# Fast and Efficient Method for Block Edge Classification and Its Application in H.264/AVC Video Coding

Hongliang Li, *Member, IEEE*, King Ngi Ngan, *Fellow, IEEE*, and Zhenyu Wei, *Student Member, IEEE*

***Abstract*—Edge is an important feature in video classification which finds applications in video representation and coding. In H.264/AVC, intra-prediction mode decision (a computationally intensive process) is based on the orientation of the edges in the macroblock. In this paper, we first investigate the difference properties derived from three coefficients in the non-normalized Haar transform (NHT) domain and present a fast and efficient method to classify block edge using these properties. The proposed method significantly reduces the number of computational operations in the edge models determination with no multiplications and less addition operations. The use of edge classification for fast intra-prediction mode decision in H.264/AVC video coding is then presented. The experimental results show the effectiveness of the proposed method.**

***Index Terms*—AVC, block edge, edge detection, H.264, intra-prediction, non-normalized Haar transform (NHT), video coding.**

## I. Introduction

EDGE as a key visual feature plays an important role in digital image/video processing, pattern recognition, and computer vision. The task of identifying image edges efficiently is generally considered as a crucial step for the detection and segmentation of the objects of interest [1]–[10]. Unlike other visual features, such as color or motion, edge is usually employed as a powerful cue for shape description in the content-based image and video representation systems.

The well-known approaches, such as Canny edge detector [11], express edges as sets of pixels. To reduce the noise effects on the edge detection, the smoothing filter is first employed to perform the convolution on the image. Then the masks in different directions are applied to highlight regions with larger gradient magnitude for the smoothed image. It can be found that the edge detection involves many convolution operations that result in high computational complexity. In addition, parameterized edge pixels, such as linear expression, would be required to describe the edge feature in a given region of interest [12].

Manuscript received June 26, 2006; revised April 24, 2007. This work was supported in part by research grants from the Shun Hing Institute of Advanced Engineering, and the Chinese University of Hong Kong Focused Investment Scheme Project 1903003. This paper was recommended by Associate Editor I. Ahmad.

The authors are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong SAR (e-mail: hlli@ee.cuhk.edu.hk; knngan@ee.cuhk.edu.hk; zywei@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.918778

In order to distinguish the edges from the homogeneous regions and access these edge regions quickly, several approaches have been proposed based on the simplified edge models. To obtain a block-based image segmentation in terms of edge, monotone, or textured blocks [13], the block classifier was employed by considering the stochastic distribution of the gray level in the block [14]. In this work, each edge block was classified as horizontal, vertical, or diagonal edge (i.e., $\pi/4$ or $3\pi/4$). The decision whether the given image block is homogeneous or not is based on the log-likelihood calculation in terms of the average gray level and variance for assumed Gaussian distribution. Unlike the above method, other methods concentrate on the edge block classification of discrete cosine transform (DCT)-compressed images [15]–[18]. Based on an ideal edge model determined from the relative values of different DCT coefficients, a simple method was provided to estimate the strength and orientation of an edge in [15]. This method was extended in [16] by combining more DCT coefficients based on the analysis of the integral approximation of these coefficients. In [17], two categories of ideal linear edges cutting through a block were considered. This work introduced the general rules of edge classification from the normalized DCT coefficients. Recently, by adopting the similar idea in [15], the approximate edge classification method was derived to estimate four edge patterns from three DCT coefficients in [18].

In this paper, we propose a fast and efficient method to classify the block edge in terms of the non-normalized Haar transform (NHT) coefficients. The objective of using NHT is that it can help us to quickly assess the block edge with very low computational cost. Unlike the DCT algorithms, the NHT method not only employs the fast multiresolution structure but the Walsh-like transform. Because only addition and subtraction operations are involved in the 2-D basis functions, the computational complexity is greatly reduced in the computation of NHT coefficients. Using these NHT coefficients, 42 block edge models can be derived according to the simple geometrical properties of the coefficients. We then present a fast intra-prediction mode decision algorithm for H.264/AVC video coding based on the edge classification results. It demonstrates how the edge classification result can be applied to the mode decision with greater time saving and negligible loss of coding quality.

Compared with the traditional edge detection methods like Canny or Gabor wavelet based method, the advantage of the proposed algorithm using NHT is the very low computational load to classify the block edge model. On the other hand, high

computational complexity is usually found in Canny or Gabor wavelet method. For example, in order to perform the edge detection based on Gabor wavelet, the Gabor wavelet representation of an image is first calculated by the convolution of the image with the Gabor kernel at different orientations and scales. The second advantage is that our method only needs three NHT coefficients to identify 42 edge models according to the simple geometrical properties. Only addition and subtraction operations are required to calculate those coefficients. However, in order to obtain the block edge model, the change at each pixel must be computed for Canny or Gabor wavelet based method. For example, after the Gaussian lowpass filter is first employed to smooth the original image, the edge detectors in horizontal and vertical directions are applied to estimate the gradient magnitude at each pixel in Canny algorithm. Then, the histogram of edge directions in this block can be used to determine the dominant edge model [28]. Similarly, for the Gabor wavelet, the original image is first transformed into wavelet domain using different Gabor filters. Gabor coefficients in the transform domain are then used to detect edges. Finally, edge classification in a given block may be obtained in terms of the statistical features of the Gabor coefficients.

This paper is organized as follows. The fast edge classification algorithm will be presented in Section II. Experimental results are provided in Section III to support the efficiency of our proposed algorithm. The use of edge classification in H.264/AVC video coding will be given in Section IV. Finally, in Section V, conclusions are drawn and further research is presented.

## II. Fast and Efficient Edge Classification Algorithm

There has been a growing interest in integer wavelet transform due to its better computational efficiency and minimal memory usage [19]. The input data are first split into two signals corresponding to evenly and oddly indexed samples. Then, one of the samples is used to subtract from the other. After the difference, sum, and bit-shift operations, the lowpass and highpass coefficients are obtained. In this section, we will first briefly introduce the fast NHT, which is a type of integer wavelet transform. More details on this transform, and its applications to signal processing can be found in [20]–[22]. Then we present our block edge classification algorithm using the obtained NHT coefficients.

### A. Analysis of Block Properties for 2-D NHT Coefficients

Let $x(n), n = 0, \ldots, N-1$, with $N$ even, denote a sequence of integers. The one-dimensional (1-D) NHT can be represented by two sequences, the approximate $l(n)$ and detail coefficients $h(n)$, defined as follows:

$$\begin{aligned} l(n) &= x(2n) + x(2n+1), \quad n = 0, \ldots, \frac{N}{2} - 1 \\ h(n) &= x(2n) - x(2n+1), \quad n = 0, \ldots, \frac{N}{2} - 1. \end{aligned} \tag{1}$$

The 1-D inverse NHT transform is given by

$$\begin{aligned} x(2n) &= \frac{l(n) + h(n)}{2} \\ x(2n+1) &= x(2n) - h(n). \end{aligned} \tag{2}$$

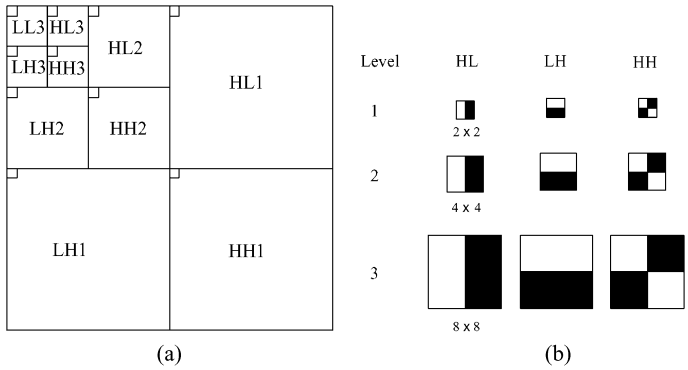


Fig. 1. (a) Multiresolution representation of 2-D NHT. (b) Block features at different level.

From (1), we can see that the NHT has the most efficient computational cost among the discrete transforms, such as Walsh and discrete Fourier transforms [23], [24]. The computation of the transform requires only integer additions and subtractions because its basis functions have values of only $+1$ and $-1$. For example, only 14 addition operations are needed to compute the three level NHT coefficients of eight discrete points, while 24 addition operations are required for Walsh transform. About 10 additions can be saved for the NHT transform. In comparison to the NHT, the well-known Walsh transform which also has only addition and subtraction operations has a higher computational cost.

The 2-D NHT is carried out by applying the transformation (1) sequentially to the rows and columns of the image. The corresponding hierarchical pyramid structure is shown in Fig. 1(a). It is noted that the same transformation is only applied to the reduced resolution LL subband to form the hierarchical pyramid. If this transformation is also performed in the high frequency subbands, such as HL, LH, and HH, the obtained NHT coefficients in the final decomposition level will exactly correspond to the result of Walsh transform. It means that the Walsh transform can be performed by the non-normalized Haar packet transform.

Let us consider the 2-D multiresolution structure, as shown in Fig. 1(a). It is found that the coefficients at the $\lambda$th level correspond to the $2^\lambda \times 2^\lambda$ pixel blocks due to the constant decomposition structure given in (1). For example, the coefficients in the subband LL3 are the sum of intensities of the $8 \times 8$ pixel blocks, and they also form another image with half the resolution, which is decomposed by the same transformation to obtain the hierarchical pyramid. In other words, if the $4 \times 4$ block information is required, we only need to find the coefficients at second level in the transform domain. Therefore, a good mapping relationship between the coefficients and the blocks can be constructed based on this multiresolution structure. Furthermore, for different subbands, i.e., LL, LH, HL, and HH, difference information corresponding to the original image can be observed. As shown in the Fig. 1(b), the same subband at different levels, such as LH1, LH2, LH3,..., represents the gray level variation in the same way. The coefficients in LH subband denote the gray level change in the vertical direction, while the HL and HH subbands represent the changes in the horizontal and diagonal directions, respectively. Let $M$ and $N$ denote the height and width of the image, respectively. From (1), the 2-D

NHT coefficients in different subbands at level $\lambda$ can be calculated by

$$\begin{aligned}\mathrm{LL}_\lambda(j,k) &= \sum_{m=2j}^{2j+1}\sum_{n=2k}^{2k+1}\mathrm{LL}_{\lambda-1}(m,n)\\ &= \sum_{m=2^i j}^{2^i(j+1)-1}\sum_{n=2^i k}^{2^i(k+1)-1}\mathrm{LL}_{\lambda-i}(m,n)\\ &= \sum_{m=2^\lambda j}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k}^{2^\lambda(k+1)-1}x(m,n)\\ &\quad j=0,1,\ldots,\frac{M}{2^\lambda}-1; k=0,1,\ldots,\frac{N}{2^\lambda}-1 \qquad (3)\end{aligned}$$

$$\begin{aligned}\mathrm{LH}_\lambda(j,k) &= \sum_{m=2^i j}^{2^i j+2^{i-1}-1}\sum_{n=2^i k}^{2^i(k+1)-1}\mathrm{LL}_{\lambda-i}(m,n)\\ &\quad - \sum_{m=2^i j+2^{i-1}}^{2^i(j+1)-1}\sum_{n=2^i k}^{2^i(k+1)-1}\mathrm{LL}_{\lambda-i}(m,n)=\cdots\\ &= \sum_{m=2^\lambda j}^{2^\lambda j+2^{\lambda-1}-1}\sum_{n=2^\lambda k}^{2^\lambda(k+1)-1}x(m,n)\\ &\quad - \sum_{m=2^\lambda j+2^{\lambda-1}}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k}^{2^\lambda(k+1)-1}x(m,n),\\ &\quad j=0,1,\ldots,\frac{M}{2^\lambda}-1, k=0,1,\ldots,\frac{N}{2^\lambda}-1 \qquad (4)\end{aligned}$$

$$\begin{aligned}\mathrm{HL}_\lambda(j,k) &= \sum_{m=2^i j}^{2^i(j+1)-1}\sum_{n=2^i k}^{2^i k+2^{i-1}-1}\mathrm{LL}_{\lambda-i}(m,n)\\ &\quad - \sum_{m=2^i j}^{2^i(j+1)-1}\sum_{n=2^i k+2^{i-1}}^{2^i(k+1)-1}\mathrm{LL}_{\lambda-i}(m,n)=\cdots\\ &= \sum_{m=2^\lambda j}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k}^{2^\lambda k+2^{\lambda-1}-1}x(m,n)\\ &\quad - \sum_{m=2^\lambda j}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k+2^{\lambda-1}}^{2^\lambda(k+1)-1}x(m,n),\\ &\quad j=0,1,\ldots,\frac{M}{2^\lambda}-1; k=0,1,\ldots,\frac{N}{2^\lambda}-1 \qquad (5)\end{aligned}$$

$$\begin{aligned}\mathrm{HH}_\lambda(j,k) &= \sum_{m=2^\lambda j}^{2^\lambda j+2^{\lambda-1}-1}\sum_{n=2^\lambda k}^{2^\lambda k+2^{\lambda-1}-1}x(m,n)\\ &\quad + \sum_{m=2^\lambda j+2^{\lambda-1}}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k+2^{\lambda-1}}^{2^\lambda(k+1)-1}x(m,n)\\ &\quad - \sum_{m=2^\lambda j+2^{\lambda-1}}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k}^{2^\lambda k+2^{\lambda-1}-1}x(m,n)\\ &\quad - \sum_{m=2^\lambda j}^{2^\lambda j+2^{\lambda-1}-1}\sum_{n=2^\lambda k+2^{\lambda-1}}^{2^\lambda(k+1)-1}x(m,n),\\ &\quad j=0,1,\ldots,\frac{M}{2^\lambda}-1; k=0,1,\ldots,\frac{N}{2^\lambda}-1. \qquad (6)\end{aligned}$$

Here, $\lambda$ ranges from 0 to $\min\{\log_2(M),\log_2(N)\}$. From (3) to (6), we can capture the relationship between the NHT coefficients and block structures. The approximate coefficients corresponding to zero horizontal and vertical frequency, called the LL coefficients, contain most of the signal energy in the $2^\lambda\times 2^\lambda$ block data. The detail coefficients in LH, HL, and HH, which are equivalent to the ac coefficients, represent the variations in intensities in certain direction at a certain rate. Now, we will employ these difference information involved in the detail coefficients to estimate the edge features in the block data.

### *B. Fast Block Edge Classification Based on NHT Coefficients*

In this section, we propose a fast algorithm to extract the edge information directly from the NHT domain, which offers better efficiency in computation. The difference properties of the NHT coefficients are used to provide the classification criteria.

Let $C_\lambda(j,k)$ denote the $2^\lambda\times 2^\lambda$ block with respect to the four subband coefficients $(j,k)$ at level $\lambda$. For example, $C_4(1,2)$ denotes the 16 × 16 data block, which corresponds to the coefficients $\mathrm{LL}_4(1,2)$, $\mathrm{LH}_4(1,2)$, $\mathrm{HL}_4(1,2)$, and $\mathrm{HH}_4(1,2)$. It is known that an edge is generated by the intensity changes in a certain direction. Therefore, two important features can be used to describe the edge, namely edge orientation and edge strength. Firstly, let us concentrate on the detection of edge orientation in the given block. To simplify, let $B_0$, $B_1$, $B_2$, and $B_3$ denote the sum of the intensities of four nonoverlap subbands in the given block $C_\lambda(j,k)$, respectively, which are defined as

$$\begin{aligned}B_0 &= \sum_{m=2^\lambda j}^{2^\lambda j+2^{\lambda-1}-1}\sum_{n=2^\lambda k}^{2^\lambda k+2^{\lambda-1}-1}x(m,n)\\ B_1 &= \sum_{m=2^\lambda j}^{2^\lambda j+2^{\lambda-1}-1}\sum_{n=2^\lambda k+2^{\lambda-1}}^{2^\lambda(k+1)-1}x(m,n)\\ B_2 &= \sum_{m=2^\lambda j+2^{\lambda-1}}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k}^{2^\lambda k+2^{\lambda-1}-1}x(m,n)\\ B_3 &= \sum_{m=2^\lambda j+2^{\lambda-1}}^{2^\lambda(j+1)-1}\sum_{n=2^\lambda k+2^{\lambda-1}}^{2^\lambda(k+1)-1}x(m,n). \qquad (7)\end{aligned}$$

Then, the NHT coefficients given in (3)–(6) can be rewritten as

$$\mathrm{LL}_\lambda(j,k) = B_0+B_1+B_2+B_3 \qquad (8)$$

$$\mathrm{LH}_\lambda(j,k) = B_0+B_1-B_2-B_3 \qquad (9)$$

$$\mathrm{HL}_\lambda(j,k) = B_0-B_1+B_2-B_3 \qquad (10)$$

$$\mathrm{HH}_\lambda(j,k) = B_0-B_1-B_2+B_3. \qquad (11)$$

As shown in Fig. 2, let $\theta$ denote the edge orientation in the given block. Based on the above difference information in the coefficients, we can derive the following rules for its estimation.

*Rule 1*: if $\mathrm{LH}_\lambda(j,k)\cdot\mathrm{HL}_\lambda(j,k) > 0$, then the edge direction in the block $C_\lambda(j,k)$ is likely to be the auxiliary diagonal, namely, $\theta \le (\pi/2)$.

*Rule 2*: if $\mathrm{LH}_\lambda(j,k)\cdot\mathrm{HL}_\lambda(j,k) < 0$, then the edge direction in the block $C_\lambda(j,k)$ is likely to be the principal diagonal, namely, $\theta > (\pi/2)$.

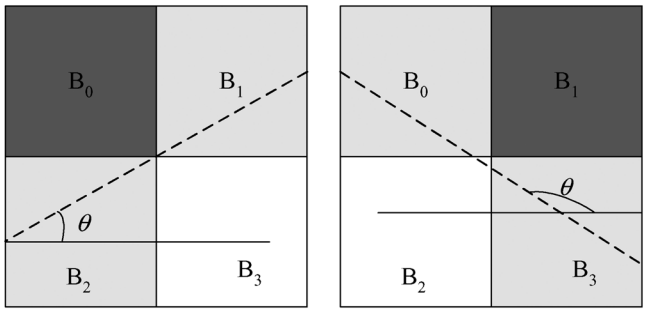


Fig. 2. Possible edge orientation in the given block. (a) $\theta \leq (\pi/2)$. (b) $\theta > (\pi/2)$.

We can see that if the condition $\mathrm{LH}_\lambda(j,k) \cdot \mathrm{HL}_\lambda(j,k) > 0$ is satisfied, then we have

$$\begin{cases} B_0 + B_1 > B_2 + B_3 \\ B_0 + B_2 > B_1 + B_3 \end{cases} \text{ or } \begin{cases} B_0 + B_1 < B_2 + B_3 \\ B_0 + B_2 < B_1 + B_3 \end{cases}$$
$$\Longrightarrow \quad B_0 > B_3 \quad \text{or} \quad B_0 < B_3 \quad \Longrightarrow \quad \theta \leq \frac{\pi}{2}. \tag{12}$$

It means that when the intensity level in the subblock $B_0$ is higher or lower than $B_3$, there exists a intensity change from top-left to bottom-right for the block, as shown in Fig. 2(a). Therefore, the possible edge direction for this case should be the auxiliary diagonal, namely, $\theta \leq (\pi/2)$. The result for $\theta > (\pi/2)$ can be easily derived from *Rule 2* using the similar approach.

Certainly, it is not enough to describe an edge in the block using only the coarse orientation information. In order to obtain more accurate edge information, we assume that there is an ideal edge cutting through a $2^\lambda \times 2^\lambda$ block. The similar assumption in DCT domain can also be found in [17].

If EM stands for the edge model in the block $C_\lambda(j,k)$, the edge models that satisfy the conditions of *Rule* 1 and *Rule* 2 will be denoted as EMA (auxiliary diagonal) and EMP (principal diagonal), respectively. For the case of $\theta \leq (\pi/2)$, there are two edge models namely EMA-I and EMA-II, which correspond to the first and the second rows of Fig. 3, respectively. In each model, two homogeneous regions are employed to represent the step edge in the block, namely the gray and white areas. $I_1$ and $I_2$ are used to denote the average intensity values of the white and gray regions, respectively. Let $d_{x1}$, $d_{x2}$, $d_{y1}$, and $d_{y2}$ be the edge offsets from the block boundary, and $S_1$, $S_2$, and $S_0$ the area values. Without loss of generity, we can use the integral operation to approximate the sum computation in (3)–(6). Therefore, for the case of EMA-I shown in the first row of Fig. 3, we have

$$\begin{aligned} \mathrm{LL}_\lambda(j,k) &= (S_0 + S_1 + S_2)I_1 + (2^{\lambda+1} - S_1 - S_2 - S_0)I_2 \\ \mathrm{LH}_\lambda(j,k) &= (S_0 + S_1 - S_2)(I_1 - I_2) \\ \mathrm{HL}_\lambda(j,k) &= (S_0 - S_1 + S_2)(I_1 - I_2) \\ \mathrm{HH}_\lambda(j,k) &= (S_0 - S_1 - S_2)(I_1 - I_2). \end{aligned} \tag{13}$$

The three area values are given by

$$\begin{aligned} S_0 &= \frac{\mathrm{LH}_\lambda(j,k) + \mathrm{HL}_\lambda(j,k)}{2(I_1 - I_2)} \\ S_1 &= \frac{\mathrm{LH}_\lambda(j,k) - \mathrm{HH}_\lambda(j,k)}{2(I_1 - I_2)} \\ S_2 &= \frac{\mathrm{HL}_\lambda(j,k) - \mathrm{HH}_\lambda(j,k)}{2(I_1 - I_2)}. \end{aligned} \tag{14}$$

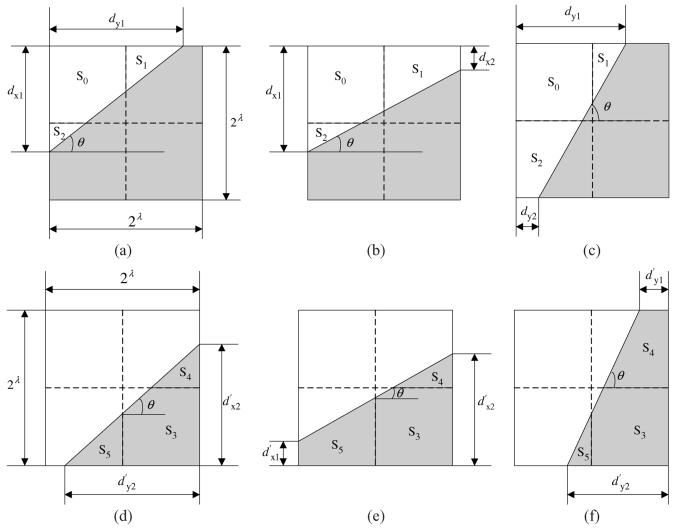


Fig. 3. Two edge models in the given block for the case of $\theta \leq (\pi/2)$. (a)–(c) First model with $S_0 \leq 2^{2\lambda-2}$, namely EMA-IA, EMA-IB, and EMA-IC. (d)–(f) Second model with $S_3 \leq 2^{2\lambda-2}$, namely EMA-IIA, EMA-IIB, and EMA-IIC.

Similarly, for the case of EMA-II corresponding to the second row of Fig. 3, we have

$$\begin{aligned} \mathrm{LL}_\lambda(j,k) &= (S_3 + S_4 + S_5)I_2 + (2^{\lambda+1} - S_3 - S_4 - S_5)I_1 \\ \mathrm{LH}_\lambda(j,k) &= (S_3 - S_4 + S_5)(I_1 - I_2) \\ \mathrm{HL}_\lambda(j,k) &= (S_3 + S_4 - S_5)(I_1 - I_2) \\ \mathrm{HH}_\lambda(j,k) &= (S_3 - S_4 - S_5)(I_2 - I_1). \end{aligned} \tag{15}$$

The corresponding area values can be calculated by

$$\begin{aligned} S_3 &= \frac{\mathrm{LH}_\lambda(j,k) + \mathrm{HL}_\lambda(j,k)}{2(I_1 - I_2)} \\ S_4 &= \frac{\mathrm{HL}_\lambda(j,k) + \mathrm{HH}_\lambda(j,k)}{2(I_1 - I_2)} \\ S_5 &= \frac{\mathrm{LH}_\lambda(j,k) + \mathrm{HH}_\lambda(j,k)}{2(I_1 - I_2)}. \end{aligned} \tag{16}$$

According to (14) and (16), the classification criterion for the two edge models EMA-I and EMA-II can be given as follows:

$$\mathrm{EMA} = \begin{cases} \mathrm{EMA-I}, & \text{if } \mathrm{HH}_\lambda(j,k) \geq 0 \text{ and } I_1 > I_2 \\ & \text{or } \mathrm{HH}_\lambda(j,k) \leq 0 \text{ and } I_1 < I_2 \\ \mathrm{EMA-II}, & \text{if } \mathrm{HH}_\lambda(j,k) < 0 \text{ and } I_1 > I_2 \\ & \text{or } \mathrm{HH}_\lambda(j,k) > 0 \text{ and } I_1 < I_2 \end{cases} \tag{17}$$

with

$$\begin{aligned} I_1 > I_2, &\quad \text{if } \mathrm{LH}_\lambda(j,k) > 0 \text{ and } \mathrm{HL}_\lambda(j,k) > 0 \\ I_1 < I_2, &\quad \text{if } \mathrm{LH}_\lambda(j,k) < 0 \text{ and } \mathrm{HL}_\lambda(j,k) < 0 \end{aligned} \tag{18}$$

The derivation of (17) is given in the Appendix at the end of this paper.

From Fig. 3, it is found that three cases should be investigated with respect to each of the edge models EMA-I or II. Here, we simply use EMA-I(A, B, C), and EMA-II(A, B, C) to denote the corresponding models given in Fig. 3. For the edge model

belonging to EMA-I, further classifications can be achieved by the following rules:

$$\mathrm{EMA-I}=\begin{cases}\mathrm{EMA-IA}, & \text{if } u1 \text{ or } u2 \text{ is satisfied}\\ \mathrm{EMA-IB}, & \text{if } u3 \text{ or } u4 \text{ is satisfied}\\ \mathrm{EMA-IC}, & \text{if } u5 \text{ or } u6 \text{ is satisfied}\end{cases} \tag{19}$$

where the detailed conditions are defined as

$$u1=\begin{cases}P1\geq 0\\ P2\geq 0\\ I_1>I_2\end{cases},\quad u2=\begin{cases}P1\leq 0\\ P2\leq 0\\ I_1<I_2\end{cases},\quad u3=\begin{cases}P1<0\\ P2\geq 0\\ I_1>I_2\end{cases}$$
$$u4=\begin{cases}P1>0\\ P2\leq 0\\ I_1<I_2\end{cases},\quad u5=\begin{cases}P1\geq 0\\ P2<0\\ I_1>I_2\end{cases},\quad u6=\begin{cases}P1\leq 0\\ P2>0\\ I_1<I_2\end{cases} \tag{20}$$

with

$$\begin{aligned}P1&=\mathrm{HL}_\lambda(j,k)-\mathrm{LH}_\lambda(j,k)+\mathrm{HH}_\lambda(j,k)\\ P2&=\mathrm{LH}_\lambda(j,k)-\mathrm{HL}_\lambda(j,k)+\mathrm{HH}_\lambda(j,k).\end{aligned} \tag{21}$$

Similarly, we can obtain the classification algorithm for the edge model EMA-II, which has the same rules as (19) except that the $P_1$ and $P_2$ are replaced by

$$\begin{aligned}P1&=\mathrm{HL}_\lambda(j,k)-\mathrm{LH}_\lambda(j,k)-\mathrm{HH}_\lambda(j,k)\\ P2&=\mathrm{LH}_\lambda(j,k)-\mathrm{HL}_\lambda(j,k)-\mathrm{HH}_\lambda(j,k).\end{aligned} \tag{22}$$

The derivation of (19) will be given in the Appendix at the end of this paper. It should be noted that this derivation can also be applied to other similar edge models.

As shown in Fig. 3, we can see that there are still many changes for every edge model with respect to different edge offsets. In this paper, we use simple methods to get the coarse offsets in the $2^\lambda \times 2^\lambda$ EMA block, namely

$$\begin{cases}0<d_{x1}\leq 2^{\lambda-1}, & \text{if } \mathrm{HL}_\lambda(j,k)=\mathrm{HH}_\lambda(j,k)\\ 2^{\lambda-1}<d_{x1}\leq 2^\lambda, & \text{otherwise}\end{cases}$$
$$\begin{cases}0<d_{y1}\leq 2^{\lambda-1}, & \text{if } \mathrm{LH}_\lambda(j,k)=\mathrm{HH}_\lambda(j,k)\\ 2^{\lambda-1}<d_{y1}\leq 2^\lambda, & \text{otherwise}\end{cases}$$
$$\begin{cases}0<d'_{x2}\leq 2^{\lambda-1}, & \text{if } \mathrm{HL}_\lambda(j,k)=-\mathrm{HH}_\lambda(j,k)\\ 2^{\lambda-1}<d'_{x2}\leq 2^\lambda, & \text{otherwise}\end{cases}$$
$$\begin{cases}0<d'_{y2}\leq 2^{\lambda-1}, & \text{if } \mathrm{LH}_\lambda(j,k)=-\mathrm{HH}_\lambda(j,k)\\ 2^{\lambda-1}<d'_{y2}\leq 2^\lambda, & \text{otherwise.}\end{cases} \tag{23}$$

Generally, we can estimate the coarse edge orientation in terms of the edge offsets $d_x$ and $d_y$. For example, we have $\theta<(\pi/4)$ in the cases of EMA-IB and EMA-IIB, and $\theta>(\pi/4)$ in the cases of EMA-IC and EMA-IIC. However, for the cases of EMA-IA and EMA-IIA with $2^{\lambda-1}<d_{x1},d_{y1}<2^\lambda$ and $2^{\lambda-1}<d'_{x2},d'_{y2}<2^\lambda$, the edge angle $\theta$ can be further estimated by

$$\begin{cases}0<\theta\leq\frac{\pi}{4} & \text{if } \mathrm{LH}_\lambda(j,k)\geq\mathrm{HL}_\lambda(j,k) \text{ and } I_1>I_2\\ & \text{or } \mathrm{LH}_\lambda(j,k)\leq\mathrm{HL}_\lambda(j,k) \text{ and } I_1<I_2\\ \frac{\pi}{4}<\theta<\frac{\pi}{2} & \text{if } \mathrm{LH}_\lambda(j,k)<\mathrm{HL}_\lambda(j,k) \text{ and } I_1>I_2\\ & \text{or } \mathrm{LH}_\lambda(j,k)>\mathrm{HL}_\lambda(j,k) \text{ and } I_1<I_2.\end{cases} \tag{24}$$

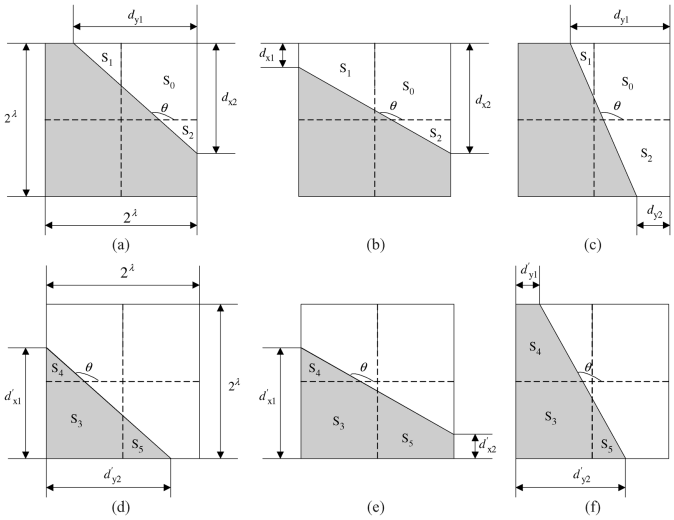


Fig. 4. Two edge models in the given block for the case of $\theta>(\pi/2)$. (a)–(c) First model with $S_0\leq 2^{2\lambda-2}$, namely EMP-IA, EMP-IB, and EMP-IC. (d)–(f) Second model with $S_3\leq 2^{2\lambda-2}$, namely EMP-IIA, EMP-IIB, and EMP-IIC.

Following the above analysis method, we use similar way to deal with the edge models in the case of $\theta>(\pi/2)$, which is depicted in Fig. 4. The calculated parameters are given by

$$\begin{aligned}S_0&=\frac{\mathrm{LH}_\lambda(j,k)-\mathrm{HL}_\lambda(j,k)}{2(I_1-I_2)}\\ S_1&=\frac{\mathrm{LH}_\lambda(j,k)+\mathrm{HH}_\lambda(j,k)}{2(I_1-I_2)}\\ S_2&=\frac{\mathrm{HH}_\lambda(j,k)-\mathrm{HL}_\lambda(j,k)}{2(I_1-I_2)},\\ S_3&=\frac{\mathrm{LH}_\lambda(j,k)-\mathrm{HL}_\lambda(j,k)}{2(I_1-I_2)}\\ S_4&=\frac{\mathrm{HL}_\lambda(j,k)+\mathrm{HH}_\lambda(j,k)}{2(I_2-I_1)}\\ S_5&=\frac{\mathrm{LH}_\lambda(j,k)-\mathrm{HH}_\lambda(j,k)}{2(I_1-I_2)}.\end{aligned} \tag{25}$$

According to (25), we express the classification criterion for the two edge models EMP-I and EMP-II as follows:

$$\mathrm{EMP}=\begin{cases}\mathrm{EMP-I}, & \text{if } \mathrm{HH}_\lambda(j,k)\geq 0 \text{ and } I_1<I_2\\ & \text{or } \mathrm{HH}_\lambda(j,k)\leq 0 \text{ and } I_1>I_2\\ \mathrm{EMP-II}, & \text{if } \mathrm{HH}_\lambda(j,k)<0 \text{ and } I_1<I_2\\ & \text{or } \mathrm{HH}_\lambda(j,k)>0 \text{ and } I_1>I_2\end{cases} \tag{26}$$

with

$$\begin{aligned}I_1>I_2, & \quad \text{if } \mathrm{LH}_\lambda(j,k)>0 \text{ and } \mathrm{HL}_\lambda(j,k)<0\\ I_1<I_2, & \quad \text{if } \mathrm{LH}_\lambda(j,k)<0 \text{ and } \mathrm{HL}_\lambda(j,k)>0.\end{aligned} \tag{27}$$

For the edge model belonging to EMP-I, further classification can be performed according to the following rules:

$$\mathrm{EMP-I}=\begin{cases}\mathrm{EMP-IA}, & \text{if } v1 \text{ or } v2 \text{ is satisfied}\\ \mathrm{EMP-IB}, & \text{if } v3 \text{ or } v4 \text{ is satisfied}\\ \mathrm{EMP-IC}, & \text{if } v5 \text{ or } v6 \text{ is satisfied}\end{cases} \tag{28}$$

where the detail conditions are defined as

$$v1=\begin{cases} Q1\le 0 \\ Q2\ge 0 \\ I_1>I_2 \end{cases},\quad v2=\begin{cases} Q1\ge 0 \\ Q2\le 0 \\ I_1<I_2 \end{cases},\quad v3=\begin{cases} Q1>0 \\ Q2\ge 0 \\ I_1>I_2 \end{cases}$$
$$v4=\begin{cases} Q1<0 \\ Q2\le 0 \\ I_1<I_2 \end{cases},\quad v5=\begin{cases} Q1\le 0 \\ Q2<0 \\ I_1>I_2 \end{cases},\quad v6=\begin{cases} Q1\ge 0 \\ Q2>0 \\ I_1<I_2 \end{cases} \tag{29}$$

with

$$\begin{aligned} Q1 &= \mathrm{LH}_\lambda(j,k)+\mathrm{HL}_\lambda(j,k)+\mathrm{HH}_\lambda(j,k) \\ Q2 &= \mathrm{LH}_\lambda(j,k)+\mathrm{HL}_\lambda(j,k)-\mathrm{HH}_\lambda(j,k). \end{aligned} \tag{30}$$

Similarly, we can obtain the classification approach for the edge model EMP-II, which has the same rules as (28) except that the $Q_1$ and $Q_2$ are replaced by

$$\begin{aligned} Q1 &= \mathrm{LH}_\lambda(j,k)+\mathrm{HL}_\lambda(j,k)-\mathrm{HH}_\lambda(j,k) \\ Q2 &= \mathrm{LH}_\lambda(j,k)+\mathrm{HL}_\lambda(j,k)+\mathrm{HH}_\lambda(j,k). \end{aligned} \tag{31}$$

As mentioned previously, there are many changes for every edge model with respect to different edge offsets. Based on the similar method in (23), we can get coarse offset information in the $2^\lambda \times 2^\lambda$ EMP block.

In addition, as shown in Fig. 5, there are two special cases of horizontal and vertical edges, which are denoted by EMS. The corresponding estimation criteria is employed in terms of difference properties of these coefficients

$$\theta=\begin{cases} 0 & \text{if } \mathrm{HL}_\lambda(j,k)=\mathrm{HH}_\lambda(j,k)=0 \text{ and } \mathrm{LH}_\lambda(j,k)\neq 0 \\ \frac{\pi}{2} & \text{if } \mathrm{LH}_\lambda(j,k)=\mathrm{HH}_\lambda(j,k)=0 \text{ and } \mathrm{HL}_\lambda(j,k)\neq 0. \end{cases} \tag{32}$$

It is noted that the edge offsets $d_{x1}$ and $d_{y1}$ cannot be estimated from the NHT coefficients at the $\lambda$th level due to the insufficient information provided by only one nonzero high frequency coefficient. However, we can incorporate the coefficients at the $(\lambda-1)$th level to locate the current block edge offsets. The general scheme can be formulated as (33)–(34), shown at the bottom of the page.

Finally, based on the above analysis, we summarize the proposed edge classification algorithm, as depicted in Fig. 6, in which there are a total of 42 edge models that can be determined from three NHT coefficients. Each step in the classification process, such as $\mathrm{EM}\rightarrow\mathrm{EMA}$, $\mathrm{EMA}\rightarrow\mathrm{EMA}-\mathrm{I}$, corresponds to a certain criterion defined in this paper. The detailed representation range of each model is described in Table I.

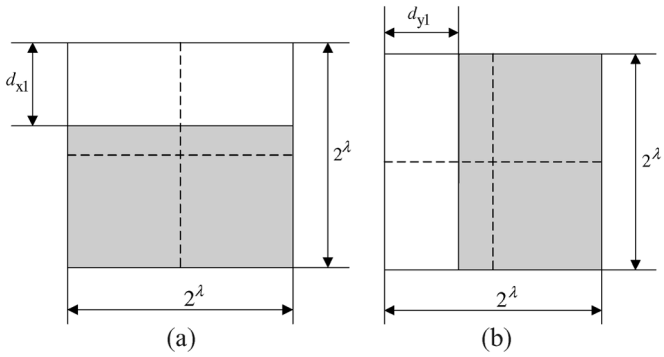


Fig. 5. Two special edge models (EMS) in the given block. (a) $\theta=0$. (b) $\theta=\pi/2$.

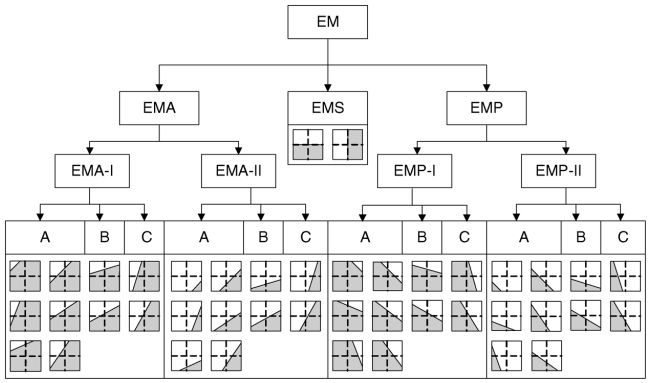


Fig. 6. Framework of the proposed edge classification method.

## III. Edge Classification Results

In this section, we evaluate the effectiveness of the proposed block edge classification method. For the evaluation, we use several still images and a standard video sequence. Firstly, we discuss the threshold value in the classification process.

### A. Thresholding Method

As mentioned in the previous section, we can identify the edge model for each image block by using the edge orienta-

$$\begin{cases} 0<d_{x1}<2^{\lambda-1}, & \text{if } \mathrm{LH}_{\lambda-1}(2j,2k)+\mathrm{LH}_{\lambda-1}(2j,2k+1) \\ & \quad >\mathrm{LH}_{\lambda-1}(2j+1,2k)+\mathrm{LH}_{\lambda-1}(2j+1,2k+1) \\ d_{x1}=2^{\lambda-1}, & \text{if } \mathrm{LH}_{\lambda-1}(2j,2k)+\mathrm{LH}_{\lambda-1}(2j,2k+1) \\ & \quad =\mathrm{LH}_{\lambda-1}(2j+1,2k)+\mathrm{LH}_{\lambda-1}(2j+1,2k+1) \\ 2^{\lambda-1}<d_{x1}<2^{\lambda}, & \text{otherwise} \end{cases} \tag{33}$$

$$\begin{cases} 0<d_{y1}<2^{\lambda-1}, & \text{if } \mathrm{HL}_{\lambda-1}(2j,2k)+\mathrm{HL}_{\lambda-1}(2j+1,2k) \\ & \quad >\mathrm{HL}_{\lambda-1}(2j,2k+1)+\mathrm{HL}_{\lambda-1}(2j+1,2k+1) \\ d_{y1}=2^{\lambda-1}, & \text{if } \mathrm{HL}_{\lambda-1}(2j,2k)+\mathrm{HL}_{\lambda-1}(2j+1,2k) \\ & \quad =\mathrm{HL}_{\lambda-1}(2j,2k+1)+\mathrm{HL}_{\lambda-1}(2j+1,2k+1) \\ 2^{\lambda-1}<d_{y1}<2^{\lambda}, & \text{otherwise} \end{cases} \tag{34}$$

TABLE I
REPRESENTATION RANGE OF EACH EDGE MODEL

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| EHM-IA$_{1,1}$ | EHM-IA$_{2,1}$ | EHM-IA$_{3,1}$ | EHM-IA$_{1,2}$ | EHM-IA$_{2,2}$ | EHM-IA$_{3,2}$ | EHM-IB$_1$ | EHM-IB$_2$ | EHM-IC$_1$ | EHM-IC$_2$ |
| $(0,\pi/2)$ | $(\pi/4,\pi/2)$ | $(0,\pi/4)$ | $\pi/4$ | $(0.15\pi,\pi/4)$ | $(\pi/4,0.35\pi)$ | $(0,0.15\pi)$ | $(0,\pi/4)$ | $(0.35\pi,\pi/2)$ | $(\pi/4,\pi/2)$ |
| EHM-IIA$_{1,1}$ | EHM-IIA$_{2,1}$ | EHM-IIA$_{3,1}$ | EHM-IIA$_{1,2}$ | EHM-IIA$_{2,2}$ | EHM-IIA$_{3,2}$ | EHM-IIB$_1$ | EHM-IIB$_2$ | EHM-IIC$_1$ | EHM-IIC$_2$ |
| $(0,\pi/2)$ | $(\pi/4,\pi/2)$ | $(0,\pi/4)$ | $\pi/4$ | $(0.15\pi,\pi/4)$ | $(\pi/4,0.35\pi)$ | $(0,0.15\pi)$ | $(0,\pi/4)$ | $(0.35\pi,\pi/2)$ | $(\pi/4,\pi/2)$ |
| EVM-IA$_{1,1}$ | EVM-IA$_{2,1}$ | EVM-IA$_{3,1}$ | EVM-IA$_{1,2}$ | EVM-IA$_{2,2}$ | EVM-IA$_{3,2}$ | EVM-IB$_1$ | EVM-IB$_2$ | EVM-IC$_1$ | EVM-IC$_2$ |
| $(\pi/2,\pi)$ | $(3\pi/4,\pi)$ | $(\pi/2,3\pi/4)$ | $3\pi/4$ | $(3\pi/4,0.85\pi)$ | $(0.65\pi,3\pi/4)$ | $(0.85\pi,\pi)$ | $(3\pi/4,\pi)$ | $(\pi/2,0.65\pi)$ | $(\pi/2,3\pi/4)$ |
| EVM-IIA$_{1,1}$ | EVM-IIA$_{2,1}$ | EVM-IIA$_{3,1}$ | EVM-IIA$_{1,2}$ | EVM-IIA$_{2,2}$ | EVM-IIA$_{3,2}$ | EVM-IIB$_1$ | EVM-IIB$_2$ | EVM-IIC$_1$ | EVM-IIC$_2$ |
| $(\pi/2,\pi)$ | $(3\pi/4,\pi)$ | $(\pi/2,3\pi/4)$ | $3\pi/4$ | $(3\pi/4,0.85\pi)$ | $(0.65\pi,3\pi/4)$ | $(0.85\pi,\pi)$ | $(3\pi/4,\pi)$ | $(\pi/2,0.65\pi)$ | $(\pi/2,3\pi/4)$ |

tion feature. However, those blocks with very small intensity changes should be classified as the homogeneous blocks because of the very small edge strength. Therefore, a certain threshold should be employed to measure the edge strength in the given block. It is known that no automatic parameter selection process exists, mainly due to the strong dependency of the optimal parameter set of an edge detector on the input image [30]. Most of the existing methods used a hard threshold to classify the coefficients. For example, the threshold between 15 and 20 is recommended by experimental observation in [18]. It is known that the sensitivity of human eyes to discriminate between difference in intensity depends not only on the difference itself but also on the intensity level. Since an image usually contains multiple regions, which may have great spatial variations, it is difficult to extract these edges based only on a hard threshold. Those weak block edges are easily missed by this method. Therefore, in order to identify the edges efficiently, we use the contrast information as the decision function to perform the thresholding procedure, which is defined as

$$F = f_{\mathrm{LH}} + f_{\mathrm{HL}} + f_{\mathrm{HH}} \tag{35}$$

with

$$\begin{aligned} f_{\mathrm{LH}} &= \left| \frac{\mathrm{LH}_\lambda(j,k)}{\mathrm{LH}_\lambda(j,k) + \mathrm{LL}_\lambda(j,k)} \right| \\ f_{\mathrm{HL}} &= \left| \frac{\mathrm{HL}_\lambda(j,k)}{\mathrm{HL}_\lambda(j,k) + \mathrm{LL}_\lambda(j,k)} \right| \\ f_{\mathrm{HH}} &= \left| \frac{\mathrm{HH}_\lambda(j,k)}{\mathrm{HH}_\lambda(j,k) + \mathrm{LL}_\lambda(j,k)} \right|. \end{aligned} \tag{36}$$

If we incorporate (8)–(11) into (36), we can see that each variable in (36) can be represented by the contrast-related value in a certain direction, i.e.,

$$\begin{cases} f_{\mathrm{LH}} = \frac{1}{2}\left|1 - \frac{B_2+B_3}{B_0+B_1}\right| \\ f_{\mathrm{HL}} = \frac{1}{2}\left|1 - \frac{B_1+B_3}{B_0+B_2}\right| \\ f_{\mathrm{HH}} = \frac{1}{2}\left|1 - \frac{B_1+B_2}{B_0+B_3}\right|. \end{cases} \tag{37}$$

It is well known that the ratio of a target to a background, called the Weber fraction, is nearly a constant value [25], [26]. Therefore, the block is classified as a homogeneous block when the decision function $F$ is smaller than a certain threshold $Th$. In addition, each NHT coefficient with the smaller value less than $0.6Th$ will also be taken as zero. Based on a lot of experiments, it is observed that the threshold within the interval of 0.05 to 0.1 can provide good classification results.

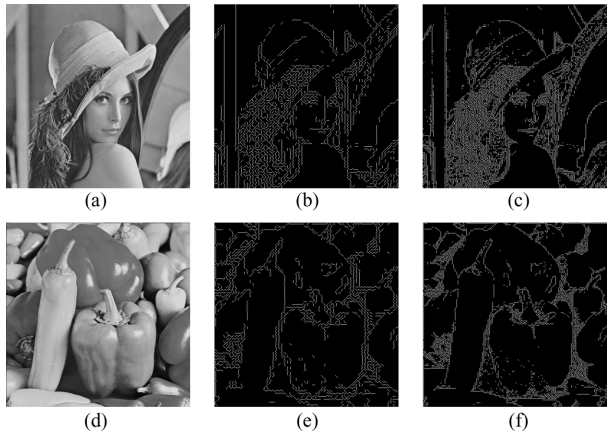

Fig. 7. Experimental results for (a) *Lena* and (d) *Peppers* images. (a) and (d) Original image. (b) and (e) Edge classification for 8 × 8 image block. (c) and (f) Edge classification for 4 × 4 image block.

TABLE II
COMPARISONS OF DIFFERENT EDGE CLASSIFICATION METHODS

| References | Domain | NoC | Edge Models | NoEM |
|---|---|---|---|---|
| [15] | DCT | 2 | one edge | 8 |
| [16] | DCT | 5 | one edge | 6 |
| [17] | DCT | 10 | one/two edges | 32 |
| [18] | DCT | 3 | one edge | 4 |
| [14] | Spacial | mean/variance | one edge | 4 |
| Proposed Method | NHT | 3 | one edge | 42 |

### B. Edge Classification Results

Fig. 7 shows the block edge classification results for *Lena* and *Peppers* images by using our proposed method. The step edge is shown for the block that satisfies the "edgy" criterion. We can see that the block edges are classified fairly well with respect to the original images. With the decrease in block size, more edge details can be extracted by our detection method.

We then compared our results with the existing methods described in [14]–[18]. Table II briefly describes the properties of these methods, where NoC and NoEM denote the number of coefficients used and the edge models, respectively. It is found that the previous four methods in this table concentrate on the classification in the DCT domain, while the method in [14] is designed for the pixel domain. Most existing methods assume that there is only one step edge in the current block except for the method [17]. For the number of edge models, our proposed method achieve 42 models while incurring less computational load.

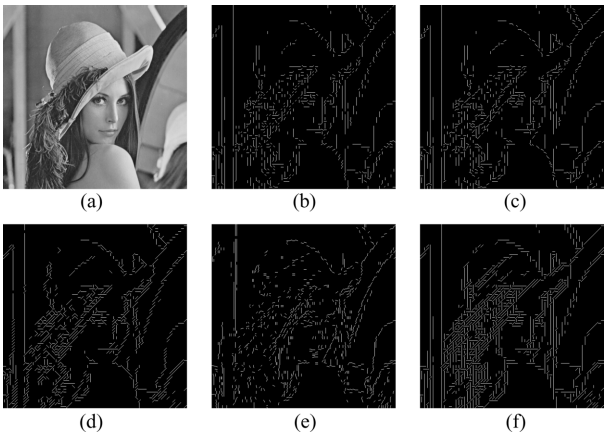

Fig. 8. Experimental results for *Lena* image by other methods. (a) Original image. (b) Algorithm in [15]. (c) Algorithm in [16]. (d) Algorithm in [18]. (e) Algorithm in [17]. (f) Algorithm in [14].

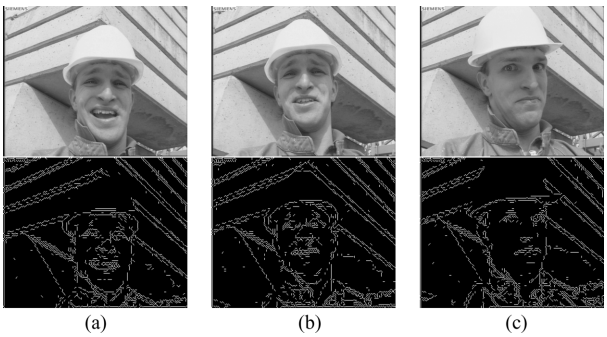

Fig. 9. Experimental results for *Foreman* sequence. (a) Result for frame 20. (b) Result for frame 60. (c) Result for frame 100.

The edge classification results obtained from these methods are shown in Fig. 8. All the experimental parameters in the implementation are selected based on the original algorithms. Since most of the methods are sensitive to the selected thresholds, we selected optimum threshold value by brute-force, i.e., testing several values and selecting the one which performs the best. From the obtained results, it is found that most of the distinct edges can be detected by choosing the suitable threshold. However, there are many edge blocks in the hair regions that cannot be identified using the existing methods. Certainly, we can decrease the thresholds to obtain more edge blocks, but a large amount of blocks in the face regions will be included as well. In addition, the edges obtained are not very accurate for the existing algorithms especially for [14] and [18] due to the use of only four edge models.

We also extended our proposed method to the standard video sequence. Fig. 9 shows the edge classification results for the *Foreman* sequence in CIF ($352 \times 288$) format. It is observed that the edge blocks have been identified accurately with respect to the original video images.

In order to provide objective classification performance, we used six images (see Fig. 10), which can provide ideal edge blocks. We first employed Canny operator to extract the edges from the given images. Then, we compared the block edge direction obtained by the existing methods with the result of the

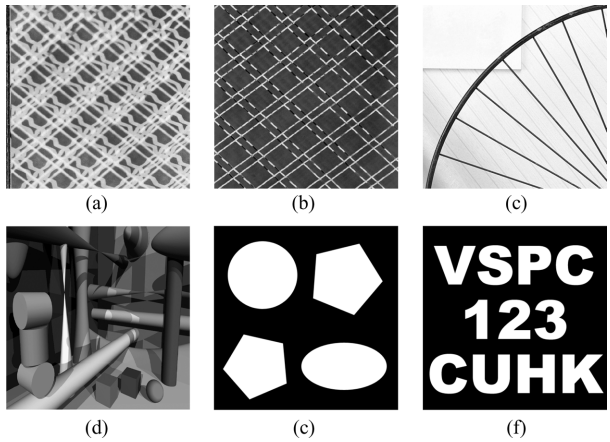


Fig. 10. Images (Im1 $\sim$ Im6 from (a)–(f), respectively) for the objective measurement.

Canny detector. The average deviation of the edge directions is used to perform the objective measurement, which is defined as

$$\frac{1}{K_b}\sum_{i=1}^{K_b}\left[\frac{1}{K_b^e}\sum_{j=1}^{K_b^e}\left\|\theta_j^i-\overline{\theta_i}\right\|\right] \tag{38}$$

where $K_b$ and $K_b^e$ denote the number of edge block and edge pixels in this block, respectively. $\theta_j^i$ represents the calculated edge direction at the edge pixel $j$ in the $i$th block. The classified edge direction of block $i$ is denoted by $\overline{\theta_i}$. The comparison results of the average deviation of the edge directions have been given in Table III. It can be clearly observed that the proposed algorithm yields good classification results compared with the existing methods.

Next, we will concentrate on the computational complexity of the proposed block edge classification method. In our method, the NHT at a certain level should be performed to obtain the NHT coefficients. However, it can be seen that only addition and subtraction computational loads are imposed on the decomposition process. After calculating the NHT coefficients, the simple comparison operations are employed to classify the block edge model. If we take out the qualification of the coefficients (i.e., only three multiplications), no multiplication operations are required in our method.

Table IV gives the comparisons of the computational complexity of different methods. It is found that only 72 additions are required to compute four NHT coefficients for a $8 \times 8$ block, whilst hundreds of multiplications and additions are needed to calculate those DCT coefficients. If we only consider the edge detection stage after the transform, algorithms [15] and [16] can quickly classify the block edge using three comparisons. Slightly less computations are used in our method with respect to the methods [17] and [18]. Since the mean and variance of the block must be calculated, higher computational operations are involved for the method [14]. It should be noted that since the existing methods [15]–[18] were devised specially for fast analysis of DCT-based compressed images or videos, more computational loads of the block edge classification can be saved by

TABLE III
OBJECTIVE MEASUREMENT OF DIFFERENT EDGE CLASSIFICATION METHODS

| References | [15] | [16] | [17] | [18] | [14] | Our method |
|---|---|---|---|---|---|---|
| Im1 | 0.7136 | 0.6332 | 0.4781 | 0.8319 | 0.7419 | 0.4162 |
| Im2 | 0.4988 | 0.4400 | 0.4918 | 0.5120 | 0.5241 | 0.3744 |
| Im3 | 0.6696 | 0.4865 | 0.4407 | 0.5379 | 0.6240 | 0.2665 |
| Im4 | 0.8847 | 0.8342 | 0.7237 | 0.9812 | 0.8346 | 0.5033 |
| Im5 | 0.4465 | 0.4533 | 0.3230 | 0.6402 | 0.7840 | 0.2183 |
| Im6 | 0.7119 | 0.6110 | 0.5686 | 0.6929 | 0.7191 | 0.4730 |

TABLE IV
COMPARISONS OF THE COMPUTATIONAL COMPLEXITY FOR DIFFERENT METHODS

| References | Multiplication | | Addition | | Comparison | Other operations | | |
|---|---|---|---|---|---|---|---|---|
| | $8 \times 8$ Transform | Other | $8 \times 8$ Transform | Other | | $\lvert \cdot \rvert$ | log | exp |
| [15] | $2 \times 64$ | 0 | $2 \times 63$ | 0 | 3 | 2 | 0 | 0 |
| [16] | $5 \times 64$ | 0 | $5 \times 63$ | 0 | 3 | 0 | 0 | 0 |
| [17] | $10 \times 64$ | 0 | $10 \times 63$ | 0 | $10 \sim 20$ | 10 | 0 | 0 |
| [18] | $3 \times 64$ | 7 | $3 \times 63$ | 6 | 6 | 5 | 0 | 0 |
| [14] | 0 | 70 | 0 | 191 | 4 | 1 | 1 | 1 |
| Proposed Method | 0 | 3 | 72 | 8 | 9 | 3 | 0 | 0 |

(a) (b)

Fig. 11. Texture block estimation for *Lena* image. (a) Original image. (b) Result for texture block classification for $16 \times 16$ block.

reading the DCT coefficients directly from the compressed domain instead of full decompression.

In addition, it should be mentioned that the proposed method is based on the assumption that there is a linear edge cutting through a block. It is reasonable to give this assumption when the block size is not very large, which means that the change of intensity can be modeled well by a linear or homogeneous form. Experimental results also indicate good performance of detecting block edge by using the proposed algorithm. However, when the block becomes larger, more and more textures including nonlinear and/or nonbinary edges might be observed. Thus, there may be a problem to simply apply the linear edge models to classify the block that may consist of many edges with different directions. In order to distinguish the texture block from other edge blocks, the variance of the block was employed in [14] to identify a textural block when its variance is larger than a preset threshold. Here, we use a simple method to separate the nonlinear edge blocks in terms of the obtained edge models. We decompose the current block at level $\lambda$ into four subblocks at level $\lambda - 1$. If different edge models such as EMA and EMP are observed in these subblocks, then we declare that the current block is a nonlinear or texture block. An illustration of the texture blocks detection is shown in Fig. 11. It can be found that most of $16 \times 16$ textural blocks in the hair regions are separated from the input image.

## IV. CODING APPLICATION

It is known that the new video coding standard H.264/AVC achieves the highest coding gain by allowing a rich set of coding modes. However, in order to select the best mode for a macroblock (MB), the current H.264/AVC reference encoder employs exhaustive search to calculate the rate-distortion cost of every mode, and the resulting complexity of the encoder is extremely high. In this section, we develop a fast mode decision algorithm in H.264/AVC intra-prediction based on our proposed edge classification method.

### A. Mode Decision for Intra-Prediction in H.264/AVC

In intra-coding of H.264/AVC, there are nine prediction modes for each $4 \times 4$ luma block, four optional modes for a $16 \times 16$ luma block and four modes for the chroma components. Fig. 12(a) shows the samples of a $4 \times 4$ luma block and the relevant neighboring pixels. The samples a-p will be predicted based on the samples A-M in terms of the optimal prediction modes. There are a total of eight prediction directions and one dc prediction mode for $4 \times 4$ luma block in H.264/AVC [27].

- Mode 0: Vertical.
- Mode 1: Horizontal.
- Mode 2: dc.
- Mode 3: Diagonal down-left.
- Mode 4: Diagonal down-right.
- Mode 5: Vertical-right.
- Mode 6: Horizontal-down.
- Mode 7: Vertical-left.
- Mode 8: Horizontal-up.

The arrows in Fig. 12(b)–(j) indicate the direction of prediction in each mode. For the dc mode corresponding to the Mode 2, all samples in $4 \times 4$ luma block are predicted by the mean of samples A–D and I–L. Modes 0 and 1 are employed to predict the pixels a-p from the vertical and horizontal directions, respectively. For Modes 3–8, the samples are calculated by a weighted average of the prediction samples A–M. For example, if we select Mode 3, the pixel "a" will be predicted by $(A+C+2B)/4$.

For $16 \times 16$ block with smoothly varying luminance, four prediction modes are used, namely

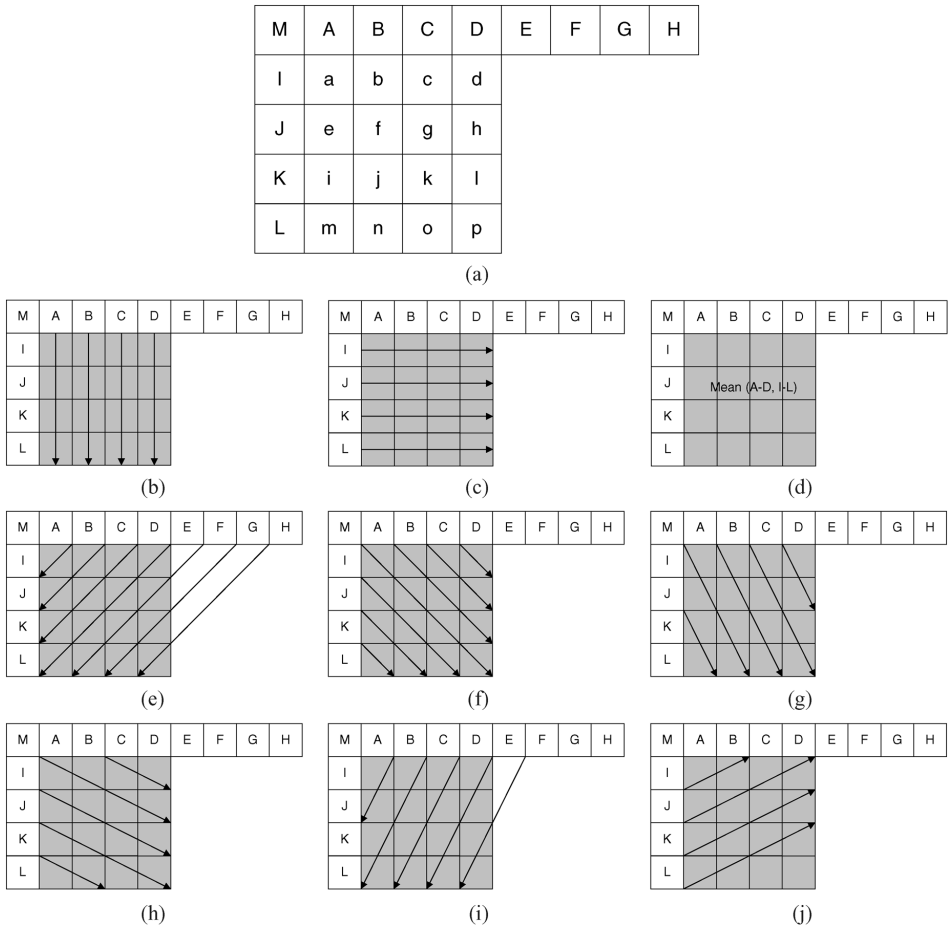


Fig. 12. (a) Prediction samples in 4 × 4 luma block. (b) Mode 0 (vertical).(c) Mode 1 (horizontal). (d) Mode 2 (DC). (e) Mode 3 (diagonal down-left). (f) Mode 4 (diagonal down-right). (g) Mode 5 (vertical-right). (h) Mode 6 (horizontal-down). (i) Mode 7 (vertical-left). (j) Mode 8 (horizontal-up).

- Mode 0: Vertical.
- Mode 1: Horizontal.
- Mode 2: dc.
- Mode 3: Plane

The plane prediction mode 3 uses a linear interpolation function between the upper and left-hand samples to predict the current samples. The 16 × 16 intra-prediction works well in homogeneous areas of an image. In addition, H.264/AVC also supports four prediction modes for each 8 × 8 chroma component of an intra-coded a MB. These modes including dc, horizontal, vertical and plane modes are very similar to the 16 × 16 luma prediction modes, which are predicted from previously encoded chroma samples.

### *B. Fast Mode Decision Based on the Edge Classification Algorithm*

Here, we employ the proposed fast edge classification algorithm to perform the mode decision for intra-prediction in H.264/AVC. The edge detection result is selected as the candidate of the prediction mode for further rate-distortion optimization (RDO) computation.

The organized procedure of the fast mode decision consists of three steps. The first step is the classification of block edge based on three NHT coefficients (LH, HL, and HH) for a given 4 × 4 block. Then the corresponding modes are selected as the candidate modes according to the edge feature. The mode which has minimum cost will be employed as the final mode for this block. The second step is to check if the current MB is homogenous according to (35). If the MB is not homogenous, the intra-16 × 16 modes will be skipped, otherwise, intra-16 × 16 modes should be checked. The third step is to compare the cost of intra-4 × 4 mode and intra-16 × 16 mode and select the one with lower cost as the final coding mode.

In our work, we use different methods to handle the different intra-blocks. For the 4 × 4 luma block, the edge classification algorithm will be used to determine the intra-prediction modes. Based on the edge mode, the corresponding prediction modes can be selected in terms of the possible edge directions. It should be noted that, the derivation process of the edge classification is based on the existing condition in the given block, which means that there are less than 42 edge modes for the 4 × 4 discrete block. Table V gives the relationship of the prediction modes and the edge models. The subscripts in this table denote the model positions in Fig. 6. For example, $\mathrm{EMA}-\mathrm{IA}_{x,2}$ refers to the model at the second column for the case of EMA-IA shown in Fig. 6. In addition, if the block is classified as a homogeneous block, only Mode 2 (dc) is selected as the candidate prediction mode.

As an alternative to the 4 × 4 luma modes, it is known that four 16 × 16 luma prediction modes work very well in smooth

TABLE V
RELATIONSHIP BETWEEN THE PREDICTION MODES AND THE EDGE MODELS FOR $4 \times 4$ LUMA BLOCK

| **Prediction Modes** | 0 | 1 | 8, 3, 7 | 5, 4, 6 | 0, 5, 4 | 4, 6, 1 | 3, 7, 0 | 1, 8, 3 |
|---|---|---|---|---|---|---|---|---|
| **Edge Model** | EMS-2 | EMS-1 | EMA-IA$_{1,1}$<br>EMA-IA$_{x,2}$<br>EMA-IIA$_{1,1}$<br>EMA-IIA$_{x,2}$ | EMP-IA$_{1,1}$<br>EMP-IA$_{x,2}$<br>EMP-IIA$_{1,1}$<br>EMP-IIA$_{x,2}$ | EMP-IA$_{3,1}$<br>EMP-IC<br>EMP-IIA$_{3,1}$<br>EMP-IIC | EMP-IA$_{2,1}$<br>EMP-IB<br>EMP-IIA$_{2,1}$<br>EMP-IIB | EMA-IA$_{2,1}$<br>EMA-IC<br>EMA-IIA$_{2,1}$<br>EMA-IIC | EMA-IA$_{3,1}$<br>EMA-IB<br>EMA-IIA$_{3,1}$<br>EMA-IIB |

TABLE VI
PERFORMANCE OF THE PROPOSED ALGORITHM COMPARED WITH JM10.1 AND ALGORITHM [28] FOR ALL INTRA-FRAMES

| **Format** | **Sequence** | Algorithm in [28] | | | Our Algorithm | | |
|---|---|---|---|---|---|---|---|
| | | **△ PSNR** (dB) | **△ Bitrate** (%) | **△ Time** (%) | **△ PSNR** (dB) | **△ Bitrate** (%) | **△ Time** (%) |
| QCIF | Foreman | -0.095 | 1.345 | -52.053 | -0.116 | 1.639 | -77.787 |
| | Container | -0.208 | 2.595 | -50.642 | -0.209 | 2.610 | -77.823 |
| | Carphone | -0.140 | 1.826 | -49.144 | -0.186 | 2.409 | -77.576 |
| | MissA | -0.279 | 4.909 | -44.606 | -0.156 | 2.740 | -76.324 |
| CIF | Bus | -0.259 | 2.865 | -58.530 | -0.226 | 2.501 | -81.295 |
| | Paris | -0.201 | 2.096 | -60.781 | -0.217 | 2.270 | -81.485 |
| | Hallmonitor | -0.150 | 2.693 | -51.364 | -0.081 | 1.459 | -77.990 |
| | Mobile | -0.256 | 2.227 | -54.206 | -0.247 | 2.108 | -78.838 |
| | Motherdaughter | -0.251 | 4.282 | -52.368 | -0.213 | 3.640 | -78.299 |
| | News | -0.260 | 3.432 | -53.010 | -0.231 | 3.027 | -77.777 |
| | Football | -0.176 | 2.393 | -51.222 | -0.178 | 2.425 | -77.687 |
| | Coastguard | -0.136 | 1.802 | -54.357 | -0.155 | 2.035 | -79.494 |
| | Tempete | -0.215 | 2.261 | -59.780 | -0.198 | 2.087 | -80.332 |
| Average | | -0.202 | 2.671 | -53.236 | -0.186 | 2.381 | -78.670 |

areas. Therefore, for the $16 \times 16$ intra-prediction block, we first use the decision function (35) to perform the homogeneous detection. If the current MB is detected as a homogeneous block by a large threshold 0.1, we will simply use the four intra-prediction modes, i.e., dc, horizontal, vertical, and plane, to give the optimal mode determination in terms of the RDO computation. If the MB is identified as edge block with horizontal or vertical direction, the $16 \times 16$ intra-prediction is also applied. Otherwise, the $16 \times 16$ MB is an EMA or EMP edge block; further processing will be skipped and the subsequent $4 \times 4$ mode decision is applied. Similarly, for intra-chroma blocks, Mode 3 will be considered if these blocks cannot be classified as horizontal or vertical edge models.

In addition, in the H.264 RDO algorithm, chroma component and luma component are combined together to perform the RDO computation. The total number of the RDO operations is the product of the two components. For a MB, a total of $4 \times (16 \times 9 + 4) = 592$ RDO operations have to be performed. Since the chroma and luma components are independent, we perform the RDO computation for the chroma and luma components, respectively.

### C. Simulation Results

In this section, the proposed fast mode decision algorithm is integrated with the JVT reference software JM10.1. The test is based on the Baseline Profile with RD optimization enabled and CAVLC entropy coding. The simulation was performed on a PC with Intel P-IV 2.8 GHZ CPU, 1 Gb RAM.

In this experiment, there are 150 frames for each sequence with quantization parameters 20, 24, 28, and 32. We encoded each sequence in all I-frames to observe the intra-prediction performance. In addition, for the $4 \times 4$ luma block, predicted modes and the dc mode have been selected as the RDO calculation. Similar idea can be found in [28]. The coding performance is measured and tabulated in terms of the difference in computation time, the change of average bit rate, and the change of average PSNR of luma and chroma components as defined in [28]. The differences of PSNR and bit rate are calculated by using the RD-curves fitting method in [29].

The last column of Table VI shows the performance comparison of the proposed algorithm with the reference software JM10.1 for 12 standard video sequences. Four QCIF and eight CIF sequences were used in our experiment, which contain the simple texture, such as *MissA*, *Motherdaughter*, and *News*, and the complex texture, such as *Bus*, *Mobile*, and *Football*. From the obtained results, we can see that the proposed algorithm can save an average 78.67% of the total coding time compared to JM10.1. For the *Bus* and *Paris* sequences, over 81% computational time can be reduced. The degradation of average PSNR and the equivalent increase in bit rate are only 0.186 dB and 2.381%, respectively. For example, only 0.081 dB loss in PSNR and 1.459% increase in bit rate were achieved for the *Hallmonitor* sequence, which indicates that there is little loss in visual quality as compared to the original. Fig. 13(a) and (b) depict the RD performances of the sequences *Foreman* and *Bus*, which show that the fast intra-prediction algorithm achieves the similar RDO performance as that of JM10.1.

To further verify the performance of our algorithm, we compared these results with those obtained by the method in [28] as shown in the third column of Table VI. It is shown that the proposed algorithm achieved about 25% reduction in encoding time with respect to the method in [28]. In addition, compared with [28], less PSNR loss (0.116 dB on average) and less bit rate increase (0.3% on average) were achieved by our method for the test video sequences. For example, a PSNR gain of 0.123 dB and

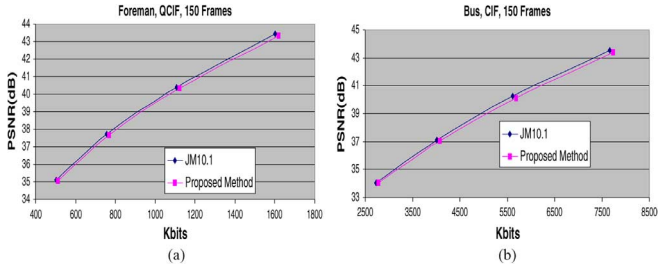


Fig. 13. (a) Foreman, $\triangle\mathrm{PSNR} = -0.116$, $\triangle\mathrm{Bits} = 1.639\%$. (b) Bus, $\triangle\mathrm{PSNR} = -0.226$, $\triangle\mathrm{Bits} = 2.501\%$.

TABLE VII
RATIO OF THE ESTIMATED EDGE MODEL COMPARED WITH THE ACTUAL MODEL IN JM10.1

| Format | Sequence | Ratio compared with JM (%) |
|---|---|---|
| QCIF | Foreman | 65.39 |
| | Container | 76.97 |
| | Carphone | 64.84 |
| | MissA | 81.82 |
| CIF | Paris | 65.86 |
| | Hallmonitor | 82.70 |
| | Mobile | 61.47 |
| | Container | 75.21 |
| | News | 78.74 |
| Average | | 72.5556 |

a bit rate decrease of 2.169% were obtained with respect to the method in [28] for the *MissA* sequence. From the experimental results, it is evidenced that our proposed algorithm outperforms that in [28] in terms of reduction in encoding time for all sequences tested without sacrificing image quality and bit rate.

In addition, we also calculated the number of the estimated edge models which are selected as the actual model with the lowest coding cost in JM10.1. The percentage of the predicted edge mode tallying with the actual mode is given in Table VII. We can see that good performance of the classification method can be achieved with the average matching ratio of 72.6 percent for nine video sequences.

## V. CONCLUSION

In this paper, a fast and efficient method was proposed to classify the block edges in an image. The basic idea is motivated by the observation that the edge in a block can be approximately represented by an ideal step edge. The method exploits the difference properties of the NHT coefficients effectively. With the proposed method, only the additions and subtractions are required when identifying the block edge, which means that the number of computational operations is significantly reduced. Experimental results were obtained by applying the proposed method to a large number of images and video sequences. It is shown that our method is capable of classifying the block edge quickly and efficiently.

The use of the proposed edge classification technique in H.264/AVC video coding standard was then presented. We proposed a fast intra-mode decision algorithm that uses the edge classification results to reduce the complexity and computation load in H.264/AVC. Experimental results show that the proposed algorithm increases the speed of intra-prediction mode selection significantly with negligible loss of coding quality and increase in bit rate.

## APPENDIX

Firstly, we present the derivation process of (17). As shown in Fig. 3, we can see that the difference between the EMA-I and EMA-II is the position of the cutting edge, where the edge is to the left or to the right of the block center for the two models.

Based on the simple geometrical knowledge, we have

$$\begin{cases} S_0 - S_1 - S_2 > 0, & \text{if the edge is to the left} \\ S_0 - S_1 - S_2 = 0, & \text{if the edge is at the center} \\ S_3 - S_4 - S_5 > 0, & \text{if the edge is to the right.} \end{cases} \tag{39}$$

From (14) and (16), we obtain

$$\begin{cases} \frac{\mathrm{HH}_\lambda(j,k)}{I_1 - I_2} > 0, & \text{if the edge is to the left} \\ \frac{\mathrm{HH}_\lambda(j,k)}{I_1 - I_2} = 0, & \text{if the edge is at the center} \\ \frac{\mathrm{HH}_\lambda(j,k)}{I_1 - I_2} < 0, & \text{if the edge is to the right.} \end{cases} \tag{40}$$

which corresponds to the original (17).

Secondly, (19) can also be derived by a simple geometrical relationship from Fig. 3(a)–(c), namely

$$\begin{cases} \frac{S_1}{S_0+S_1+S_2} \le \frac{1}{4} \\ \frac{S_2}{S_0+S_1+S_2} \le \frac{1}{4} \end{cases} \text{for EMA} - \text{IA}$$
$$\begin{cases} \frac{S_1}{S_0+S_1+S_2} > \frac{1}{4} \\ \frac{S_2}{S_0+S_1+S_2} \le \frac{1}{4} \end{cases} \text{for EMA} - \text{IB}$$
$$\begin{cases} \frac{S_1}{S_0+S_1+S_2} \le \frac{1}{4} \\ \frac{S_2}{S_0+S_1+S_2} > \frac{1}{4}. \end{cases} \text{for EMA} - \text{IC} \tag{41}$$

From (14), we can easily obtain the required conditions of (19). For example, for EMA-IA, we have

$$\begin{cases} \frac{\mathrm{HL}_\lambda(j,k) - \mathrm{LH}_\lambda(j,k) + \mathrm{HH}_\lambda(j,k)}{I_1 - I_2} > 0 \\ \frac{\mathrm{LH}_\lambda(j,k) - \mathrm{HL}_\lambda(j,k) + \mathrm{HH}_\lambda(j,k)}{I_1 - I_2} > 0. \end{cases} \tag{42}$$

The same derivation can also be used for (28).

## REFERENCES

[1] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video objects plane generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 525–538, Sep. 1998.
[2] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1190–1203, Dec. 1999.
[3] A. Khotanzad and J.-Y. Chen, "Unsupervised segmentation of textured images by edge detection in multidimensional feature," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 4, pp. 414–421, Apr. 1989.
[4] C.-C. Chu and J. K. Aggarwal, "The integration of image segmentation maps using region and edge information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1241–1252, Dec. 1993.
[5] M. A. Wani and B. G. Batchelor, "Edge-region-based segmentation of range images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 3, pp. 314–319, Mar. 1994.
[6] N. Ahuja, "A transform for multiscale image segmentation by integrated edge and region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 12, pp. 1211–1235, Dec. 1996.

[7] G. Iannizzotto and L. Vita, "Fast and accurate edge-based segmentation with no contour smoothing in 2-D real images," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1232–1237, Jul. 2000.
[8] W. Y. Ma and B. S. Manjunath, "EdgeFlow: A technique for boundary detection and image segmentation," *IEEE Trans. Image Process.*, vol. 9, pp. 1375–1388, Aug. 2000.
[9] T. Gevers, "Adaptive image segmentation by combining photometric invariant region and edge information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 848–852, Jun. 2002.
[10] P. Smith, T. Drummond, and R. Cipolla, "Layered motion segmentation and depth ordering by tracking edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 479–494, Apr. 2004.
[11] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
[12] I. Popovici and W. D. Withers, "Custom-built moments to edge location," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 637–642, Apr. 2006.
[13] C. S. Won, "A block-based MAP segmentation for image compressions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 592–601, Sep. 1998.
[14] C. S. Won, "Variable block size segmentation for image compression using stochastic models," in *Proc. IEEE ICIP*, 1996, vol. III, pp. 975–978.
[15] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," in *Proc. IS&T SPIE: Storage and Retrieval for Image and Video Databases IV*, Mar. 1996, vol. 2670, pp. 404–414.
[16] S.-W. Lee, Y.-M. Kim, and S. W. Choi, "Fast scene change detection using direct feature extraction from MPEG compressed videos," *IEEE Trans. Multimedia*, vol. 2, no. 4, pp. 240–254, Dec. 2000.
[17] H. Li, G. Liu, and Y. Li, "An effective approach to edge classification from DCT domain," in *Proc. IEEE ICIP*, Sep. 2002, vol. 1, pp. 940–943.
[18] H. S. Chang and K. Kang, "A compressed domain scheme for classifying block edge patterns," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 145–151, Feb. 2005.
[19] H. Li, G. Liu, and Z. Zhang, "Optimization of integer wavelet transforms based on difference correlation structures," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1831–1847, Nov. 2005.
[20] E. H. Adelson, E. Simonelli, and R. Hingorani, "Orthogonal pyramid transforms for image coding," in *Proc. SPIE, Visual Commun. Image Process. II*, Cambridge, MA, Oct. 1987, vol. 845, pp. 50–58.
[21] B. J. Falkowski and S. Rahardja, "WaIsh-like functions and their relations," *Proc. IEE Vis. Image Signal Process.*, vol. 143, no. 5, pp. 279–284, Oct. 1996.
[22] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[23] Y. Tadokoro and T. Higuchi, "Discrete Fourier transform computation via the Walsh transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 3, pp. 236–240, Jun. 1978.
[24] J. L. Shanks, "Computation of the fast Walsh–Fourier transform," *IEEE Trans. Computers*, vol. 18, no. 5, pp. 457–459, May 1969.
[25] C. H. Chou, "Adaptive transform coding of images based on removing just noticeable distortion," in *Proc. SPIE Visual Commun. Image Process.*, May 1995, vol. 2501, pp. 607–618.
[26] R. Schafer, "Design of adaptive and nonadaptive quantizers using subjective criteria," *Signal Process.*, vol. 5, pp. 333–345, Jul. 1983.
[27] *Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding*, ISO/IEC 14496-10:2003, Dec. 2003.
[28] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, Jul. 2005.
[29] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," presented at the 13th VCEG-M33 Meeting, Austin, TX, Apr. 2–4, 2001.
[30] Y. Yitzhaky and E. Peli, "A method for objective edge detection evaluation and detector parameter selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1027–1033, Aug. 2003.

**Hongliang Li** (M'06) received the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2005.

He is currently a Postdoctoral Fellow in the Visual Signal Processing and Communication Laboratory, the Chinese University of Hong Kong, Hong Kong SAR. His research interests include image and video coding, image segmentation, multimedia communication system, and nonstationary signal analysis.

**King Ngi Ngan** (M'79–SM'91–F'00) received the Ph.D. degree in electrical engineering from the Loughborough University, Loughborough, U.K.

He is currently a Chair Professor, Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong SAR, and was previously a Full Professor at the Nanyang Technological University, Singapore, and the University of Western Australia, Australia. He has published extensively including 3 authored books, 5 edited volumes, and over 200 refereed technical papers in the areas of image/video coding and communications.

Prof. Ngan is an Associate Editor of the *Journal on Visual Communications and Image Representation,* as well as an Area Editor of EURASIP *Journal of Signal Processing: Image Communication*, and served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and *Journal of Applied Signal Processing*. He chaired a number of prestigious international conferences on video signal processing and communications and served on the advisory and technical committees of numerous professional organizations. He will co-chair the IEEE International Conference on Image Processing (ICIP) to be held in Hong Kong in 2010. He is a Fellow of IET (U.K.), a Fellow of IEAust (Australia), and serves as an IEEE Distinguished Lecturer in 2006–2007.

**Zhenyu Wei** (S'06) received the B.E. and M.E. degrees from Tsinghua University, Beijing, China, in 2002 and 2005, respectively. He is currently pursuing the Ph.D. degree at the Department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong SAR.

His current research interests are image and video processing, perceptual distortion metric, and embedded system optimization and application.