method exploits the fact that during an abrupt scene change over an interframe, the motion estimation and compensation algorithms of the encoding process do not perform well, with the immediate outcome the intensification of the blockiness effect, which may be not perceptually observable (due to the low display duration of each frame), but it is measurable.

2 Proposed Block-based Method

Multimedia applications that distribute audiovisual content are mainly based on DCT-based digital encoding techniques (e.g., MPEG-1/2/4), which achieve high compression ratios by exploiting the spatial and temporal redundancy in video sequences. Most of the standards are based on motion estimation and compensation, using the block-based DCT. The use of the transform facilitates the exploitation in the compression technique of the various psychovisual redundancies by transforming the sequence to a domain, where different frequency ranges with dissimilar sensitivities at the human visual system (HVS) can be accessed independently.

The DCT operates on an $\mathbf{X}$ block of $N \times N$ image samples or residual values after prediction and creates $\mathbf{Y}$, which is an $N \times N$ block of coefficients. The action of the DCT can be described in terms of a transform matrix $\mathbf{A}$. The forward DCT is given by $\mathbf{Y}=\mathbf{AXA}^{\mathrm{T}}$, where $\mathbf{X}$ is a matrix of samples, $\mathbf{Y}$ is a matrix of coefficients, and $\mathbf{A}$ is an $N \times N$ transform matrix. The elements of $\mathbf{A}$ are

Paper 05210LRR received Dec. 5, 2005; revised manuscript received Mar. 9, 2006; accepted for publication Mar. 13, 2006; published online May 9, 2006.
1017-9909/2006/15(2)/020503/3/$22.00 © 2006 SPIE and IS&T.

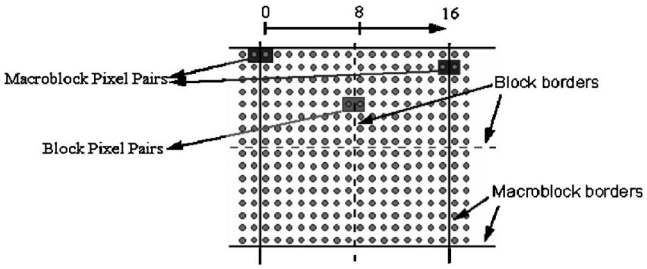

Fig. 1 Pixel pairs that the proposed algorithm uses for blockiness estimation.

$$A_{ij} = C_i \cos\frac{(2j+1)i\pi}{2N} \text{ where } C_i = \begin{cases} (1/N)^{1/2} & i = 0 \\ (2/N)^{1/2} & i > 0. \end{cases} \quad (1)$$

Therefore, the DCT can be written as

$$Y_{xy} = C_x C_y \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} X_{ij} \cos\frac{(2j+1)y\pi}{2N} \cos\frac{(2i+1)x\pi}{2N}. \quad (2)$$

Afterward, in the encoding chain, quantization of the aforementioned DCT coefficients is performed, which is the main reason for the quality degradation and the appearance of artifacts, like the blockiness effect.

The blockiness effect refers to a block pattern of size 8×8 pixels in the compressed sequence, which is the result of the independent quantization of individual blocks of block-based DCT. Due to the DCT, within a block (8×8 pixels), the luminance discontinuities between any pair of adjacent pixels are reduced by the encoding and compression process. On the contrary, for all the pairs of adjacent pixels, located across and on both edge sides of the border of adjacent DCT blocks, the luminance discontinuities are increased through the encoding process.

Especially for video services in the framework of the 3G/4G mobile communication systems, where the encoding bit rate is very low, the blockiness effect is the main present artifact. Especially during a scene change, where the motion estimation and compensation efficiency falls, the blockiness effect is intensified, without being usually noticeable by the viewer,[11] but it is easily measurable by an image processing tool. Thus, by measuring the variance of the blockiness effect during a video sequence, it is possible to identify where and when scene change takes place.

To measure the intensity of the blockiness effect, the average luminance discontinuities at the boundaries of adjacent blocks are calculated by simply comparing the corresponding luminance pixel values. The larger the difference, the more severe is the blockiness effect. For this purpose, for each frame of the video sequence, the individual offsets of the block pixel pairs that Fig. 1 demonstrates are calculated as

$$\text{offset} = |\text{pixel}_i - \text{pixel}_{i+1}|. \quad (3)$$

For clarity, Fig. 2 depicts a graphical representation of the offset that Eq. (3) calculates.

The vertical $\langle\text{offset}\rangle$ values of a frame can be defined as

$$\langle\text{offset}\rangle_V = \sum_{i=1}^{w-8/8}\sum_{j=1}^{h} \frac{|\text{pixel}_{8ij} - \text{pixel}_{8i+1j}|}{h(w-8)/8}. \quad (4)$$

Similarly the horizontal $\langle\text{offset}\rangle$ is

$$\langle\text{offset}\rangle_H = \sum_{i=1}^{w}\sum_{j=1}^{h-8/8} \frac{|\text{pixel}_{i8j} - \text{pixel}_{i8j+1}|}{(h-8)w/8}. \quad (5)$$

Thus, the $\langle\text{offset}\rangle$ for all the pixel pairs of a video frame with width w and height h is calculated as

$$\langle\text{offset}\rangle_{\text{frame}} = \frac{\langle\text{offset}\rangle_V + \langle\text{offset}\rangle_H}{2}. \quad (6)$$

Afterward, the averaged offset per frame is normalized within 0.01 and 1, where 1 denotes the highest blockiness value and 0.01 the lowest one:

$$\text{clip}(0.01, 1, \langle\text{offset}\rangle_{\text{frame}}), \quad (7)$$

where $\text{clip}(x,y,z)$ is a function that normalizes z within the range $[x,y]$. Therefore, by applying Eq. (7) to encoded video sequences, the clipped fluctuation of the averaged offset (i.e., the blockiness effect) per frame can be deduced. Based on this and taking under consideration that during a scene change the blockiness effect instantaneously is strengthened, then Eq. (7) provides a quick and simple metric of scene changes.

Due to the fact that during an abrupt scene change, the values of the $\langle\text{offset}\rangle$ become significantly larger than these of an intrascene $\langle\text{offset}\rangle$, by applying the value normalization with Eq. (7), a clear association is deduced between the clipped $\langle\text{offset}\rangle$ values and the abrupt scene change. More specifically, all the measured clipped $\langle\text{offset}\rangle$ values coming from intrascene frames are relatively low (i.e., <0.1), while the measured clipped $\langle\text{offset}\rangle$ values, resulting from a frame over an abrupt scene change, are equal to 1. The most important is that it is not observed more than a few middle values (i.e., around 0.5), which denote severe camera moving, such as zooming, panning, etc. Thus, the difference between the intrascene and interscene $\langle\text{offset}\rangle$ values is so intense that the requirement for any sophisticated threshold estimation for the shot boundary detection is eliminated.

3 Evaluation of the Proposed Method on Real Video Clips

To evaluate the proposed method, a video sequence of 1500 frames from the motion picture *Spider-Man II* was used as the test signal. The initial PAL (phase alternation line) MPEG-2 video content was transcoded to CIF (common intermediate format) MPEG-4 at 256 kbits/s advanced simple profile. On the final coded signal, an implementation[12] of the aforementioned blockiness estimation algorithm was applied to perform the shot boundary detection. Fig. 3 depicts the deduced $\langle\text{offset}\rangle$ per frame, which was calculated by this procedure.

Based on Fig. 3, it is also experimentally proved that all the measured clipped $\langle\text{offset}\rangle$ values that come from intrascene frames are relatively low (i.e., <0.1), while the measured clipped $\langle\text{offset}\rangle$ values, resulting from a frame over an abrupt scene change, are equal to 1.

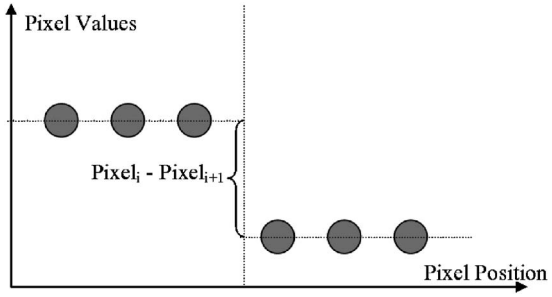

Fig. 2 Graphical representation of the offset between a macroblock/block pixel pair.

To eliminate the case of a false frame report due to the blockiness propagation from a successfully detected scene-cut frame to its successive neighboring frames, an interval of frames from the last scene change detection (e.g., 25 frames) is considered, during which no scene change is reported even if it is detected.

The efficiency of the proposed method, with the aforementioned described configuration, was also tested on a set of various heterogeneous CIF MPEG-4 video clips encoded at 256 kbits/s, containing both media clips with abrupt and gradual scene cuts. The corresponding results are depicted in Table 1, along with the performance, for the same encoding bit rate area, of two other existing threshold-exploited shot boundary detection methods for MPEG video[13] (for method 1 see Ref. 14, and method 2 see Ref. 15).

From Table 1, we can deduce that although the proposed method performs similarly to existing threshold-exploited methods regarding the recall metric, it outperforms the rest of the methods for the precision of the scene detection for both abrupt and gradual scene changes, retaining at the same time significantly lower computational cost, due to the absence of a threshold parameter.

Table 1 Comparison of the proposed method for abrupt and gradual scene changes.

	Recall	Precision
Method	Abrupt Scene Change	
Proposed method	0.60	0.89
Method 1	0.52	0.79
Method 2	0.74	0.58
	Gradual Scene Change	
Proposed method	0.20	0.40
Method 1	0.21	0.20
Method 2	0.24	0.06

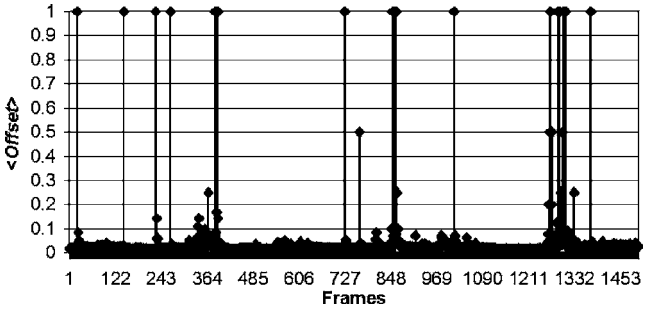

Fig. 3 Parameter ⟨offset⟩ per frame for the *Spider-Man II* test signal.

4 Conclusions

We presented a method for SBD without any threshold parameter. Using only the increment of the blockiness effect during a scene cut, the proposed method successfully detects where a scene cut occurs. The efficiency of the proposed technique was successfully tested on both abrupt and gradual scene changes and compared to other existing shot boundary detection methods.

Acknowledgments

This work was carried out within the "PYTHAGORAS II" research framework, jointly funded by the European Union and the Hellenic Ministry of Education.

References

1. H. Ueda, T. Miyatake, and S. Yoshizawa, "IMPACT: an interactive natural-motion-picture dedicated multimedia authoring system," in *Proc. of CHI*, pp. 343–350, ACM, New York (1991).
2. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.* **1**(1), 10–28 (1993).
3. R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proc. ACM Multimedia*, pp. 189–200, San Francisco, CA (1993).
4. M. Sugano, M. Furuya, Y. Nakajima, and H. Yanagihara, "Shot classification and scene segmentation based on MPEG compressed movie analysis," in *IEEE Pacific Rim Conf. on Multimedia (PCM) 2004*, pp. 271–279 (2004).
5. K. Hoashi, M. Sugano, M. Naito, K. Matsumoto, F. Sugaya, and Y. Nakajima, "Shot boundary determination on MPEG compressed domain and story segmentation experiments for TRECVID 2004," *Text Retrieval Conf. Video Retrieval Evaluation (TRECVID)* (2004).
6. H. Lu and Y. Tan, "An effective post-refinement method for shot boundary detection," *IEEE Trans. Circuits Syst. Video Technol.* **15**(11), 1407–1421 (2005).
7. C. O'Toole, A. Smeaton, N. Murphy, and S. Marlow, "Evaluation of automatic shot boundary detection on a large video suite," presented at the 2nd U.K. Conf. Image Retrieval: The Challenge of Image Retrieval, Newcastle, U.K. (1999).
8. R. Lienhart, "Comparison of automatic shot boundary detection algorithms," *Proc. SPIE* **2670**, 170–179 (1996).
9. A. Dailianas, R. B. Allen, and P. England, "Comparison of automatic video segmentation algorithms," *Proc. SPIE* **2615**, 2–16 (1995).
10. Y. Qi, A. Hauptmann, and T. Liu, "Supervised classification for video shot segmentation," in *Proc. 2003 Int. Conf. on Multimedia and Expo.*, Vol. 2, pp. 689–692 (2003).
11. W. J. Tam, L. Stelmach, L. Wang, D. Lauzon, and P. Gray, "Visual masking at video scene cuts," *Proc. SPIE* **2411**, 111–119 (1995).
12. J. Lauterjung, *Picture Quality Measurement*, IBC, Amsterdam (Sep. 1998).
13. U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods," *IEEE Trans. Circuits Syst. Video Technol.* **10**(1), 1–11 (2000).
14. B.-L. Yeo and B. Liu, "A unified approach to temporal segmentation of motion JPEG and MPEG compressed video," in *Proc. IEEE 2nd Int. Conf. Multimedia Computing and Systems*, pp. 81–83 (1995).
15. K. Shen and E. J. Delp, "A fast algorithm for video parsing using MPEG compressed sequences," in *Proc. IEEE Int. Conf. Image Processing*, pp. 252–255 (1995).