Fast Scene Change Detection using Direct Feature Extraction from MPEG Compressed Videos

Seong-Whan Lee, *Senior Member, IEEE*, Young-Min Kim, and Sung Woo Choi

***Abstract*—In order to process video data efficiently, a video segmentation technique through scene change detection must be required. This is a fundamental operation used in many digital video applications such as digital libraries, video on demand (VOD), etc. Many of these 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. First, we derive binary edge maps from the AC coefficients in blocks which were discrete cosine transformed. Second, we measure edge orientation, strength and offset using correlation between the AC coefficients in the derived binary edge maps. Finally, we match two consecutive frames using these two features (edge orientation and strength). This process was made possible by a new mathematical formulation for deriving the edge information directly from the discrete cosine transform (DCT) coefficients. We have shown that the proposed algorithm is faster or more accurate than the previously known scene change detection algorithms.

***Index Terms*—Direct feature extraction, discrete cosine transform, edge information, MPEG compressed video, scene change detection.**

I. Introduction

With rapid advances in communication and multimedia computing technologies, accessing mass amounts of multimedia data is becoming a reality on the information superhighway. Video and text data are now equally used in multimedia information. So the demand for various video services—such as video on demand (VOD), digital library, etc.—is rapidly increasing [1], [2]. As the amount and complexity of video information grows, the need for more intelligent video manipulating techniques becomes evident. For efficient video storage and management, video segmentation using scene change detection (cut detection) must be performed prior to all other process [3], [4].

Manuscript received December 19, 1999; revised September 6, 2000. This work was supported by Creative Research Initiatives of the Korean Ministry of Science and Technology. The associate editor coordinating the review of this paper and approving it for publication was Prof. Mohammed Ghanbari.

S.-W. Lee and Y.-M. Kim are with the Center for Artificial Vision Research/Department of Computer Science and Engineering, Korea University, Seoul 136-701, Korea (e-mail: swlee@image.korea.ac.kr; ymkim@image.korea.ac.kr).

S. W. Choi was with the Center for Artificial Vision Research/Department of Computer Science and Engineering, Korea University, Seoul 136-701, Korea. He is now with the Max-Planck-Institut Informatik, D-66123 Saarbrücken, Germany.

Publisher Item Identifier S 1520-9210(00)10248-2.

Generally, video data consist of frame, shot and scene. Video segmentation is a technique that divides video data into physical units, generally called shots. A shot is a video segment that consists of one continuous action. A shot usually consists of several frames, while a scene can consist of several shots.

For scene change detection, a matching process between two consecutive frames is required. Many researchers have used the luminance pixel-wise difference or luminance or color histogram difference to match two frames [5], however, luminance or color is sensitive to small change, so these features produce false alarms. By contrast, humans can easily identify some objects from their edge maps and edge maps which are not sensitive to luminance or color change. We have derived such binary edge maps as a representation of key-frames. Two frames can then be compared by calculating a correlation between their edge maps [6]. 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. However most current scene change detection algorithms operate on uncompressed video sequences. Experiments show, however, that in image/video decoding, approximately 40% of CPU time is spent in inverse discrete cosine transform (IDCT) even using available fast discrete cosine transform (DCT) algorithms [7]. Therefore, these compressed video sequences have to undergo computationally intensive processing steps to be de-compressed, prior to the application of any scene change detection algorithms [8].

In this paper, we propose a fast scene change detection algorithm using direct feature extraction from MPEG compressed videos. Overall, the proposed algorithm can be split into three parts. First, we derive binary edge maps from the AC coefficients in blocks which were discrete cosine transformed. Second, we measure edge orientation, strength and offset using the correlation between the AC coefficients in the derived binary edge maps. Finally, we match two consecutive frames using these two features (edge orientation and strength). In Fig. 1, the shaded blocks are the most time consuming processes. The proposed algorithm can extract edge information directly from MPEG video data 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. Moreover, compared to the previously proposed formulas [17], ours show better accuracy results.

This paper is organized as follows. In Section II, we review previous works related to scene change detection. In Section III, we explain the background for understanding this paper. In Sec-

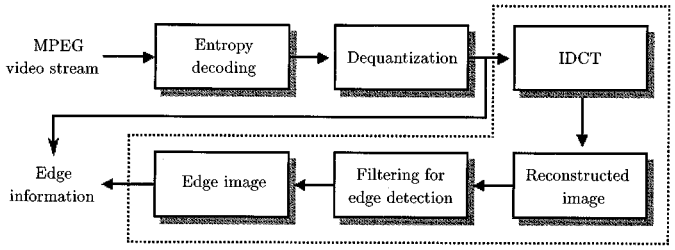

Fig. 1. Process of edge information extraction from MPEG compressed images.

tion IV, we propose a new scene change detection algorithm. In Section V, experimental results with various video data are analyzed to evaluate the performance of the proposed algorithm. Finally, conclusions and some directions for further researches will be given in Section VI.

II. Related Works

Scene change detection is a method used to divide a video sequence into its elementary scene (generally called "shots") and is a common first step in video processing. Scene change detection algorithm can be divided into algorithm on uncompressed video and compressed video.

Scene change detection algorithms for uncompressed video data are further divided into pixel-based methods, local area-based methods, and frame-based methods. For example, the method using pixel-wise difference [5], the method using local luminance histogram difference [9] and the method using edge image difference[10], etc. Especially, Zabih *et al.* [10] proposed an edge-based method, which detects the appearance of intensity edges that are distant from edges in the previous frame and appears to be more accurate at detecting and classifying scene change detection points that are difficult to detect with intensity histograms.

However, Zabih *et al.* [10] used methods for scene change detection algorithm on uncompressed video. If the input video data are in a compressed form, then they must first be decompressed in their entirety and edge image information is then extracted from the Entire decompressed image. Since the algorithms are already compute-intensive, additional computation time is highly undesirable. A more effective approach is to develop tools that can work directly on compressed representations [11].

Scene change detection algorithms for compressed video data are divided into three categories; the DC image-based method, the motion vector-based method and the DCT coefficient-based method. For example, the method using the luminance histogram difference of DC images[13], [14], the method using macro block types [15], and the method using correlations of DCT coefficients [11], [12].

Yeo and Liu [13] proposed to detect scene changes on the DC image of the compressed video data. The DC sequence is first reconstructed using the approximation method prior to its use in the identification of scene change detection. They discussed successive pixel difference and color statistical comparison. The successive pixels difference is sensitive to camera and object motion. But because DC sequences are smoothed images of the corresponding full images, they are less sensitive to camera and object movements. Color statistical comparison is found to be less sensitive to motion, but more expensive to be computed. This approach is nonetheless very promising and produces the best reported results.

Kobla and Doermann [15] proposed an algorithm for detecting scene change points in MPEG compressed video without performing expensive decoding computations. They used the macroblocks (MB) of the P and B frames in MPEG video and used the DCT information of I frames only for cases where the MB information proves insufficient. This method ,however, also has a motion compensation problem in that related information tends to be unreliable and unpredictable in the case of gradual transitions.

Arman *et al.* [12] proposed segmenting JPEG video with a difference metric based on the correlation between the DCT coefficients of consecutive frames. This approach is computationally efficient. However, it dose not address gradual transition and the experimental evaluation of the algorithm is not very sufficient.

Zhang *et al.* [11] proposed the method using the pair-wise difference of DCT coefficients. They have developed a novel hybrid approach, which exploits the advantages of simpler difference metric to the comparison of DCT coefficients and motion information edcoded in MPEG data. Motion compensation related information, on the other hand, tends to be unreliable and unpredictable in the case of gradual transitions, resulting in failure.

Sethi and Patel [14] used only the DC coefficients of I frames of MPEG compressed video to detect scene changes based on luminance histogram. Their algorithm works as follows. First, I frames are extracted from the compressed video streams. Second, the luminance histogram of I frames are generated using the first DC coefficient. Finally, the luminance histograms of consecutive frames are compared using one of the three statistical tests (χ histogram comparison test).

III. Background

A. Meaning of AC Coefficients

MPEG video is specifically designed for compression of video sequences. A video sequence is simply a series of pictures taken at closely spaced intervals in time. Except for the special case of a scene change, these pictures tend to be quite similar from one to the next. Intuitively, a compression system ought to be able to take advantage of this similarity [16]. MPEG uses a two-dimensional eight point by eight point form of the DCT. The 2-D DCT becomes

$$\mathrm{AC}_{uv} = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1)u\pi}{16} \cdot \cos \frac{(2j+1)v\pi}{16} f(i,j) \tag{1}$$

where

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

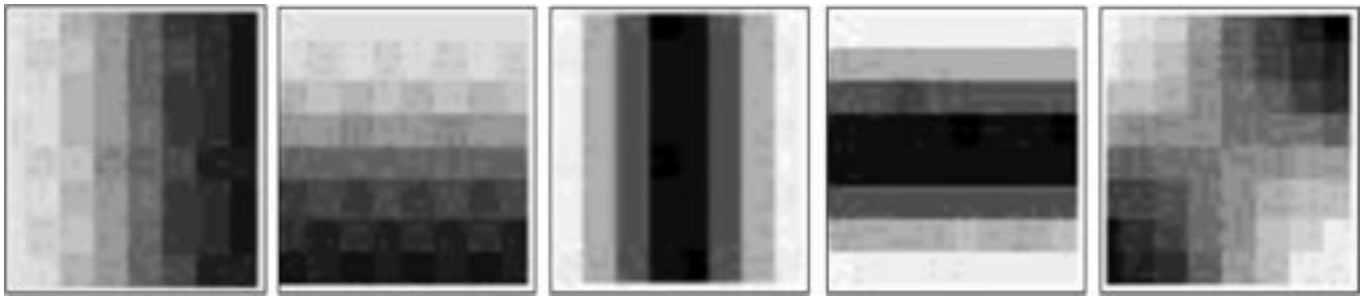

Fig. 2. The 2-D basis functions of the 8 × 8 DCT.

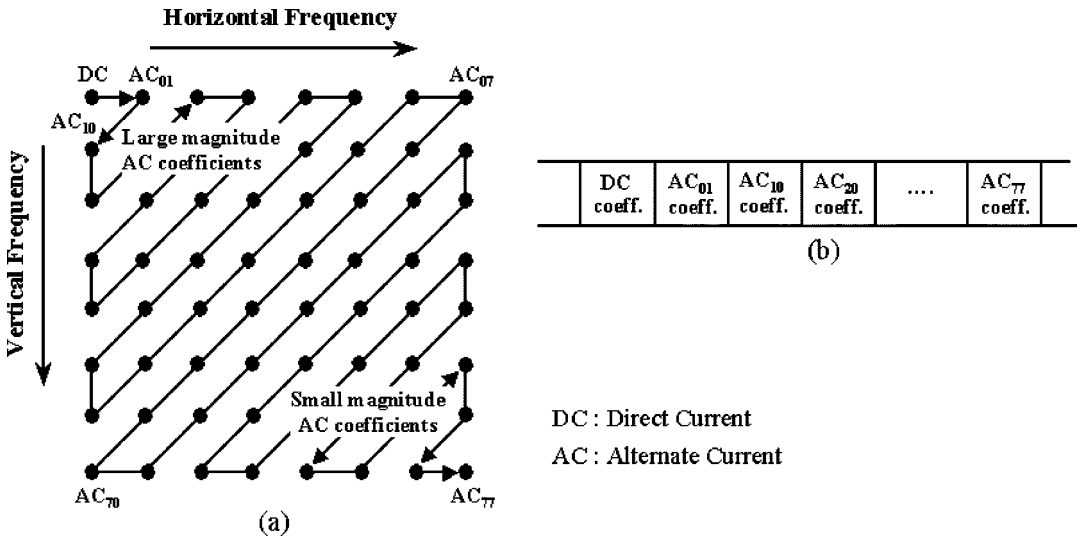

Fig. 3. Zigzag ordering of DC and AC coefficients.

We will use the relationship between the pixels' DCT coefficients for the extraction of edge information. The coefficient in the upper left corner of a DCT encoded block is called "dc coefficient" and the others are called "ac coefficients." The DC coefficient represents the average of the original image and the AC coefficients represent variations in gray values in certain direction at a certain rate. For example, consider the coefficient AC_{01} and AC_{10}. From (1), we have

$$\mathrm{AC}_{01} = \frac{C_0 C_1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2j+1)\pi}{16} f(i,j) \qquad (2)$$

$$\mathrm{AC}_{10} = \frac{C_1 C_0}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1)\pi}{16} f(i,j) \qquad (3)$$

which can be represented as

$$\mathrm{AC}_{01} = \frac{C_0 C_1}{4} \left\{ \cos \frac{\pi}{16} \left(\sum_{j=0}^{7} f(0,j) - \sum_{j=0}^{7} f(7,j) \right) + \cos \frac{3\pi}{16} \left(\sum_{j=0}^{7} f(1,j) - \sum_{j=0}^{7} f(6,j) \right) + \cos \frac{5\pi}{16} \left(\sum_{j=0}^{7} f(2,j) - \sum_{j=0}^{7} f(5,j) \right) + \cos \frac{7\pi}{16} \left(\sum_{j=0}^{7} f(3,j) - \sum_{j=0}^{7} f(4,j) \right) \right\} \qquad (4)$$

$$\mathrm{AC}_{10} = \frac{C_1 C_0}{4} \left\{ \cos \frac{\pi}{16} \left(\sum_{i=0}^{7} f(i,0) - \sum_{i=0}^{7} f(i,7) \right) + \cos \frac{3\pi}{16} \left(\sum_{i=0}^{7} f(i,1) - \sum_{i=0}^{7} f(i,6) \right) + \cos \frac{5\pi}{16} \left(\sum_{i=0}^{7} f(i,2) - \sum_{i=0}^{7} f(i,5) \right) + \cos \frac{7\pi}{16} \left(\sum_{i=0}^{7} f(i,3) - \sum_{i=0}^{7} f(i,4) \right) \right\}. \qquad (5)$$

Equations (2)–(4) means that AC_{01} and AC_{10} essentially depend on intensity differences in the horizontal and vertical directions, respectively. The 2-D basis functions are shown as 8 × 8 grayscale arrays in Fig. 2.

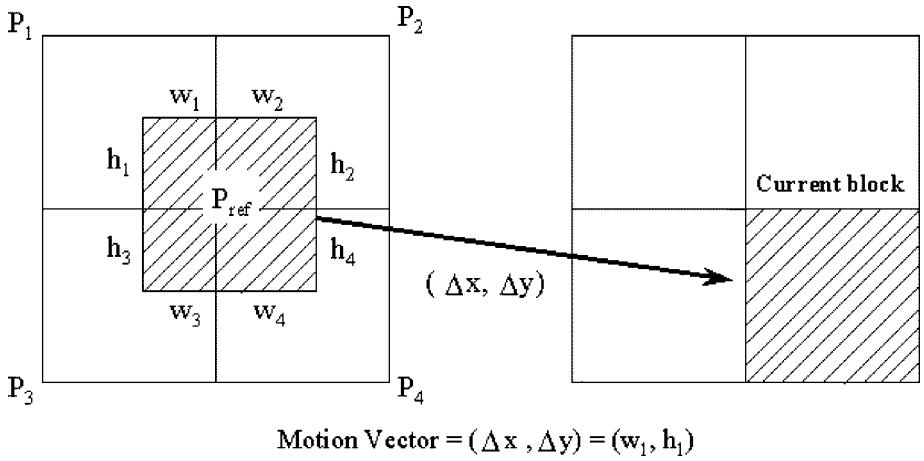

Fig. 4. Reference block (P_{ref}), motion vectors, and original blocks.

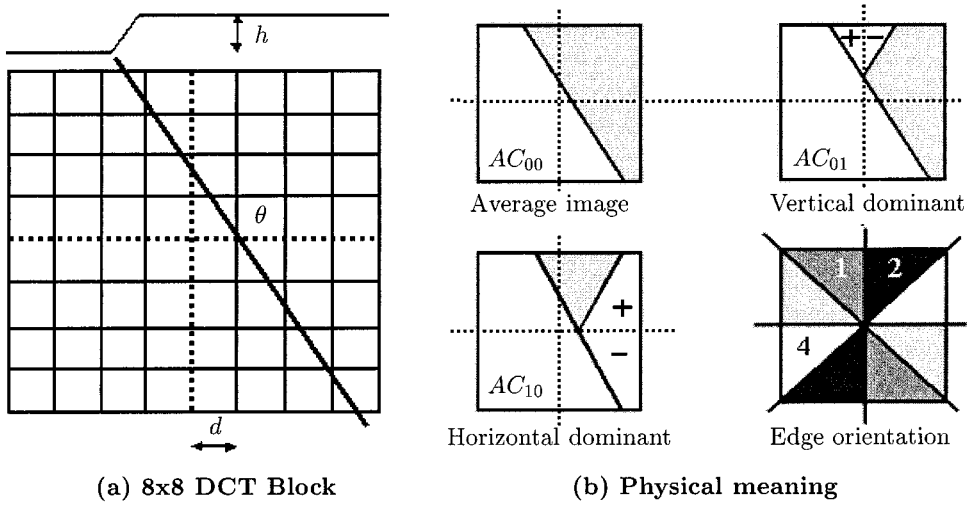

Fig. 5. Ideal step edge model and physical meanings of AC coefficients in a block [17].

B. Extraction of AC Coefficients

MPEG divides the pictures in a sequence into three basic categories: I-picture, P-picture, and B-picture. Intra-coded pictures or I-pictures are coded without reference to preceding or upcoming pictures in the sequence. Predicted pictures or P-pictures are coded with respect to the temporally closest preceding I-picture or P-picture in the sequence. Bidirectionally coded pictures or B-pictures are interspersed between the I-pictures and P-pictures in the sequence, and are coded with respect to the immediately adjacent I- and P-pictures either preceding, upcoming, or both. Even though several B-pictures may occur in immediate succession, B-pictures may never be used to predict another picture [16].

1) Extraction From I Frames: Since I-pictures are coded without reference to neighboring pictures in the sequence, we can get the AC coefficients without using any other processes. The de-correlation provided by the DCT permits the AC coefficients to be coded independently of one another, and this greatly simplifies the extraction process.

2) Extraction From P or B Frames: The decorrelation property of the DCT is too colloquial applicable only to intra coded pictures. Non-intra pictures are coded relative to a prediction from another picture, and the process of predicting strongly decorrelates the data. So we must consider nonintra blocks in P- and B-frames.

TABLE I
MATRICES S_{i1} AND S_{i2}

Subblock	Position	S_{i1}	S_{i2}
P_1	lower right	$\begin{pmatrix} 0 & I_{h_1} \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ I_{w_1} & 0 \end{pmatrix}$
P_2	lower left	$\begin{pmatrix} 0 & I_{h_2} \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & I_{w_2} \\ 0 & 0 \end{pmatrix}$
P_3	upper right	$\begin{pmatrix} 0 & 0 \\ I_{h_3} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ I_{w_3} & 0 \end{pmatrix}$
P_4	upper left	$\begin{pmatrix} 0 & 0 \\ I_{h_4} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & I_{w_4} \\ 0 & 0 \end{pmatrix}$

Yeo [18] proposed a method that extracts DC and two AC coefficients (AC_{10}, AC_{01}) from P- or B-frames. If one restricts the summations in the evaluation of (6)–(8) for P- and B-frames to be performed only over $(l,\ m)$ such that $l+m \leq 1$, then one is essentially lowpass-filtering the anchor frame before motion-compensation is carried out. Fig. 4 represent the relationship

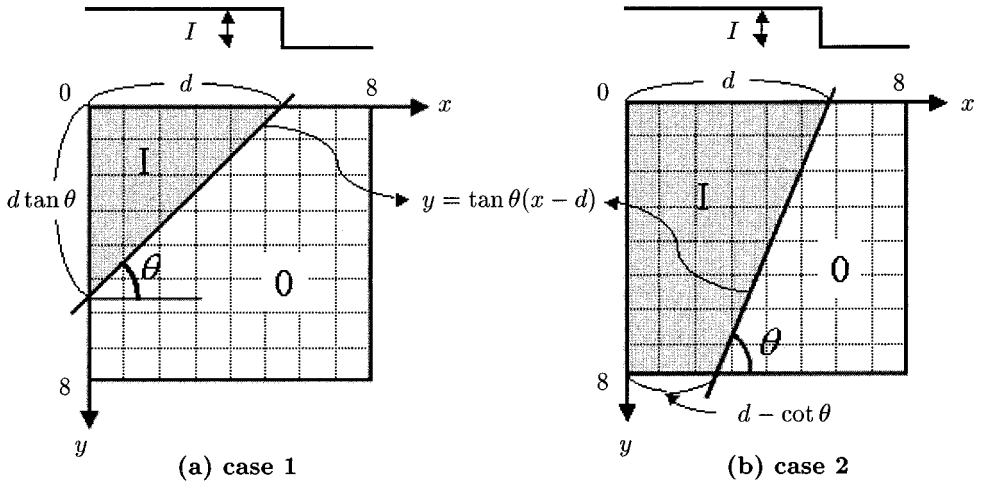

Fig. 6. Proposed ideal step edge model.

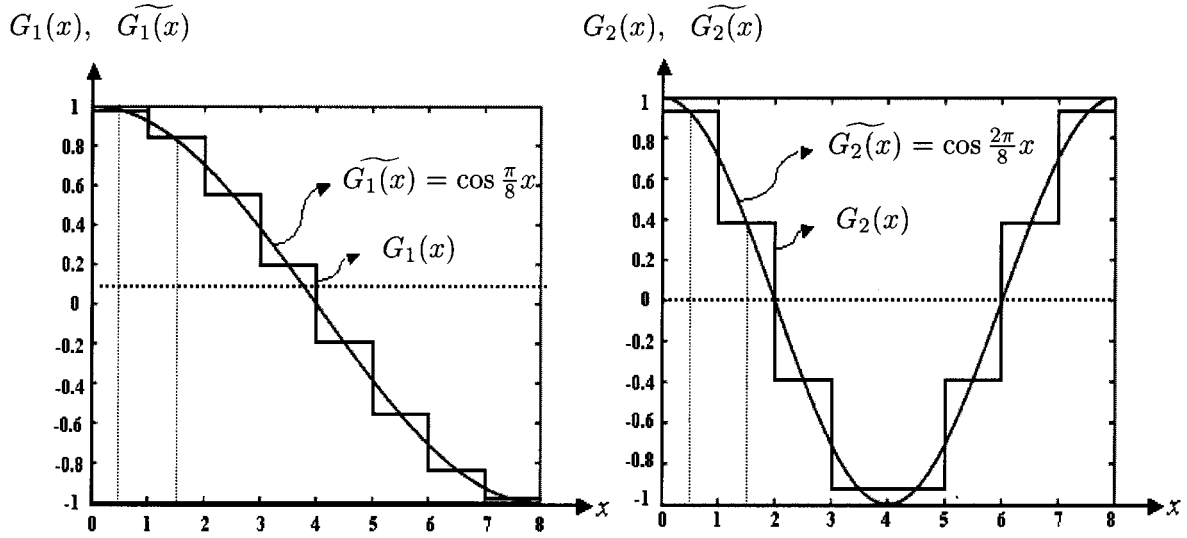

Fig. 7. Meaning of approximation for extracting AC coefficients (a single dimension).

between reference block P_{ref} and original blocks.

$$\mathrm{DCT}(P_{ref})_{00} = \sum_{i=1}^{4}\left(\sum_{m=0}^{7}\sum_{l=0}^{7} w_{ml}^{i}(\mathrm{DCT}(P_i))_{ml}\right) \tag{6}$$

$$\mathrm{DCT}(P_{ref})_{01} = \sum_{i=1}^{4}\left(\sum_{m=0}^{7}\sum_{l=0}^{7} \hat{w}_{ml}^{i}(\mathrm{DCT}(P_i))_{ml}\right) \tag{7}$$

$$\mathrm{DCT}(P_{ref})_{10} = \sum_{i=1}^{4}\left(\sum_{m=0}^{7}\sum_{l=0}^{7} \tilde{w}_{ml}^{i}(DCT(P_i))_{ml}\right) \tag{8}$$

where

$$\begin{aligned} w_{ml}^{i} &= \mathrm{DCT}(S_{i1}))_{0m}\ \mathrm{DCT}(S_{i2})_{l0} \\ \hat{w}_{ml}^{i} &= \mathrm{DCT}(S_{i1}))_{0m}\ \mathrm{DCT}(S_{i2})_{l1} \\ \tilde{w}_{ml}^{i} &= \mathrm{DCT}(S_{i1}))_{1m}\ \mathrm{DCT}(S_{i2})_{l0}. \end{aligned}$$

This image reconstruction is called a "dc +2AC image." If we use (6)–(8), each coefficient requires a maximum of 256 multiplicatioins, but if we use (9), each coefficient requires a maximum of only 12 multiplications. Therefore, we used this approximate reconstruction method for the extraction of DC and AC coefficients from P- and B-frames. Where, $(P)_{ij}$ represent the (i, j) component of P.

$$\begin{aligned} &(\mathrm{DCT}(P_{ref}))_{a,b} \\ &\quad\approx \sum_{i=1}^{4}\left(\sum_{m+l\le 1}^{7} c(h_i, w_i, a, m, l, b)\,(\mathrm{DCT}(P_i))_{m,l}\right) \end{aligned} \tag{9}$$

where

$$c(h_i, w_i, a, m, l, b) = \begin{cases} r_{a,m}^{h_i} l_{l,b}^{w_i}, & i = 1 \\ r_{a,m}^{h_i} r_{l,b}^{w_i}, & i = 2 \\ l_{a,m}^{h_i} l_{l,b}^{w_i}, & i = 3 \\ l_{a,m}^{h_i} r_{l,b}^{w_i}, & i = 4 \end{cases} \tag{10}$$

for $a+b \le 1$. Here, h_i and w_i are the height and width of block P_i in Fig. 4, respectively; $l_{i,j}^{n}$ and $r_{i,j}^{n}$ are the coefficients of matrices L_n and R_n. If we represent each block as an 8×8

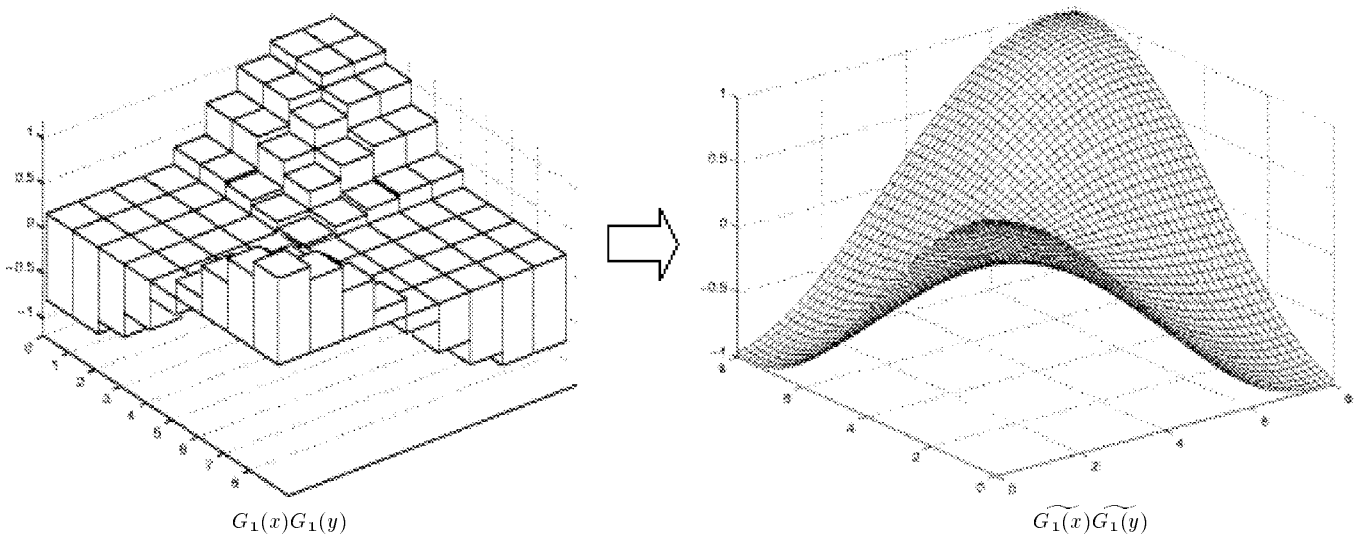

Fig. 8. Meaning of approximation for extracting AC coefficients (two dimensions).

matrix, then we can describe it in the spatial domain through matrix multiplication

$$P_{ref} = \sum_{i=1}^{4} S_{i1} P_i S_{i2}. \tag{11}$$

Here, S_{ij}'s are matrices of the following form:

$$L_n = \begin{pmatrix} 0 & 0 \\ I_n & 0 \end{pmatrix} \quad \text{or} \quad R_n = \begin{pmatrix} 0 & 0 \\ I_n & 0 \end{pmatrix}.$$

Each I_n is an identity matrix of size n. There are four possible locations for the subblock of interest: upper-left, upper-right, lower-right and lower-left. The actions in terms of matrices are tabulated in Table I.

IV. Proposed Scene Change Detection Algorithm

We can divide the proposed algorithm into direct edge information extraction and scene change detection by comparing two consecutive frames.

A. Direct Edge Information Extraction

MPEG—an international video compression standard—is based on 8 × 8 DCT. The DCT coefficients in each 8 × 8 DCT-block are related to the luminance or chrominance of 8 × 8 pixels in the spatial domain. In particular, the DC coefficient is the average of the luminance or chrominance signal of all pixels within the 8 × 8 block, and the AC coefficients essentially depend upon intensity differences in the vertical or horizontal direction [(5) and (4)].

1) Ideal Edge Model in DCT Domain: Using the DCT coefficients directly, Shen and Sethi [17] proposed an ideal step edge model and extracted areas and edges in a coded image. In the model shown in Fig. 5, they considered an ideal step edge cutting through a block of size 8, which is the standard size for JPEG/DCT. They are interested in relating three kinds of edge information: edge orientation, strength, and offset from center.

But the algorithm depends on experimental results and intuition rather than on mathematical formulation. In this paper,

$AC_{10}>0$, $AC_{01}>0$	$AC_{20}=0$		(1) case 2	$AC_{10}>0$, $AC_{01}<0$	$AC_{20}=0$		(11) case 2
	$AC_{02}=0$		(2) case 2		$AC_{02}=0$		(12) case 2
	$AC_{11}>0$		(3) case 1		$AC_{11}>0$		(13) case 1
	$AC_{11}<0$		(4) case 1		$AC_{11}<0$		(14) case 1
$AC_{10}<0$, $AC_{01}<0$	$AC_{20}=0$		(5) case 2	$AC_{10}<0$, $AC_{01}>0$	$AC_{20}=0$		(15) case 2
	$AC_{02}=0$		(6) case 2		$AC_{02}=0$		(16) case 2
	$AC_{11}>0$		(7) case 1		$AC_{11}>0$		(17) case 1
	$AC_{11}<0$		(8) case 1		$AC_{11}<0$		(18) case 1
$AC_{10}=0, AC_{01}>0$			(9) case 2	$AC_{10}\neq 0, AC_{01}>0$			(19) case 2
$AC_{10}=0, AC_{01}<0$			(10) case 2	$AC_{10}\neq 0, AC_{01}<0$			(20) case 2

Fig. 9. Proposed edge extraction rule using the correlation between AC coefficients.

we propose a new algorithm based on mathematical formulation which extracts edge information directly from MPEG video data. We consider orientation, strength and edge offset to be the important components definining the edge shape. Fig. 6 shows the proposed ideal step edge model. θ, d, I means orientation, offset and intensity value, respectively.

2) Approximation: We approximate G_λ by $\tilde{G_\lambda}$ which is defined by $\tilde{G_\lambda}(x) = \cos(\lambda\pi/8)x$, $(0 \le x \le 8)$. Fig. 7 shows G_1, G_2 and $\tilde{G_1}$, $\tilde{G_2}$. If u and v are small enough, we can approximate AC_{uv} by the following integral $\widetilde{\mathrm{AC}}_{uv}$.

$$\begin{aligned}\widetilde{AC}_{uv} &= \tfrac{1}{4} C_u C_v \int_0^8 \int_0^8 \tilde{G}_u(x)\tilde{G}_v(y) f(x,y)\,dx\,dy \\ &= \frac{1}{4} C_u C_v \int_0^8 \int_0^8 \cos\frac{u\pi}{8}x \cos\frac{v\pi}{8} y f(x,y)\,dx\,dy.\end{aligned} \tag{12}$$

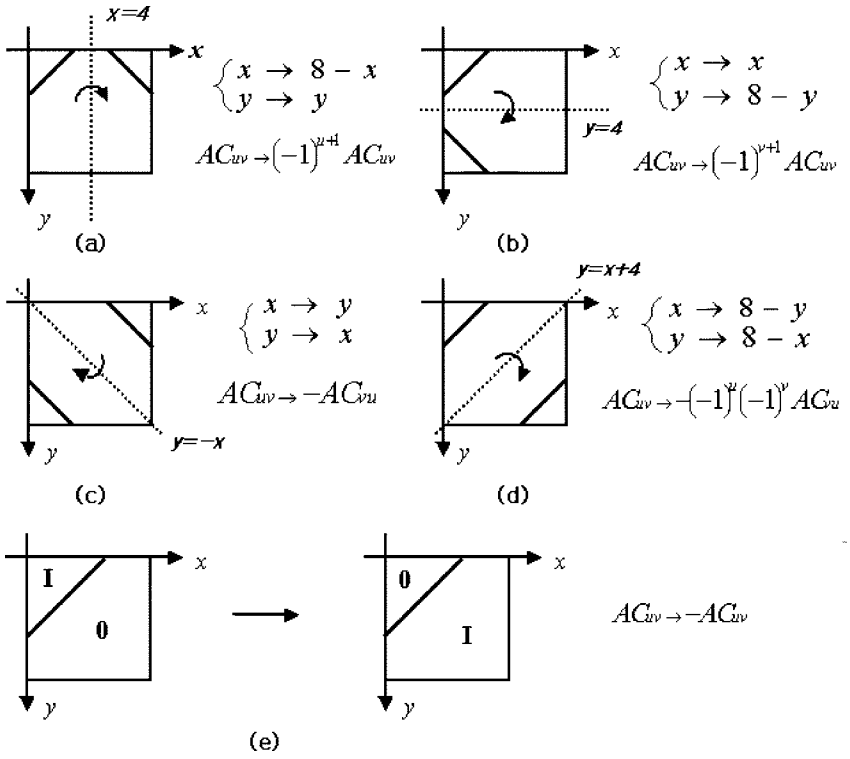

Fig. 10. Proposed symmetry rules for calculating AC coefficients.

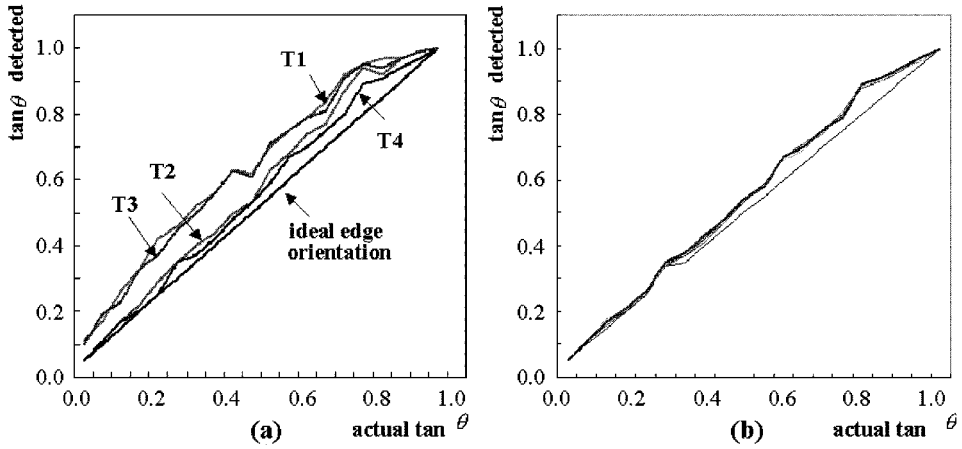

Fig. 11. Comparison of the method for detecting of edge orientation. (a) Real edge orientation versus detected edge orientation. (b) Performance of the proposed metric ($T4$) against edge strength: 50, 100, 150, 200.

Figs. 7 and 8 represent the meaning of approximation AC_{uv} by $\widetilde{\mathrm{AC}_{uv}}$. We will use only $u, v = 0, 1, 2$. In these cases, the approximation of AC_{uv} by $\widetilde{\mathrm{AC}_{uv}}$ is reasonable.The calculations in Section IV-B enable us to compute orientation ($\tan\theta$), strength (I) and offset (d) from the $\widetilde{\mathrm{AC}}$ coefficients, for case (2) and case (3) in Fig. 9. Since $\widetilde{\mathrm{AC}}$ coefficients are approximations to AC coefficients, we will denote $\widetilde{\mathrm{AC}}$ also by AC from now on. Note that we use only AC_{10}, AC_{20}, AC_{11}, AC_{01} and AC_{02}. We present different metrics of DCT coefficients to obtain accurate edge orientation, strength and offset information:

B. Calculation of $\widetilde{\mathrm{AC}}$ Coefficients: Approximation of AC Coefficients

We divide the cases of edge configurations as in Fig. 9. By exploiting the symmetry, it suffices to consider only two among all the cases in Fig. 9, namely case 2) and case 3). The conditions for the discrimination of the cases in Fig. 9 will be apparent, after we calculate the orientation, the strength and the offset in terms of the AC coefficients out linked in Section IV-C.

1) Case 1: (3) in Fig. 9: From Fig. 6(a) and (12), we have

$$\begin{aligned}\widetilde{\mathrm{AC}_{u0}} &= \frac{1}{4\sqrt{2}}\int_0^8\int_0^8 \cos\frac{u\pi}{8}\, x f(x,y)\,dx\,dy \\ &= \frac{I}{4\sqrt{2}}\int_0^d\left(\int_0^{-\tan\theta(x-d)}\cos\frac{u\pi}{8}\,x\,dy\right)dx \\ &= \frac{I}{4\sqrt{2}}\left(\frac{8}{u\pi}\right)^2\tan\theta\left(1-\cos\frac{u\pi}{8}\,d\right) \qquad (u\neq 0)\end{aligned} \tag{13}$$

$$\begin{aligned}\widetilde{\mathrm{AC}_{0v}} &= \frac{1}{4\sqrt{2}}\int_0^8\int_0^8 \cos\frac{v\pi}{8}\,yf(x,\,y)\,dx\,dy \\ &= \frac{I}{4\sqrt{2}}\int_0^d\left(\int_0^{-\tan\theta(x-d)}\cos\frac{v\pi}{8}\,y\,dy\right)dx \\ &= \frac{I}{4\sqrt{2}}\left(\frac{8}{v\pi}\right)^2\cot\theta \\ &\quad\cdot\left\{1-\cos\left(\frac{v\pi}{8}\,d\tan\theta\right)\right\} \qquad (v\neq 0)\end{aligned} \tag{14}$$

$$\begin{aligned}\widetilde{\mathrm{AC}_{11}} &= \frac{1}{8}\int_0^8\int_0^8 \cos\frac{\pi}{8}\,x\cos\frac{\pi}{8}\,yf(x,\,y)\,dx\,dy \\ &= \frac{I}{8}\int_0^d\left(\int_0^{-\tan\theta(x-d)}\cos\frac{\pi}{8}\,x\cos\frac{\pi}{8}\,y\,dy\right)dx \\ &= \frac{I}{8}\left(\frac{8}{\pi}\right)^2\frac{1}{2}\left(\frac{1}{1+\tan\theta}-\frac{1}{1-\tan\theta}\right) \\ &\quad\cdot\left\{\cos\frac{\pi}{8}\,d-\cos\frac{\pi}{8}(d\tan\theta)\right\}.\end{aligned} \tag{15}$$

From (13), we have

$$\widetilde{\mathrm{AC}_{10}} = \frac{1}{4\sqrt{2}}\left(\frac{8}{\pi}\right)^2\tan\theta\left(1-\cos\frac{\pi}{8}\,d\right), \tag{16}$$

$$\widetilde{\mathrm{AC}_{20}} = \frac{1}{4\sqrt{2}}\left(\frac{8}{2\pi}\right)^2\tan\theta\left(1-\cos\frac{2\pi}{8}\,d\right). \tag{17}$$

From (14), we have

$$\widetilde{\mathrm{AC}_{01}} = \frac{1}{4\sqrt{2}}\left(\frac{8}{\pi}\right)^2\cot\theta\left(1-\cos\frac{\pi}{8}\,d\tan\theta\right) \tag{18}$$

$$\widetilde{\mathrm{AC}_{02}} = \frac{1}{4\sqrt{2}}\left(\frac{8}{2\pi}\right)^2\cot\theta\left(1-\cos\frac{2\pi}{8}\,d\tan\theta\right). \tag{19}$$

2) Case 2: (2) in Fig. 9: From Fig. 6(b) and (12), we have

$$\begin{aligned}\widetilde{\mathrm{AC}_{u0}} &= \frac{1}{4\sqrt{2}}\int_0^8\int_0^8 \cos\frac{u\pi}{8}\,xf(x,\,y)\,dx\,dy \\ &= \frac{I}{4\sqrt{2}}\left[8\int_0^{d-8\cot\theta}\left(\int_0^8\cos\frac{u\pi}{8}\,x\,dy\right)dx\right. \\ &\qquad\left.-\tan\theta\int_{d-8\cot\theta}^d(x-d)\cos\frac{u\pi}{8}\,x\,dx\right] \\ &= -\frac{I}{4\sqrt{2}}\left(\frac{8}{u\pi}\right)^2\tan\theta \\ &\quad\cdot\left\{\cos\frac{u\pi}{8}-\cos\frac{u\pi}{8}(d-8\cot\theta)\right\} \qquad (u\neq 0)\end{aligned} \tag{20}$$

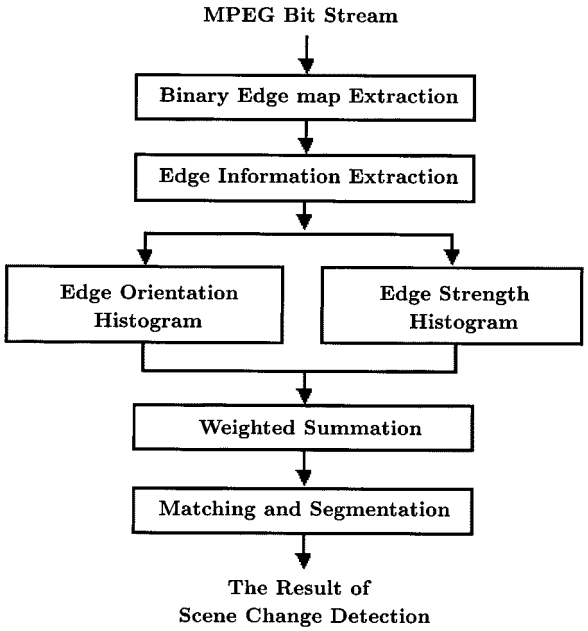

Fig. 12. Overview of the proposed frame matching.

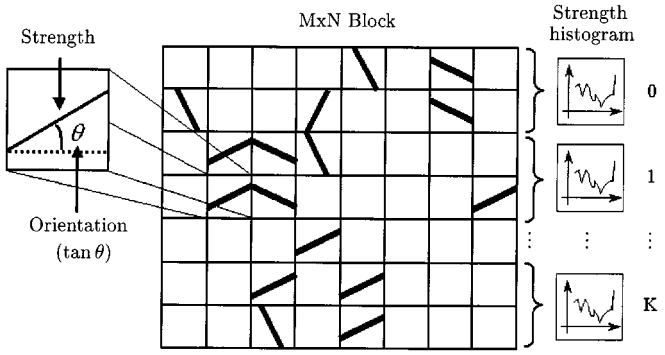

Fig. 13. Edge orientation and strength.

$$\begin{aligned}\widetilde{\mathrm{AC}_{0v}} &= \frac{1}{4\sqrt{2}}\int_0^8\int_0^8 \cos\frac{v\pi}{8}\,yf(x,\,y)\,dx\,dy \\ &= \frac{I}{4\sqrt{2}}\left[\int_0^{d-8\cot\theta}\left(\int_0^8\cos\frac{u\pi}{8}\,y\,dy\right)dx\right. \\ &\qquad\left.+\int_{d-8\cot\theta}^d\left(\int_0^{-\tan\theta(x-d)}\cos\frac{u\pi}{8}\,y\,dy\right)dx\right] \\ &= \frac{I}{4\sqrt{2}}\left(\frac{8}{v\pi}\right)^2\cot\theta(1-\cot v\pi) \qquad (v\neq 0)\end{aligned} \tag{21}$$

$$\begin{aligned}\widetilde{\mathrm{AC}_{11}} &= \frac{1}{8}\int_0^8\int_0^8 \cos\frac{\pi}{8}\,x\cos\frac{\pi}{8}\,yf(x,\,y)\,dx\,dy \\ &= \frac{I}{8}\left[\int_0^{d-8\cot\theta}\cos\frac{\pi}{8}\,x\left(\int_0^8\cos\frac{\pi}{8}\,y\,dy\right)dx\right. \\ &\qquad+\int_{d-8\cot\theta}^d\cos\frac{\pi}{8}\,x \\ &\qquad\left.\cdot\left(\int_0^{-\tan\theta(x-d)}\cos\frac{\pi}{8}\,y\,dy\right)dx\right] \\ &= \frac{I}{8}\left(\frac{8}{\pi}\right)^2\frac{1}{2}\left(\frac{1}{1+\tan\theta}-\frac{1}{1-\tan\theta}\right) \\ &\quad\cdot\left\{\cos\frac{\pi}{8}\,d+\cos\frac{\pi}{8}(d-8\cot\theta)\right\}.\end{aligned} \tag{22}$$

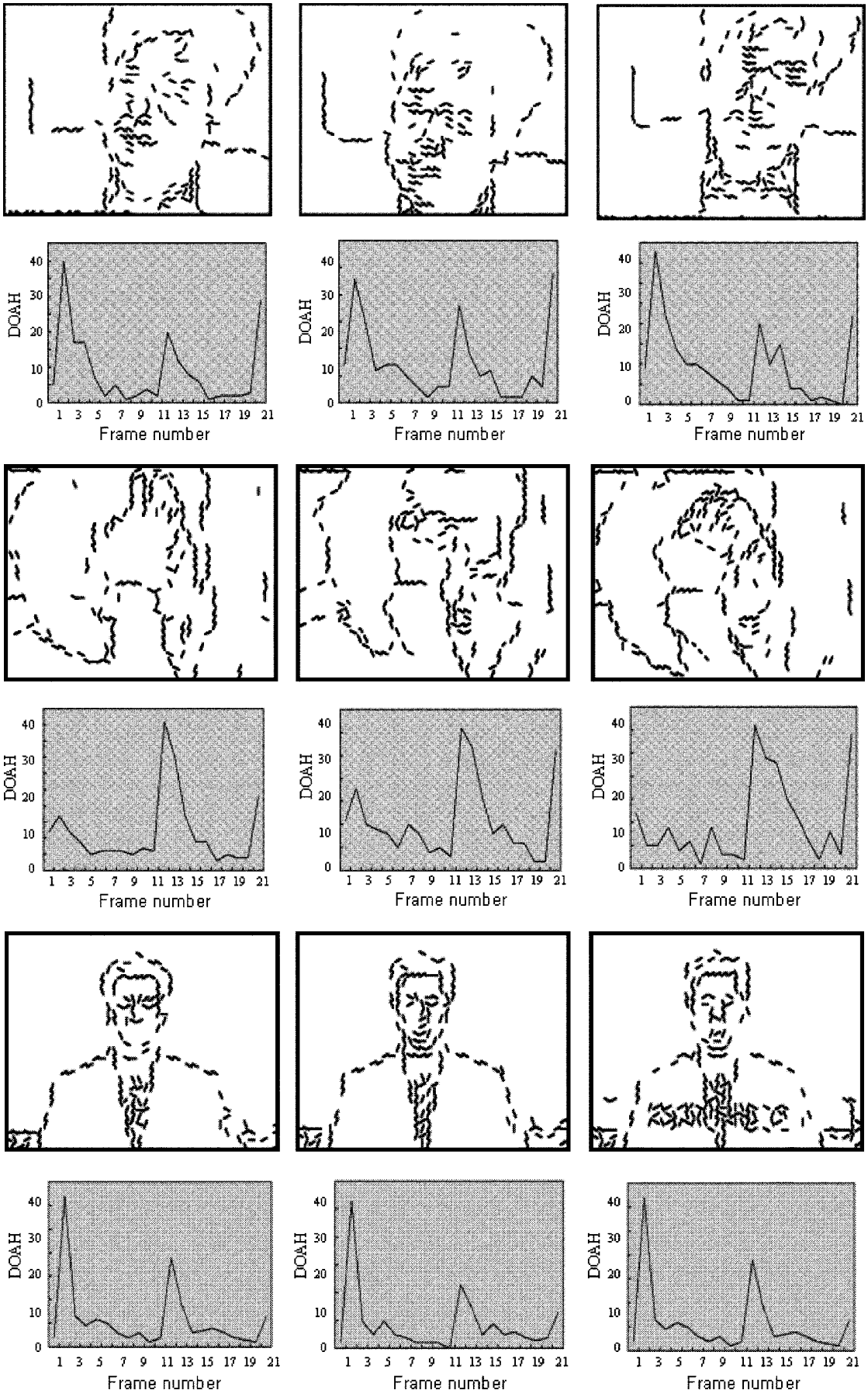

Fig. 14. Examples of orientation histogram.

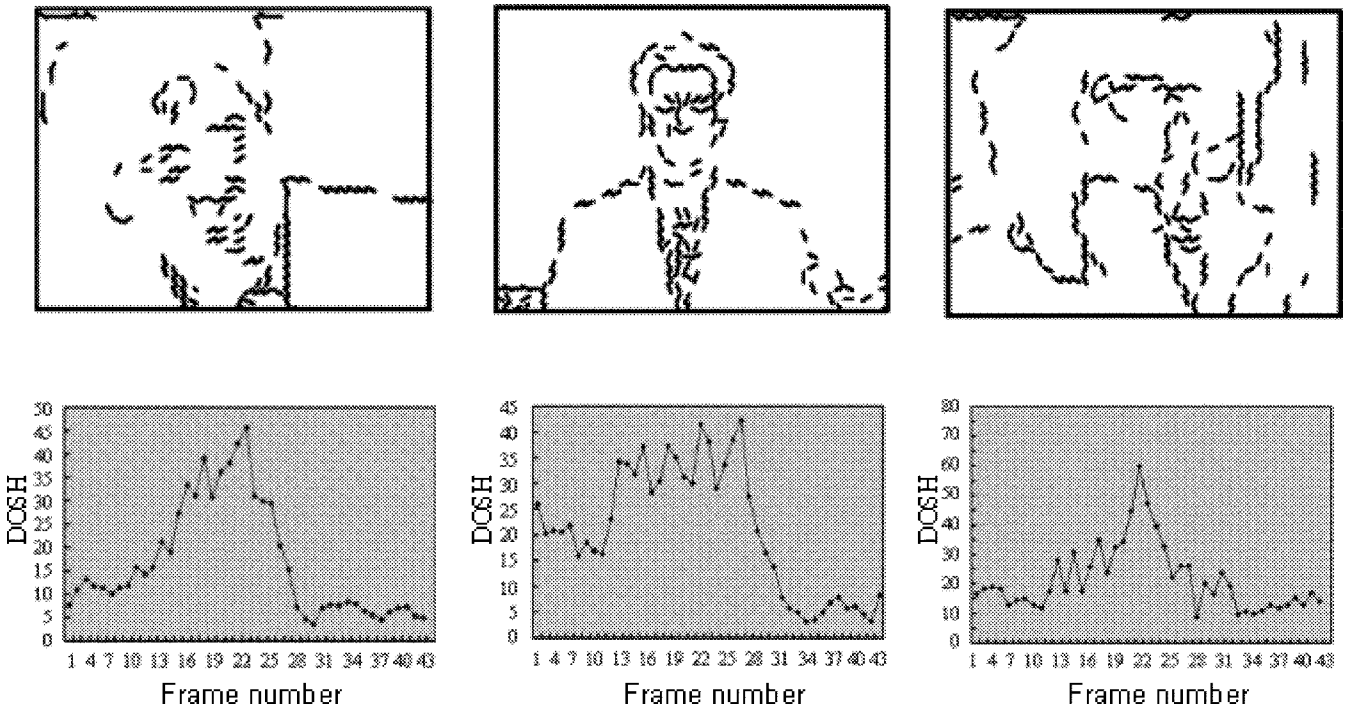

Fig. 15. Examples of strength histogram.

From (20), we have

$$\widetilde{\mathrm{AC}_{10}} = -\frac{1}{4\sqrt{2}}\left(\frac{8}{\pi}\right)^2 \cdot \tan\theta\left\{\cos\frac{\pi}{8}d - \cos\frac{\pi}{8}(d - 8\cot\theta)\right\} \tag{23}$$

$$\widetilde{\mathrm{AC}_{20}} = -\frac{1}{4\sqrt{2}}\left(\frac{8}{2\pi}\right)^2 \cdot \tan\theta\left\{\cos\frac{2\pi}{8}d - \cos\frac{2\pi}{8}(d - 8\cot\theta)\right\}. \tag{24}$$

From (21), we have

$$\widetilde{\mathrm{AC}_{01}} = \frac{1}{2\sqrt{2}}\left(\frac{8}{\pi}\right)^2 \cot\theta \tag{25}$$

$$\widetilde{\mathrm{AC}_{02}} = 0. \tag{26}$$

C. Calculation of Edge Information

1) Case 1: (3) in Fig. 9:

- offset (d)

From (16) and (17), we have

$$\frac{\mathrm{AC}_{20}}{\mathrm{AC}_{10}} = \frac{1}{4}\cdot\frac{1-\cos 2\frac{\pi}{8}d}{1-\cos\frac{\pi}{8}d} \tag{27}$$

$$\cos\frac{\pi}{8}d = 2\frac{\mathrm{AC}_{20}}{\mathrm{AC}_{10}} - 1 \tag{28}$$

$$d = \frac{8}{\pi}\arccos\left(2\frac{\mathrm{AC}_{20}}{\mathrm{AC}_{10}} - 1\right). \tag{29}$$

- orientation ($\tan\theta$)

TABLE II
VIDEO DATA FOR EXPERIMENTS

Type	Abbr.	Duration(hh:mm:ss)	Quantity	N_{total}	N_{Rcut}
News	New	0:15:43	173MB	28,280	186
Sitcom	Sit	0:15:01	165MB	27,017	142
Music Video	Mus	0:05:23	69MB	9,704	189
Documentary	Doc	0:10:35	90MB	15,886	94

From (16)–(18), we have

$$\tan\theta = \frac{\mathrm{AC}_{01}}{\mathrm{AC}_{10}}\sqrt{\frac{\mathrm{AC}_{01}-\mathrm{AC}_{20}}{\mathrm{AC}_{10}-\mathrm{AC}_{20}}}. \tag{30}$$

- strength (I)

From (16)–(19), we have

$$I = 2\sqrt{2}\left(\frac{\pi}{8}\right)^2\frac{\mathrm{AC}_{10}\mathrm{AC}_{01}}{\sqrt{(\mathrm{AC}_{10}-\mathrm{AC}_{20})(\mathrm{AC}_{01}-\mathrm{AC}_{02})}}. \tag{31}$$

2) Case 2: (2) in Fig. 9:

- offset (d)

From (22)–(25), we have

$$d = \frac{\mathrm{AC}_{20}}{\mathrm{AC}_{10}}\cdot\frac{\sqrt{2}\mathrm{AC}_{10}\mathrm{AC}_{11}-\mathrm{AC}_{20}\mathrm{AC}_{01}}{\sqrt{2}\mathrm{AC}_{01}\mathrm{AC}_{11}}. \tag{32}$$

- orientation ($\tan\theta$)

From (22)–(25), we have

$$\tan\theta = \sqrt{\frac{\sqrt{2}\mathrm{AC}_{10}\mathrm{AC}_{11}}{\sqrt{2}\mathrm{AC}_{10}\mathrm{AC}_{11}-\mathrm{AC}_{20}\mathrm{AC}_{01}}}. \tag{33}$$

- strength (I)

From (22)–(25), we have

$$I = 2\sqrt{2}\left(\frac{\pi}{8}\right)^2\mathrm{AC}_{01}\sqrt{\frac{\sqrt{2}\mathrm{AC}_{10}\mathrm{AC}_{11}}{\sqrt{2}\mathrm{AC}_{10}\mathrm{AC}_{11}-\mathrm{AC}_{20}\mathrm{AC}_{01}}}. \tag{34}$$

Fig. 16. Examples of extracted edge image from MPEG video data. (a) Original images, (b) example of extracted edge image using the proposed method, and (c) images which are filtered by the Sobel filter.

D. Symmetry Rules

From the results in Section IV-C, we can calculate the edge information for the rest of the cases in Fig. 9. This is achieved by using symmetry; for example, case 4) in Fig. 9 is obtained by applying the transformation (d) in Fig. 10, followed by the transformation (e) in Fig. 10. So, we need only to replace AC_{uv} by $(-1)^u(-1)^v\mathrm{AC}_{uv}$. Thus, for case 4), we have

- orientation ($\tan\theta$)

$$\tan\theta = \frac{-\mathrm{AC}_{10}}{-\mathrm{AC}01}\sqrt{\frac{-\mathrm{AC}_{10}-\mathrm{AC}_{02}}{-\mathrm{AC}_{01}-\mathrm{AC}_{02}}} \tag{35}$$

- strength (I)

$$I = 2\sqrt{2}\left(\frac{\pi}{8}\right)^2 \frac{(-\mathrm{AC}_{01})(-\mathrm{AC}_{10})}{\sqrt{(-\mathrm{AC}_{01}-\mathrm{AC}_{02})(-\mathrm{AC}_{10}-\mathrm{AC}_{20})}} \tag{36}$$

- offset (d)

$$d = \frac{8}{\pi}\arccos\left(2\frac{\mathrm{AC}_{02}}{-\mathrm{AC}_{01}}-1\right). \tag{37}$$

Of course, $\tan(\theta)$ and d here should be interpreted appropriately.

E. Comparison of Methods for Extracting Edge Information

We experimented with the proposed metric for extracting edge orientation. The result is shown in Fig. 11. For comparison purposes, we also show the edge orientation estimate for $T1$, $T2$, $T3$, and $T4$. The proposed algorithm, $T4$ provides the best edge orientation estimate and has little to do with edge strength. So, we use the proposed $T4$ metric for extracting the accurate edge orientation information. In Fig. 11, $T3$ and $T4$ represent the Sobel edge operator and the proposed metric, respectively. $T1$ and $T2$ are defined by

$$T1 = \frac{\mathrm{AC}_{01}}{\mathrm{AC}_{10}} \tag{38}$$

$$T2 = \frac{\sum_{v=1}^{7}\mathrm{AC}_{0v}}{\sum_{u=1}^{7}\mathrm{AC}_{u0}}. \tag{39}$$

TABLE III
COMPARISON OF THE PROPOSED METHOD WITH THE OTHERS

method	New			Sit			Doc			Mus		
	N_C	N_{FN}	N_{FP}	N_C	N_{FN}	N_{FP}	N_C	N_{FN}	N_{FP}	N_C	N_{FN}	N_{FP}
DC	181	5	8	141	1	5	90	4	12	183	6	23
FB	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

TABLE IV
ACCURACY COMPARISON OF SCENE CHANGE DETECTION METHODS VIA THE PRECISION AND RECALL PARAMETERS

method	New		Sit		Doc		Mus	
	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
DC	0.9577	0.9731	0.9658	0.9930	0.8824	0.9574	0.8883	0.9683
FB	0.9840	0.9946	0.9930	1.0000	0.9789	0.9894	0.9689	0.9894
PM	0.9734	0.9839	0.9930	0.9930	0.9479	0.9680	0.9541	0.9894

F. Frame Matching Phase

If we use the proposed direct edge detection algorithm mentioned in previous section, we can match two consecutive frames independent of luminance or color changes. Fig. 12 shows the

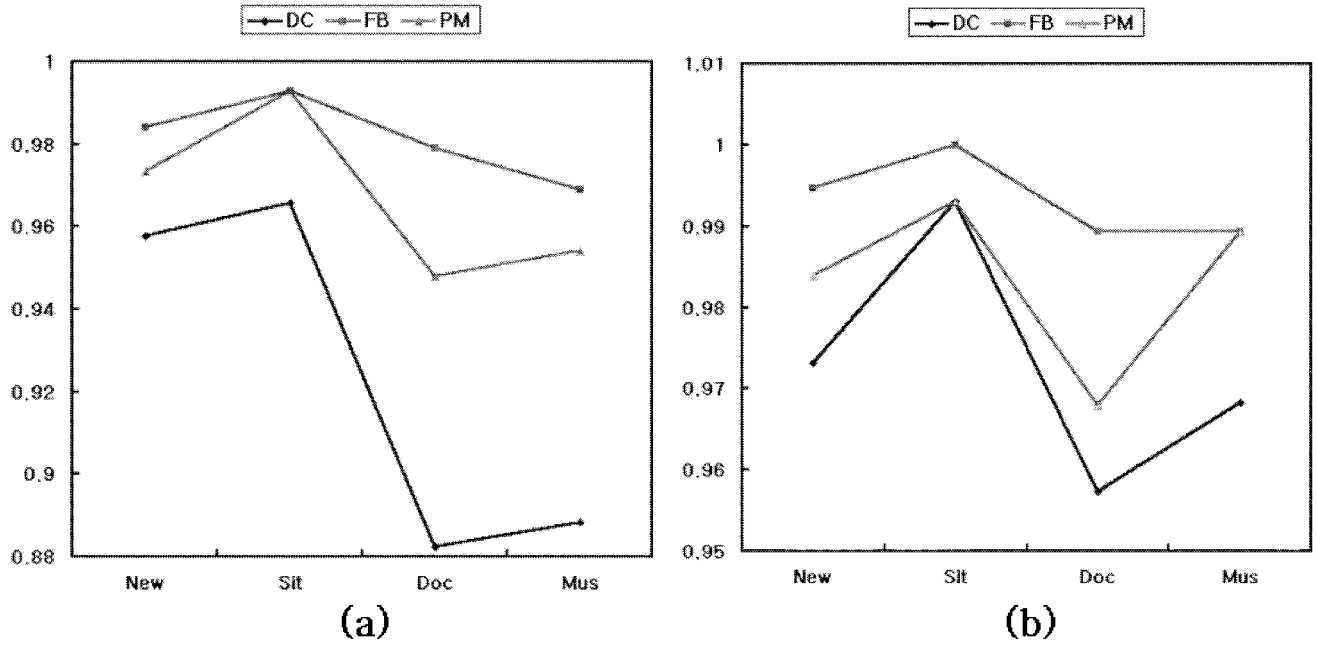

Fig. 17. Comparison of experimental results on sample video via the precision and recall parameters: (a) precision parameter and (b) recall parameter.

overview of the proposed frame matching. In this paper, we directly extract edge information (orientation and strength) from MPEG video data, and perform a comparison of two frames' orientation and strength histogram. Finally, we detect scene change frames using comparison results. Fig. 13 shows the example of the edge orientation and strength features.

1) Edge Orientation Histogram Comparison: The extracted edge in 8×8 blocks have their own orientation. Therefore, we can use edge orientation for frame matching. We can get the edge orientation histogram difference for frame matching using (40). f_n means nth frame, DOAH (f_n, f_{n+1}) means the difference of the angle histograms between the nth frame and the $(n+1)$th frame, Q_A means the number of phases in the orientation histogram, and $\mathrm{AH}_n(i)$ means ith orientation histogram of frame n. Fig. 14 shows examples of the orientation histogram.

$$\mathrm{DOAH}(f_n, f_{n+1}) = \sum_{i=0}^{Q_A-1} |\mathrm{AH}_n(i) - \mathrm{AH}_{n+1}(i)|. \quad (40)$$

2) Edge Strength Histogram Comparison: The information that we can extract from MPEG video data directly is not only the edge orientation but also the edge strength. If we use only the orientation histogram, the differences of edge orientation histograms are sensitive to camera speed or camera rotation. To improve this situation, our algorithm uses an edge strength histogram. We can get edge strength histogram using (41)–(43). 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 $\mathrm{ST}_n(i, j)$ means the edge strength of the (i, j)th block of the nth frame. Fig. 15 shows the examples of strength histograms. $\mathrm{DOSH}(f_n, f_{n+1})$ represents the difference of strength histogram between nth and $n+1$th frame.

$$Y_n(k) = \sum_{j=k(N/K)}^{\{(k+1)(N/K)\}-1} \sum_{i=0}^{M-1} \mathrm{ST}_n(i, j) \quad (41)$$

$$D_{nk} = |Y_n(k) - Y_{n+1}(k)| \quad (42)$$

TABLE V
SPEED COMPARISON OF THE SCENE CHANGE DETECTION METHODS

Type	News	Sitcom	Documentary	Music Video
DC	11.4 frame/sec.	10.3 frame/sec.	11.2 frame/sec.	10.8 frame/sec.
FB	2.1 frame/sec.	2.7 frame/sec.	2.3 frame/sec.	2.1 frame/sec.
PM	11.4 frame/sec.	11.3 frame/sec.	10.7 frame/sec.	10.8 frame/sec.

$$\mathrm{DOSH}(f_n, f_{n+1}) = \frac{\sum_{k=0}^{K-1} \varphi_{n,n+1}(k)}{K} \quad (43)$$

$$\varphi_{n,n+1}(k) \begin{cases} 1, & \text{if } D_{nk} > \Delta_T \\ 0, & \text{otherwise} \end{cases} \quad (44)$$

where Δ_T is the threshold.

3) Frame Matching Using Edge Orientation and Strength: In this paper, we match two consecutive frames using edge orientation and strength information. The offset information is a very important component of an edge. So, we perform frame matching using only edge orientation and strength information which are derived in the previous section. Equation (45) shows the weighted summation of orientation and strength histogram. In (45), f_n means nth frame, DOF (f_n, f_{n+1}) means the difference of the nth and the $(n+1)$th frames, and α means the weight $(0 \le \alpha \le 1)$.

$$\mathrm{DOF}(f_n, f_{n+1}) = (1-\alpha)\,\mathrm{DOAH}(f_n, f_{n+1}) + \alpha\,\mathrm{DOSH}(f_n, f_{n+1}) \quad (45)$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Environment

In order to evaluate the proposed scene change detection algorithm using direct edge information extraction, we performed our experiments using a Pentium II 350 MHz PC, under the Windows NT 4.0 operating system. We used various video data—Sitcom video data, News video data, and Documentary video data—of which the types and sizes are shown in Table II.

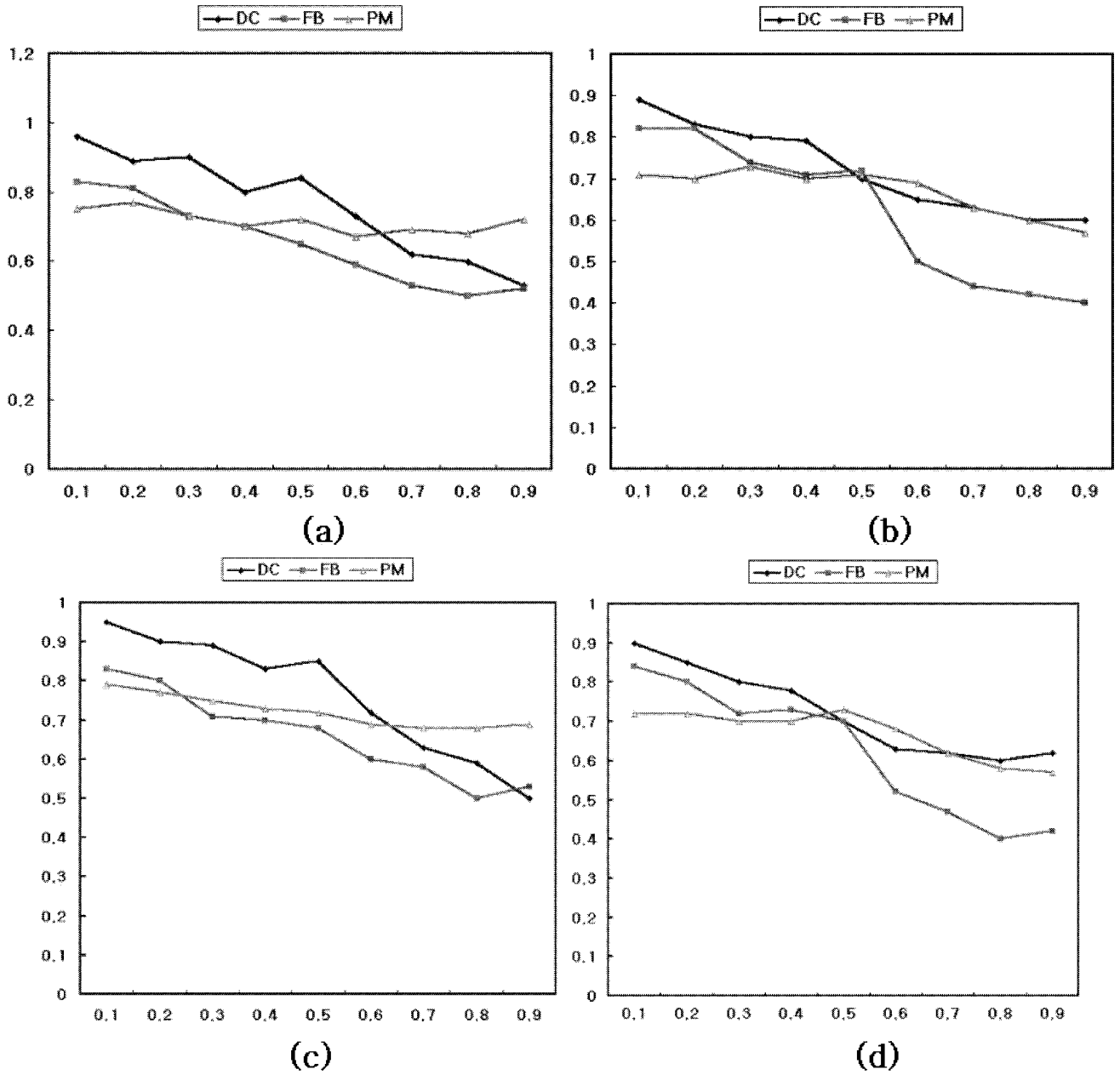

Fig. 18. Relationship between the precision and recall parameters. (a) News video data, (b) sitcom video data, (c) documentary video data, and (d) music video data.

These video data were collected by the Optibase MPEG Fusion System MPEG-1/2 Encoder. N_{total} represents the number of the total frames, N_{Rcut} represents the number of scene changes.

B. Experimental Results

The experimental results demonstrate the efficiency of the proposed scene change detection algorithm. Fig. 16(a) are the original images, Fig. 16(b) are the examples of the extracted edge images directly from MPEG video data, and Fig. 16(c) are the images which are filtered by the Sobel filter.

Table III shows the scene change detection results with experimental video data in Table II using the method of feature-based (FB) [10], the method of DC images (DC) [13], and the proposed method (PM). The method using edge-based features in the uncompressed domain proposed by Zabih, that is a very accurate scene change detection algorithm, and the method using DC image in compressed domain proposed by Yeo and Liu [13], is very promising and produces the best results among the previous works.

In Table III, N_C means the number of correct scene change detections, N_{FN} means the number of missed scene detections, and N_{FP} means the number of incorrectly detected scene change detections. The method using DC images is very sensitive to luminance and color change, so many false scene change frames were detected when we used music video data and documentary data. But the method using edge-based feature and the proposed method are not sensitive to luminance or color change.

$$Precision = \frac{N_C}{N_C + N_{FP}} \tag{46}$$

$$Recall = \frac{N_C}{N_C + N_{FN}}. \tag{47}$$

The performance is given in terms of precision and recall parameters defined in (46) and (47). Table IV and Fig. 17 show performance comparisons of the scene change detection methods via the precision and recall parameter and Fig. 18 represents the correlation between these precision and the recall parameters.

Table V demonstrates the speed comparisons of scene change detection methods. The method using the DC image and the proposed method are performed using direct feature extraction in the compressed domain, therefore these methods are faster than the method using the edge-based feature in the uncompressed domain. The experiments show that the method using DC images and the proposed method are five to six times faster than the method using edge-based features.

Two experiments demonstrate that the proposed method is not only more accurate but also faster than the previously known scene change detection algorithms such as FB and DC.

VI. 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. First, we derived binary edge maps from the AC coefficients in blocks which was discrete cosine transformed. Second, we measured edge orientation, strength and offset using the correlation between the AC coefficients in the derived binary edge maps. Finally, we matched two consecutive frames using these two features (edge orientation and strength). The accuracy of the proposed algorithm was shown to be comparable to the accuracy of the method using FB [10], and definitely higher than the accuracy of the method using DC images [13]. The proposed algorithm is comparable to the DC method in speed, and was found to be five to six times faster than the FB method. This was made possible by a new mathematical formulation for deriving the edge information directly from the DCT coefficients.

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] M.-S. Lee and S.-W. Lee, "Automatic video parsing using shot boundary detection and camera operation analysis," *Pattern Recognit.*, vol. 34, no. 3, 2001.

[2] A. Hampapur, R. Jain, and T. E. Weymouth, "Production model based digital video segmentation," *Multimedia Tools Applicat.*, vol. 1, no. 1, pp. 9–46, 1995.

[3] H. Zhang and S. W. Smoliar, "Developing power tools for video indexing and retrieval," in *Proc. SPIE'94—Storage and Retrieval for Image and Video Databases II*, vol. 2185, San Jose, CA, 1994, pp. 140–149.

[4] B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proc. SPIE'95—Digital Video Compression: Algorithm and Technologies*, vol. 2419, San Jose, CA, 1995, pp. 2–13.

[5] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.

[6] T. Kato *et al.*, "A sketch retrieval method for full color image database," in *Proc. 11th Int. Conf. Pattern Recognition*, 1992, pp. 530–533.

[7] E. Feig and S. Winograg, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, Sept. 1992.

[8] J. Meng, Y. Juan, and S.-F. Chang, "Scene change detection in a MPEG compressed video sequence," in *IS&T/SPIE Symp. Proc.*, vol. 2419, San Jose, 1995.

[9] D. Swanberg, C. F. Shu, and R. Jain, "Knowledge guided parsing in video database," in *Proc. SPIE'93—Storage and Retrieval for Image and Video Database*, vol. 1908, San Jose, CA, 1993, pp. 13–24.

[10] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proc. ACM Multimedia '95*, 1995, pp. 189–200.

[11] H. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools Applicat.*, vol. 1, pp. 89–111, 1995.

[12] F. Arman, A. Hsu, and M. Y. Chiu, "Image processing on encoded video sequences," *Multimedia Syst.*, vol. 1, pp. 211–219, 1994.

[13] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Systems Video Technol.*, vol. 5, no. 6, pp. 533–544, 1995.

[14] I. K. Sethi and N. V. Patel, "A statistical approach to scene change detection," in *IS&T SPIE: Storage and Retrieval for Image and Video Databases III*, vol. 2420, San Jose, CA, 1995, pp. 329–339.

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

[16] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video: Compression Standard*. London, U.K.: Chapman & Hall, 1996.

[17] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," in *IS&T SPIE: Storage and Retrieval for Image and Video Databases IV*, vol. 1995–1996, CA, pp. 33–49.

[18] B. Yeo, "Efficient processing of compressed images and video," Ph.D. dissertation, Elect. Eng. Dept., Princeton Univ., Princeton, NJ, January 1996.

Seong-Whan Lee (S'87–M'91–SM'96) received the B.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1984, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Seoul, in 1986 and 1989, respectively.

From February 1989 to February 1995, he was an Assistant Professor in the Department of Computer Science at Chungbuk National University, Cheongju, Korea. In March 1995, he joined the faculty of the Department of Computer Science and Engineering, Korea University, Seoul, as an Associate Professor. Currently, he is the Director of the Center for Artificial Vision Research (CAVR), supported by the Korean Ministry of Science and Technology. He has been the Co-Editor-in-Chief of the *International Journal on Document Analysis and Recognition* since 1998 and the Associate Editor of the *Pattern Recognition Journal*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *International Journal of Computer Processing of Oriental Languages* since 1997. His research interests include pattern recognition, computer vision and neural networks. He has more than 200 publications on these areas in international journals and conference proceedings.

Dr. Lee was the Winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He received the First Outstanding Young Researcher Award at the 2nd International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Professor Award from Chungbuk National University in 1994. He also received the Outstanding Research Award from the Korea Information Science Society in 1996. He was the Program Co-Chair of the 6th International Workshop on Frontiers in Handwriting Recognition, the 2nd International Conference on Multimodal Interface, the 17th International Conference on Computer Processing of Oriental Languages, the 7th International Conference on Neural Information Processing, and the 5th International Conference on Document Analysis and Recognition, and also the Workshop Co-Chair of the 3rd International Workshop on Document Analysis Systems and the 1st IEEE International Workshop on Biologically Motivated Computer Vision. He served on the program committees of several well-known international conferences. He is a Fellow of International Association for Pattern Recognition, a senior member of the IEEE Computer Society, and a life member of the Korea Information Science Society, the International Neural Network Society, and the Oriental Languages Computer Society.

Young-Min Kim was born in Seoul, Korea, in 1975. He received the B.S. and M.S. degrees in computer science and engineering from Korea University, Seoul, Korea, in 1998 and 2000, respectively.

He is currently with the Korea Institute for Defense Analyses, Seoul, as a Research Engineer. His research interests include image processing, pattern recognition, and multimedia computing.

Sung Woo Choi received the B.S. degree in physics from Seoul National University, Korea, in 1991, and the M.S. and Ph.D. degrees in mathematics from Seoul National University in 1993 and 1999, respectively.

From March 1999 to February 2000, he was with Center for Artificial Vision Research (CAVR), supported by the Korean Ministry of Science and Technology at Korea University, Seoul, as a Research Assistant Professor. Since March 2000, he is a Postdoctoral Fellow at the Max Planck Institute for Computer Science, Saarbrücken, Germany. His research interests include mathematical modeling, computer vision, and image processing.

Dr. Choi is a Member of Society for Industrial and Applied Mathematics (SIAM) and American Mathematical Society (AMS).