Fast Scene Change Detection Using Direct Feature Extraction from MPEG Compressed Videos *

Young-Min Kim, Sung Woo Choi and Seong-Whan Lee
Center for Artificial Vision Research, Korea University
Anam-dong, Seongbuk-ku, Seoul 136-701, Korea
{ymkim, swchoi, swlee}@image.korea.ac.kr

Abstract

In order to process video data efficiently, a video segmentation technique through scene change detection must be employed. Many of advanced video applications require manipulations of compressed video signals. So, the scene change detection process is achieved by analyzing the video directly in the compressed domain, thereby avoiding the overhead of decompressing video into individual frames in the pixel domain.

In this paper, we propose a fast scene change detection algorithm using direct feature extraction from MPEG compressed videos, and evaluate this technique using sample video data. This process was made possible by a new mathematical formulation for deriving the edge information directly from the DCT coefficients.

1. Introduction

For scene change detection, a matching process between two consecutive frames is required. Humans can easily identify some objects from their edge maps and edge maps are not sensitive to luminance or color change. We can derive such binary edge maps as a representation of key-frames. Two frames can then be compared by calculating a correlation between their edge maps[1]. Therefore, in this paper, we used edge information for the frame matching feature.

Due to the large amount of data, video sequences are often compressed for efficient transmission or storage on-line. The compressed video sequences have to undergo computationally intensive processing steps to be de-compressed, prior the application of any scene change detection algorithms.

*This research was supported by Creative Research Initiatives of the Ministry of Science and Technology, Korea.

In Figure 1, the shaded blocks are the most time consuming processes. In this paper, we propose a fast scene change detection algorithm using direct feature extraction from MPEG compressed videos without the process of the shaded blocks. Especially, we developed new formulas based on mathematical analysis which give directly the edge information such as orientation, strength and offset from the DCT coefficients.

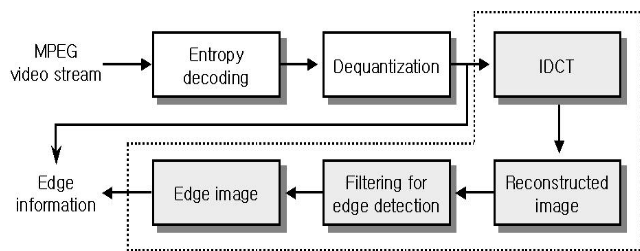


Figure 1. The process of edge information extraction from MPEG compressed images

2. Related works

Scene change detection algorithm for uncompressed video data are divided into the method using pixel-wise difference[6], the method using histogram difference based[3], the method using edge image difference[5], etc.

Scene change detection algorithm for compressed video data are divided into the method using luminance histogram difference of DC images[4], the method using macro block types[2], and the method using correlations of DCT(Discrete Cosine Transform) coefficients[7].

3. The proposed scene change detection algorithm

We can divide the proposed algorithm into direct edge information extraction and scene change detection through matching between two consecutive frames. Further details for this section can be found in Journal of The Korea Information Science Society, vol. 27.(To be appeared)

3.1. Direct edge information extraction

The AC coefficients essentially depend upon intensity differences in the vertical or horizontal direction.

3.1.1 Ideal edge model in DCT domain

In this paper, we propose a new algorithm based on mathematical formulation which extracts edge information directly from MPEG video data using the relation of AC coefficients. We consider orientation, strength and edge offset to be the important components definining the edge shape. Figure 2 shows the proposed ideal step edge model. θ, d, I means orientation, offset and intensity value, respectively.

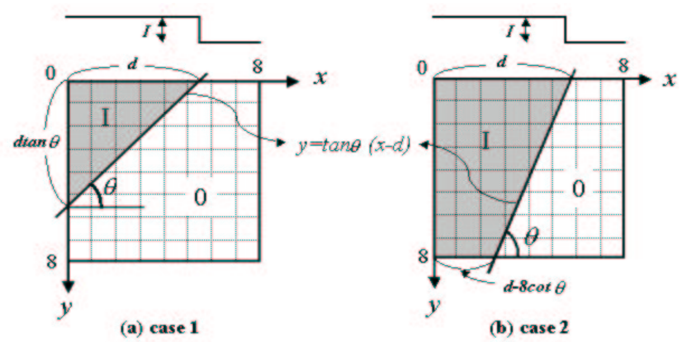


Figure 2. The proposed ideal step edge model

3.1.2 Approximation

We can derive AC_{uv} coefficients using the equation below.

$$AC_{uv} = \frac{1}{4}C_uC_v\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}f(i,j), \quad (1)$$

where

$$C_\lambda = \begin{cases} \frac{1}{\sqrt{2}} & for\ \lambda = 0 \\ 1 & for \lambda = 1,2,...,7. \end{cases}$$

We can consider $f(i,j)$ as a continuous function $f(x,y)$ $(0 \leq x \leq 8,\ 0 \leq y \leq 8)$. So, the Equation (1) is exactly equal to the equation below.

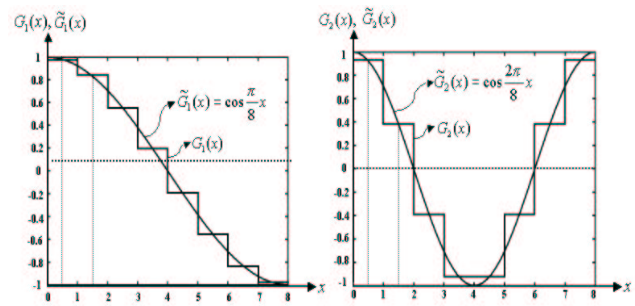
$$AC_{uv} = \frac{1}{4}C_uC_v\int_0^8\int_0^8 G_u(x)G_v(y)f(x,y)dxdy \quad (2)$$

where G_λ (λ = 0, 1, 2, ..., 7) is defined by

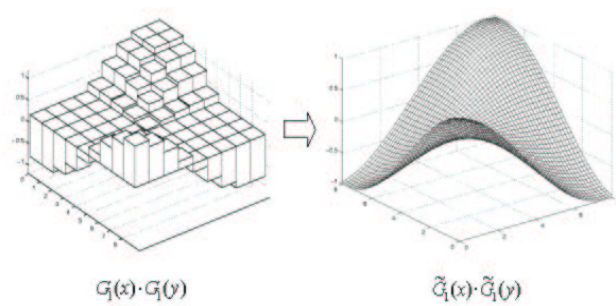
$$G_\lambda(x) = \cos\frac{(2i+1)\lambda\pi}{16},\ (i \leq x \leq i+1,\ i = 0,1,2,...,7).$$

We approximate G_λ by $\widetilde{G_\lambda}$ which is defined by $\widetilde{G_\lambda(x)} = \cos\frac{\lambda\pi}{8}x$, $(0 \leq x \leq 8)$. Figure 3 shows G_1, G_2 and $\widetilde{G_1}$, $\widetilde{G_2}$. If u and v are small enough, we can approximate AC_{uv} by the following integral $\widetilde{AC_{uv}}$.

$$\begin{aligned}\widetilde{AC_{uv}} &= \frac{1}{4}C_uC_v\int_0^8\int_0^8\widetilde{G_u}(x)\widetilde{G_v}(y)f(x,y)dxdy \quad (3)\\ &= \frac{1}{4}C_uC_v\int_0^8\int_0^8\cos\frac{u\pi}{8}x\cos\frac{v\pi}{8}yf(x,y)dxdy\end{aligned}$$



(a) The case of a single dimension ($\lambda = 1$ and $\lambda = 2$)



(b) The case of two dimensions

Figure 3. The meaning of approximation for extracting AC coefficients

The conditions for the discrimination of the cases in Figure 4 will be apparent, after calculate the orientation, the strength and the offset in terms of the relation of AC coefficients using $\widetilde{AC_{uv}}$.

3.2. Calculation of edge information

Section 3.1 enable us to compute orientation ($\tan\theta$), strength(I) and offset(d) from the $\widetilde{AC}$ coefficients, for case

(2) and case (3) in Figure 4. Since $\widetilde{AC}$ coefficients are approximations to AC coefficients, we will denote $\widetilde{AC}$ also by AC from now on. We present different metrics of DCT coefficients to obtain accurate edge orientation, strength and offset information.

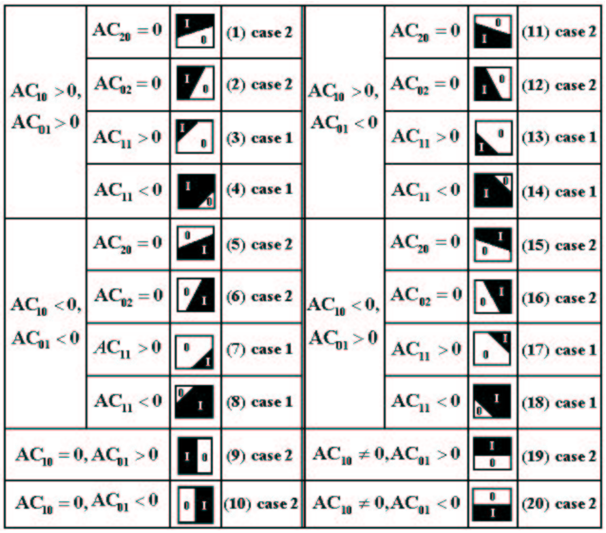


Figure 4. The proposed edge extraction algorithm using the correlation between AC coefficients

3.3. Symmetry rules

From the results in Section 3.2, we can calculate the edge information for the rest of the cases in Figure 4 using symmetry. Further details for symmetry rules can be found in Appendix B.

3.4. Frame matching phase

3.4.1 Edge orientation histogram comparison

We can use edge orientation histogram for frame matching. In Equation (4), f_n means n-th frame, DOAH(f_n,f_{n+1}) means the difference of the angle histograms between the n-th frame and the $(n+1)$-th frame, Q_A means the number of phases of the orientation histogram, and $AH_n(i)$ means i-th orientation histogram of frame n.

$$DOAH(f_n, f_{n+1}) = \sum_{i=0}^{Q_A-1} |AH_n(i) - AH_{n+1}(i)| \quad (4)$$

3.4.2 Edge strength histogram comparison

Because the differences of edge orientation histograms are sensitive to camera speed or camera rotation, our algorithm uses an edge strength histogram. In Equations (5) - (8), M and N are the numbers of the horizontal and the vertical blocks of a frame respectively, K is the number of groups into which the vertical blocks are divided, and $ST_n(i,j)$ means the edge strength of the (i,j)-th block of the n-th frame.

$$Y_n(k) = \sum_{j=k\frac{N}{K}}^{\{(k+1)\frac{N}{K}\}-1} \sum_{i=0}^{M-1} ST_n(i,j) \quad (5)$$

$$D_{nk} = |Y_n(k) - Y_{n+1}(k)| \quad (6)$$

$$\varphi_{n,n+1}(k) \begin{cases} 1 & if\ D_{nk} > \Delta_T \\ 0 & otherwise \end{cases} \quad (7)$$

$$DOSH(f_n, f_{n+1}) = \frac{\sum_{k=0}^{K-1} \varphi_{n,n+1}(k)}{K} \quad (8)$$

where Δ_T is the prescribed threshold.

3.4.3 Frame matching using edge orientation and strength

In this paper, we match two consecutive frames using edge orientation and strength information which are acquired in the previous section. Equation (9) shows the weighted summation of orientation and strength histogram. In Equation (9), DOF(f_n,f_{n+1}) means the difference of the n-th and the $(n+1)$-th frames, and α means the prescribed weight($0 \leq \alpha \leq 1$).

$$\begin{aligned} DOF(f_n, f_{n+1}) &= (1-\alpha)DOAH(f_n, f_{n+1}) \\ &+ \alpha DOSH(f_n, f_{n+1}) \end{aligned} \quad (9)$$

4. Experimental results and analysis

Table 1 shows the scene change detection results with experimental video data using the method of Edge-based Features(EF method)[5], that is a very accurate scene change detection algorithm in uncompressed domain, the method of DC image(DC method)[4], that is very promising and produces the best results among the previous works in compressed domain, and the Proposed Method(PM method).

Table 1. Comparison of the proposed method with the others(N_C, N_{FN}, N_{FP})

	New	Sit	Doc	Mus
DC	(181,5,8)	(141,1,5)	(90,4,12)	(183,6,23)
EF	(185,1,3)	(142,0,1)	(93,1,2)	(187,2,6)
PM	(183,3,5)	(141,1,1)	(91,3,5)	(187,2,9)

In Table 1, N_C means the number of correct scene change detection, N_{FN} means the number of false negatives, and N_{FP} means the number of false positives. DC

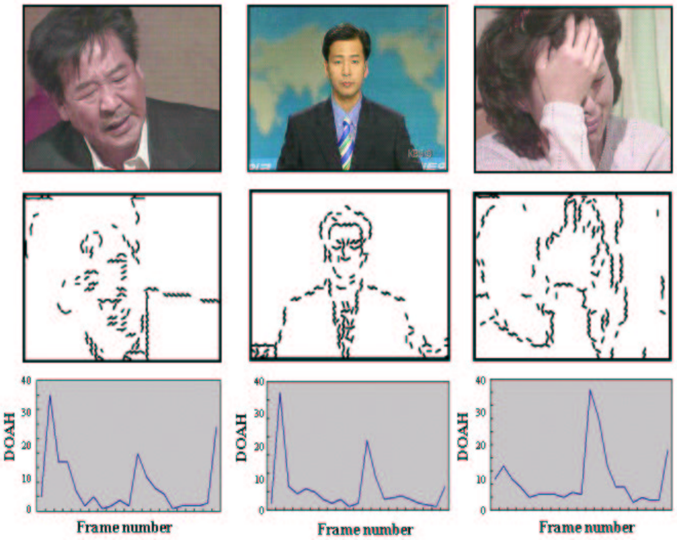


Figure 5. The examples of original image, edge image and orientation histogram graph

method is very sensitive to luminance or color change, so many false scene change frames were detected when we used music video data and documentary data. But EF method and PM method are not sensitive to luminance or color change.

$$Precision = \frac{N_C}{N_C + N_{FP}}, \quad Recall = \frac{N_C}{N_C + N_{FN}} \tag{10}$$

Table 2 shows performance comparisons of the scene change detection methods via the precision and recall parameters.

Table 2. Accuracy comparison of scene change detection method via the precision and recall parameter(precision, recall)

	New	Sit	Doc	Mus
DC	(0.96,0.97)	(0.97,0.99)	(0.88,0.96)	(0.89,0.97)
EF	(0.98,0.99)	(0.99,1.00)	(0.98,0.99)	(0.97,0.99)
PM	(0.97,0.98)	(0.99,0.99)	(0.95,0.97)	(0.95,0.99)

Table 3 demonstrates the speed comparisons of scene change detection methods. DC method and PM are performed using direct feature extraction in the compressed domain, therefore these methods are faster than EF method in the uncompressed domain. The experiments show that DC method and PM method are five to six times faster than EF method.

Experimental results demonstrate that the proposed method is more accurate than DC method, and faster than EF method.

5. Conclusion and further research

In this paper, we proposed a new scene change detection algorithm using direct edge information extraction from MPEG video data, and evaluated this technique using sample video data. The proposed algorithm is comparable to the DC method[4] in speed, and was found to be five to six times faster than the EF method[5]. This was made possible by a new mathematical formulation for deriving the edge information directly from the DCT coefficients.

Table 3. Speed comparison of the scene change detection methods(frame/sec.)

	New	Sit	Doc	Mus
DC	11.4	10.3	11.2	10.8
EF	2.1	2.7	2.3	2.1
PM	11.4	11.3	10.7	10.8

We are investigating the possibilities of developing gradual scene detection methods and frame matching using global motion information. If the proposed method is augmented with such additional machineries, then the overall scene change detection algorithm is expected to be much improved.

References

[1] T. Kato, T. Kurita, N. Otsu, and K. Hirata. A sketch retrieval method for full color image database. *Proc. of 11th International Conference on Pattern Recognition*, pages 530–533, 1992.

[2] V. Kobla, D. S. Doermann, and K.-I. Lin. Archiving, indexing, and retrieval of video in the compressed domain. *Proc. of SPIE : Multimedia Storage and Archiving Systems*, 2916:78–89, 1996.

[3] D. Swanberg, C. F. Shu, and R. Jain. Knowledge guided parsing in video database. *Proc. of SPIE'93 - Storage and Retrieval for Image and Video Database*, 1908:13–24, 1993.

[4] B. Yeo and B. Liu. Rapid scene analysis on compressed videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):533–544, 1995.

[5] R. Zabiha, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breadks. *Proc. of ACM Multimedia '95*, pages 189–200, 1995.

[6] H. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

[7] H. Zhang, C. Y. Low, and S. W. Smoliar. Video parsing and browsing using compressed data. *Multimedia Tools and Applications*, 1:89–111, 1995.