H.264/Advanced Video Control Perceptual Optimization Coding Based on JND-Directed Coefficient Suppression

Zhengyi Luo, Li Song, *Member, IEEE,* Shibao Zheng, *Member, IEEE,* and Nam Ling, *Fellow, IEEE*

***Abstract*—The field of video coding has been exploring the compact representation of video data, where perceptual redundancies in addition to signal redundancies are removed for higher compression. Many research efforts have been dedicated to modeling the human visual system's characteristics. The resulting models have been integrated into video coding frameworks in different ways. Among them, coding enhancements with the just noticeable distortion (JND) model have drawn much attention in recent years due to its significant gains. A common application of the JND model is the adjustment of quantization by a multiplying factor corresponding to the JND threshold. In this paper, we propose an alternative perceptual video coding method to improve upon the current H.264/advanced video control (AVC) framework based on an independent JND-directed suppression tool. This new tool is capable of finely tuning the quantization using a JND-normalized error model. To make full use of this new rate distortion adjustment component the Lagrange multiplier for rate distortion optimization is derived in terms of the equivalent distortion. Because the H.264/AVC integer discrete cosine transform (DCT) is different from classic DCT, on which state-of-the-art JND models are computed, we analytically derive a JND mapping formula between the integer DCT domain and the classic DCT domain which permits us to reuse the JND models in a more natural way. In addition, the JND threshold can be refined by adopting a saliency algorithm in the coding framework and we reduce the complexity of the JND computation by reusing the motion estimation of the encoder. Another benefit of the proposed scheme is that it remains fully compliant with the existing H.264/AVC standard. Subjective experimental results show that significant bit saving can be obtained using our method while maintaining a similar visual quality to the traditional H.264/AVC coded video.**

***Index Terms*—Coefficient suppression, H.264/AVC, perceptual, video coding.**

I. Introduction

With the development of multimedia technologies, video applications have gained increasing popularity over the last two decades. To meet the increasingly large size of video data like high definition and high quality contents, continued efforts have been made to improve the compression performance of the H.264/advanced video control (AVC) video coding standard [1]. Recently, efforts have been made by the so-called Joint Collaborative Team on Video Coding (JCT-VC) to develop a high efficiency video coding (HEVC) standard [2]. Like previous video coding standards, video compression methods in H.264/AVC and the latest HEVC mainly focus on exploiting statistical correlation of signal and develop advanced tools to remove spatial, temporal, and symbol redundancy. However, such pure signal processing methods are probably hard to produce higher compression efficiencies as many of them are approaching the ceiling of performance. Since human eyes usually act as the ultimate receivers, many efforts have been dedicated to designing human visual system (HVS) friendly coding approaches to further remove perceptual redundancies since Mannos's benchmark paper in 1974 [3]. A typical example is the widely used visual frequency weighting matrix (also called scaling matrix or quantization matrix) between transform and quantization in many image and video coding standards, such as JPEG [4], H.264/AVC, and HEVC. Recent research shows that further gains can be expected along this direction by integrating advanced HVS models into latest video coding frameworks [5]–[21].

One way of implementing perceptual coding is to perform HVS guided preprocessing [5]–[7], which filters out inconspicuous information from the original video for higher compressibility. But it is difficult for preprocessing to control filtering parameters in a rate distortion optimization (RDO) way, as later the encoder works independently. More commonly, HVS characteristics work at the quantization stage of the encoder [8]–[13], where perceptual unimportant regions are coarsely quantized. In this way, fewer bits are allocated to regions that can withstand greater distortion; as a result the coding bitrate is reduced. Nevertheless, quantization parameters normally can only be adjusted at the macroblock level instead of the individual coefficient level.

Recently, increasing attention has also been paid to perceptual coding based on residual processing [14], which enables much finer adjustment of image quality at the pixel or frequency component level. As far as H.264/AVC coding is concerned, several methods of this class have also been proposed. Cheng *et al.* [15] introduced reduced resolution

Manuscript received February 24, 2012; revised June 8, 2012; accepted September 27, 2012. Date of publication January 25, 2013; date of current version May 31, 2013. This work was supported in part by the National 863 Program, under Grant 2012AA011703, the China MIIT Program, under Grant 2010ZX03004–003, the NSF, under Grant 60902020, the STCSM, under Grant 12DZ2272600, and the 111 project. This paper was recommended by Associate Editor W. Zeng.

Z. Luo, L. Song and S. Zheng are with the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: llzzyynjupt@sjtu.edu.cn; song_li@sjtu.edu.cn; sbzh@sjtu.edu.cn).

N. Ling is with the Department of Computer Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: nling@scu.edu).

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.2013.2240919

coding modes to the coding framework, which improved compression efficiency by downsampling the prediction residuals of some macroblocks while maintaining good subjective video quality. Schuur *et al.* [16] proposed to remove part of transform coefficients of prediction residuals alternately, so that acceptable subjective quality could be kept with fewer bits. However, they both did not take account of the images' specific visual characteristics. Mak *et al.* [17] proposed to discard some H.264/AVC transform coefficients using just noticeable distortion (JND) of the classic DCT domain. However, they neglected the specific differences between the H.264/AVC transform and the classic DCT transform. Chen *et al.* [18] chose to discard some prediction residuals in the pixel domain, but the effects of transform and quantization were not taken into account. Recently, Naccari *et al.* [19] proposed to perceptually modify the multiplication factors and prescaling factors in the forward and inverse quantization processes of H.264/AVC, but they did not produce H.264/AVC compliant bitstreams. Later they proposed a similar technique to the HEVC framework [20]. Besides, Wang *et al.* [21] proposed to normalize the transform coefficients before quantization, but they also did not produce standard compliant bitstreams.

In this paper, a novel method of perceptual coding for the H.264/AVC standard is presented based on HVS guided residual adjustment. The main contributions of this paper are as follows. First, a key rate distortion tool is integrated into the existing encoding framework to suppress the transform coefficients of prediction residuals in a frame-adaptive and sensitivity-normalized manner. Second, the Lagrange multiplier for rate distortion optimization corresponding to the new tool is analytically derived in terms of the equivalent distortion. Third, we derive a JND mapping formula between the integer DCT domain and the classic DCT domain which permits us reuse of the JND models in a more natural way. Moreover, the JND threshold can be refined by adopting a saliency algorithm in the coding framework and we reduce the complexity of the JND computation by reusing the motion estimation of the encoder. It should be noted that the proposed method is fully compliant with the H.264/AVC standard. Experimental results show that significant bit saving can be obtained by our method at a similar visual quality to the traditional H.264/AVC coded video.

The remainder of this paper is organized as follows. Section II describes the coding framework with the new adaptive coefficient suppression tool and the analytical derivation of the corresponding Lagrange multiplier. Section III estimates the JND related perceptual parameters, including JND computation from the classic DCT domain to the integer DCT domain and JND threshold adjustment based on visual saliency. Section IV discusses the implementation issues like complexity issues and motion estimation reuse. Experimental results validating the effectiveness of our method are shown in Section V. Section VI draws the conclusion.

II. Coding with Adaptive Coefficient Suppression

Just noticeable distortion, which refers to the minimum distortion that can be perceived by HVS with respect to the original video, provides excellent cues of visual sensitivity. As reviewed above, JND thresholds have been widely used in perceptual coding [8], [19], and JND can also be used in subthreshold coding by removing the prediction residuals below the JND thresholds [17], [18]. When quantization is involved, residuals may be further suppressed as long as the total distortion is confined into a certain range of JND profile. To this end, we propose to take the quantization errors into account and conduct JND-directed coefficient suppression after quantization.

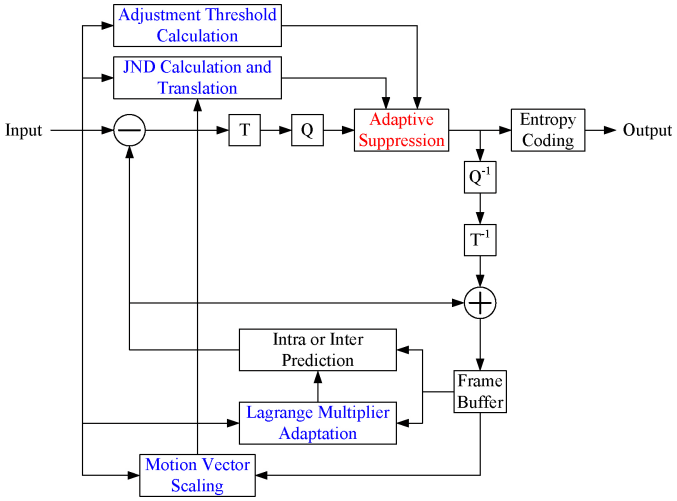


Fig. 1. Block diagram of perceptual coding with adaptive coefficient suppression.

The block diagram incorporating our method of perceptual coding is shown in Fig. 1. The paramount addition of the proposed scheme to H.264/AVC is that after quantization, coefficients of residuals are further adjusted by JND-directed *adaptive suppression*. The proposed scheme can be completely compatible with the current H.264/AVC standard, since it is unnecessary to transmit extra side information and change the coding syntax. The JND calculation and translation component is to estimate the distribution of visual sensitivity in the H.264/AVC transform domain and the adjustment threshold calculation component is to calculate the updated JND thresholds via a visual saliency map. These two components will be discussed in Section III. Here we assume both JND thresholds and adjustment thresholds are available. We first address the design of a perceptually friendly metric for coefficients suppression after quantization, and then present an analytical solution to adapt the Lagrange multiplier for RDO when the adaptive suppression tool is involved.

A. Adaptive Coefficient Suppression

Usually the bitrate of H.264/AVC coding depends highly on the number and the levels of nonzero coefficients after quantization. H.264/AVC assumes scalar quantization, which for the (i, j)th subband of the nth block can be expressed theoretically as [22]

$$l_{n,i,j} = \text{round}\left(w_{n,i,j} \cdot PF_{i,j} / Q_{\text{step}}\right) \tag{1}$$

where $w_{n,i,j}$ and $l_{n,i,j}$ denote, respectively, the transform coefficient before and after quantization, $PF_{i,j}$ is the postscaling factor, and Q_{step} is the quantization step size. To facilitate the arithmetic operations, normally the quantization process can be implemented as [22]–[24]

$$\left|l_{n,i,j}\right| = \left(\left|w_{n,i,j}\right| \cdot MF_{i,j} + f\right) >> qbits \tag{2}$$

$$\text{sign}\left(l_{n,i,j}\right) = \text{sign}\left(w_{n,i,j}\right) \tag{3}$$

where f is the offset, and $MF_{i,j}$ and $qbits$ are the precomputed multiplication factor and the number of right shift, respectively.

For a nonzero coefficient $l_{n,i,j}$, if we suppress it by k, as uniform reconstruction with no offsets are adopted in H.264/AVC [25], approximately the introduced error will be

$$e_{n,i,j}(k) \approx \left|w_{n,i,j}\right| - \left[\left(\left|l_{n,i,j}\right| - k\right) << qbits\right] / MF_{i,j}. \tag{4}$$

As far as visual effects are concerned, it is found that in the (i, j)th subband the perceptual distortion can be modeled by a function of the JND-normalized error [26]

$$\tau_{n,i,j}(k) = e_{n,i,j}(k) / J^*_{n,i,j} \tag{5}$$

where $J^*_{n,i,j}$ is the corresponding JND threshold[1] in this subband. To be in accordance with visual sensitivity, a JND-normalized adjustment threshold $T_{n,i,j}$ may be set for images. With the sensitivity distribution, coefficients of each frequency component can be suppressed in a sensitivity-normalized manner. Specifically, in the (i, j)th subband of the nth block the coefficient level after suppression $l'_{n,i,j}$ is

$$\left|l'_{n,i,j}\right| = \left|l_{n,i,j}\right| - k_{n,i,j} \tag{6}$$

$$\text{sign}\left(l'_{n,i,j}\right) = \text{sign}\left(l_{n,i,j}\right) \tag{7}$$

where the adjustment term is derived by

$$\begin{aligned} &k_{n,i,j} = \max\ k \\ &\text{s.t. } 0 \le k \le \left|l_{n,i,j}\right|,\ k \in Z \\ &\quad \tau_{n,i,j}(k) \le T_{n,i,j}. \end{aligned} \tag{8}$$

In this way, we can flexibly regulate the resultant bits based on HVS and maintain similar visual quality at lower bitrates.

To further clarify our idea, a toy example is given as follows. Suppose a transform coefficient $w = 10$ and the corresponding quantized coefficient $l = 5$. Let the JND threshold $J^* = 5$ and the adjustment threshold $T = 1$. In conventional methods w will not be suppressed as it is larger than J^*. But in our method l can be suppressed to $l' = 3$ which corresponds to the reconstructed coefficient $w' = 6$. Its reconstruction error is $e = 10 - 6 = 4$ and the corresponding JND-normalized error is $\tau = e/J^* = 4/5 < T = 1$.

All quantized coefficients of the H.264/AVC integer DCT transform can be adaptively suppressed according to (8). But since it is not easy to obtain the JND thresholds for the Hadamard transform, which the DC coefficients of luminance in the I16×16 mode and the DC coefficients of chrominance undergo additionally, we suppress these DC coefficients before the Hadamard transform for approximation. Namely they are suppressed by $T_{n,i,j} \cdot J^*_{n,i,j}$ at most toward zero.

[1]Though JND thresholds may be different for video with distortion, here constant JND thresholds are used for approximation.

B. Lagrange Multiplier Adaptation

Video encoders aim at minimizing the distortion D under a constraint R_c of the rate R, which can be formulated as

$$\begin{aligned} &\min D \\ &\text{s.t. } R \le R_c. \end{aligned} \tag{9}$$

Currently the most popular rate distortion optimization algorithm is the Lagrangian method [27], which converts the problem to minimizing the Lagrangian cost function

$$J = D + \lambda \cdot R \tag{10}$$

by the Lagrange multiplier λ. Based on the modeling of quantization distortion, the optimal Lagrange multiplier for H.264/AVC has been shown to be [19], [27]–[29]

$$\lambda\left(Q_{\text{step}}\right) = c \cdot Q^2_{\text{step}} \text{ or } 0.85 \cdot 2^{\left(QP\left(Q_{\text{step}}\right)-12\right)/3} \tag{11}$$

where c is a constant, Q_{step} is the quantization step size, and $QP\left(Q_{\text{step}}\right)$ is the quantization parameter depending on the quantization step size. The proposed coding framework, however, introduces new distortion except quantization. Therefore, the Lagrange multiplier should be adapted to accommodate the overall distortion.

Since the optimal Lagrange multiplier was derived for the normal case, we choose to adapt the Lagrange multiplier for the proposed framework by distortion equivalence to the normal case for simplicity. Specifically, the Lagrange multiplier is adapted by the equivalent quantization step size in the distortion sense. To this end, the distortion of coding with coefficient suppression has to be examined.

H.264/AVC adopts scalar quantization in the transform domain. Because distortion is usually calculated statistically, the distribution of the transform coefficients of the residuals has to be determined first. As the H.264/AVC transform correlates with the classic DCT transform via linear scaling, without loss of generality, analysis is conducted in the classic DCT domain. (Detailed derivation of the connections between the H.264/AVC transform and the classic DCT transform can be found in the appendix.)

1) *Distribution Parameter Estimation:* The transform coefficients of prediction residuals can be assumed as a zero-mean Laplace distribution [30]

$$f(x) = \frac{1}{\sqrt{2}\sigma} e^{\left(-\frac{\sqrt{2}}{\sigma}|x|\right)} \tag{12}$$

where x represents the transform coefficient and σ is the standard deviation. So the distribution of the transform coefficients depends on the standard deviations, which are estimated on a subband by subband basis as follows.

Let $r_{uv}\ (0 \le u, v \le N-1)$ denote the prediction residuals of a $N \times N$ block in the pixel domain, whose standard deviation can be approximated by [31]

$$\sigma_f \approx \sqrt{2} \cdot \frac{\sum_{u,v=0}^{N-1} |r_{uv}|}{N \times N}. \tag{13}$$

If $\sigma_F(i, j)$ represents the standard deviation of the (i, j)th DCT coefficient, then it satisfies the relation [31], [32]

$$\sigma_F^2(i, j) = \sigma_f^2 \left[CRC^T\right]_{i,i} \left[CRC^T\right]_{j,j} \tag{14}$$

where C is the DCT transform matrix

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{N-1} \\ \rho & 1 & \rho & \cdots & \\ \rho^2 & \rho & 1 & & \\ \vdots & & & \ddots & \\ \rho^{N-1} & & & & 1 \end{bmatrix} \tag{15}$$

with $\rho = 0.6$ [31], and $[\cdot]_{i,i}$ is the (i, i)th element of the matrix. So the standard deviations of the transform coefficients can be determined by the residuals in the pixel domain. In the $N = 4$ scenario, for example, the standard deviation of the DC coefficient is

$$\sigma_F(0, 0) = \sqrt{5.6074}\sigma_f. \tag{16}$$

By substituting (13) into (16) we can further obtain

$$\sigma_F(0, 0) \approx \sqrt{5.6074} \cdot \sqrt{2} \cdot \frac{\sum_{u,v=0}^{3} |r_{uv}|}{4 \times 4}. \tag{17}$$

From the above, we can see that the distribution of the transform coefficients can be obtained as long as the prediction residuals are available. Unfortunately, the final residuals are affected by the Lagrange multiplier used in coding, which results in a chicken and egg dilemma at the stage of Lagrange multiplier adaptation. To solve the dilemma, we have to estimate the residuals beforehand. Specifically, they are approximated by the minimum residuals of interprediction in interframes, while in intraframes they are approximated by the minimum residuals of intraprediction with coefficient suppression included. In this way, we can obtain the estimation of the standard deviations and further the distribution of the transform coefficients.

2) *Distortion Formulation:* Now with the distribution of the transform coefficients available, we can turn to the distortion examination for both the normal case and the proposed method.

The mean squared error of uniform quantization for a signal with the probability density function $f(x)$ can be expressed as

$$\begin{aligned} &\varepsilon^2\left(Q_{\text{step}}, DZ\right) = \int_{-DZ}^{DZ} x^2 \cdot f(x)\, dx \\ &+ \sum_{i=0}^{\infty} \Big(\int_{DZ+i\cdot Q_{\text{step}}}^{DZ+(i+1)\cdot Q_{\text{step}}} \left(x - (i+1)\cdot Q_{\text{step}}\right)^2 \cdot f(x)\, dx \\ &+ \int_{-DZ-(i+1)\cdot Q_{\text{step}}}^{-DZ-i\cdot Q_{\text{step}}} \left(x + (i+1)\cdot Q_{\text{step}}\right)^2 \cdot f(x)\, dx \Big) \end{aligned} \tag{18}$$

where Q_{step} and DZ are, respectively, the quantizer's quantization step size and dead zone size. If the signal obeys the Laplace distribution with the standard deviation σ as shown in (12), (18) can be simplified as [33]

$$\varepsilon_{\text{Lap}}^2\left(\sigma, Q_{\text{step}}, DZ\right) = 2\lambda^2 - \frac{2\lambda\Delta e^{-\alpha/\lambda} e^{-\Delta/2\lambda}}{\left(1 - e^{-\Delta/\lambda}\right)} \left[\frac{\alpha}{\lambda} + 1\right] \tag{19}$$

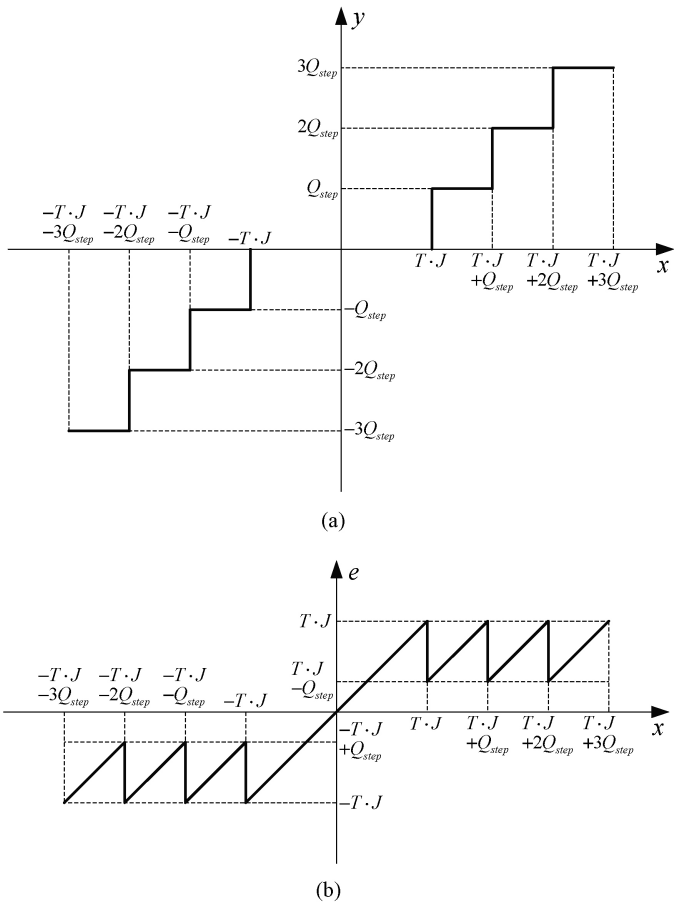


Fig. 2. Illustration of (a) quantization along with coefficient suppression and (b) quantization error.

where $\lambda = \sigma/\sqrt{2}$, $\alpha = DZ - Q_{\text{step}}/2$, $\Delta = Q_{\text{step}}$. For H.264/AVC, the preferred quantization offsets for intra and interprediction are $Q_{\text{step}}/3$ and $Q_{\text{step}}/6$, respectively [25]. So the corresponding dead zone size is

$$DZ_{\text{nor}}\left(Q_{\text{step}}\right) = \begin{cases} 2Q_{\text{step}}/3 & \text{for intraprediction} \\ 5Q_{\text{step}}/6 & \text{for interprediction} \end{cases}. \tag{20}$$

By this way, we can obtain approximately the quantization distortion of the transform coefficient with the standard deviation σ_F in the normal case

$$\varepsilon_{\text{nor}}^2\left(\sigma_F, Q_{\text{step}}\right) \approx \varepsilon_{\text{Lap}}^2\left(\sigma_F, Q_{\text{step}}, DZ_{\text{nor}}\left(Q_{\text{step}}\right)\right). \tag{21}$$

In coding with adaptive coefficient suppression, let J and T denote, respectively, the JND threshold and the JND-normalized adjustment threshold for a transform coefficient. If $T \cdot J \le DZ_{\text{nor}}\left(Q_{\text{step}}\right)$, no suppression will be allowed for the coefficient according to (8). But if $T \cdot J$ is larger, suppression constrained by the maximum error of $T \cdot J$ will be performed. In this scenario, quantization along with coefficient suppression can be illustrated in Fig. 2(a), where the x-axis represents the original coefficient and the y-axis represents the reconstructed coefficient. And the quantization error can be illustrated in Fig. 2(b), where the x-axis represents still the original coefficient and the e-axis represents the error.

Therefore, quantization along with coefficient suppression has the dead zone size

$$DZ_{pro}\left(Q_{\text{step}}\right)=\begin{cases} DZ_{\text{nor}}\left(Q_{\text{step}}\right) & T\cdot J\le DZ_{\text{nor}}\left(Q_{\text{step}}\right) \\ T\cdot J & T\cdot J> DZ_{\text{nor}}\left(Q_{\text{step}}\right)\end{cases} \tag{22}$$

and in this case the distortion of the transform coefficient with the standard deviation σ_F is approximately

$$\varepsilon^2_{pro}\left(J,T,\sigma_F,Q_{\text{step}}\right)\approx\varepsilon^2_{\text{Lap}}\left(\sigma_F,Q_{\text{step}},DZ_{pro}\left(Q_{\text{step}}\right)\right). \tag{23}$$

3) *Distortion Equivalence:* Now based on the above distortion formulation, distortion equivalence can be conducted in the closed form. And in view of the important role, the DC component plays in video quality, the DC component is selected as the representative component for simplicity.

In the 4×4 block scenario ($N=4$), there are $K=16$ blocks in a macroblock. Let J_n, T_n, and $\sigma_{F,n}$ denote, respectively, the JND threshold, the adjustment threshold, and the standard deviation of the DC coefficient for the nth block. Then in coding with coefficient suppression, the average distortion of the DC component for the macroblock is

$$\overline{\varepsilon^2_{pro}}\left(Q_{\text{step}}\right)=\sum_{n=1}^{K}\varepsilon^2_{pro}\left(J_n,T_n,\sigma_{F,n},Q_{\text{step}}\right)\Big/K \tag{24}$$

where Q_{step} is the preset quantization step size. In comparison, the average distortion in the normal case is

$$\overline{\varepsilon^2_{\text{nor}}}\left(Q_{\text{step}}\right)=\sum_{n=1}^{K}\varepsilon^2_{\text{nor}}\left(\sigma_{F,n},Q_{\text{step}}\right)\Big/K. \tag{25}$$

Thus if Q'_{step} represents the equivalent quantization step size for coding with coefficient suppression, we have the relation

$$\overline{\varepsilon^2_{pro}}\left(Q_{\text{step}}\right)=\overline{\varepsilon^2_{\text{nor}}}\left(Q'_{\text{step}}\right) \tag{26}$$

which suffices to give the value of Q'_{step} based on the above derivation. Since Q'_{step} is calculated in the distortion sense, now we can substitute it into (11) to obtain $\lambda\left(Q'_{\text{step}}\right)$. This new multiplier has accommodated the overall distortion with the capability of coding the macroblock with adaptive coefficient suppression.

To simplify the calculation of Q'_{step}, we search the closest quantization step size in this paper. Let S_Q denote the set of the quantization step sizes in the H.264/AVC standard. The equivalent step size is calculated as

$$Q'_{\text{step}}=\arg\min_{Q\in S_Q}\left|\overline{\varepsilon^2_{pro}}\left(Q_{\text{step}}\right)-\overline{\varepsilon^2_{\text{nor}}}\left(Q\right)\right| \tag{27}$$

which gives the step size minimizing the error for approximation.

Considering the possible model and calculation inaccuracy, the equivalent step size is bounded by twice the preset step size under the experimental conditions in this paper to keep reasonable rate distortion optimization. Under a typical experimental condition, the obtained equivalent quantization step sizes of representative frames are illustrated in Fig. 3, where higher gray levels indicate higher equivalent quantization step sizes. We can see that the equivalent quantization step sizes for the Lagrange multiplier are decently tuned according to prediction residuals and visual sensitivity for different macroblocks. Consequently, different from some previous improved Lagrange multiplier selection methods [30], [34], here we adapt the Lagrange multiplier in a local and perceptually adaptive way.

III. JND Related Perceptual Parameter Estimation

A. JND in the H.264/AVC Integer Transform Domain

Unlike many previous standards utilizing the classic DCT transform, H.264/AVC introduces a low-complexity 4×4 DCT transform for energy compaction, which can be computed exactly in integer arithmetic without transform mismatch. Usually a new design is required to build JND models from scratch in a new transform domain. But considering the new transform's connections with the classic DCT transform, we propose to compute the JND thresholds in the H.264/AVC transform domain directly in two steps. Firstly, JND for the classic 4×4 DCT is adapted from an existing model. Then based on the resultant distribution, the JND thresholds in the H.264/AVC transform domain are approximated by means of linear scaling.

1) *JND in the Classic DCT Domain:* As DCT is widely used in image and video processing, many JND models have been developed in the classic DCT domain [35]–[37]. Here we adopt the model in the recent literature [37] and adapt it to the 4×4 DCT.

The adopted JND model is expressed as the product of a basic threshold and some modulation factors. Let n denote the index of a block and (i,j) the index of a DCT coefficient. The corresponding JND is modeled as [37]

$$\begin{aligned}T_{JND}(n,i,j)&=T_{\text{basic}}(n,i,j)\times F_{\text{lum}}(n)\\&\times F_{\text{contrast}}(n,i,j)\times F_{\text{temporal}}(n,i,j)\end{aligned} \tag{28}$$

where T_{JND} is the JND threshold and T_{basic} is the basic threshold. The luminance adaptation factor F_{lum}, the contrast masking factor F_{contrast}, and the temporal modulation factor F_{temporal} act as the constituent modulation factors.

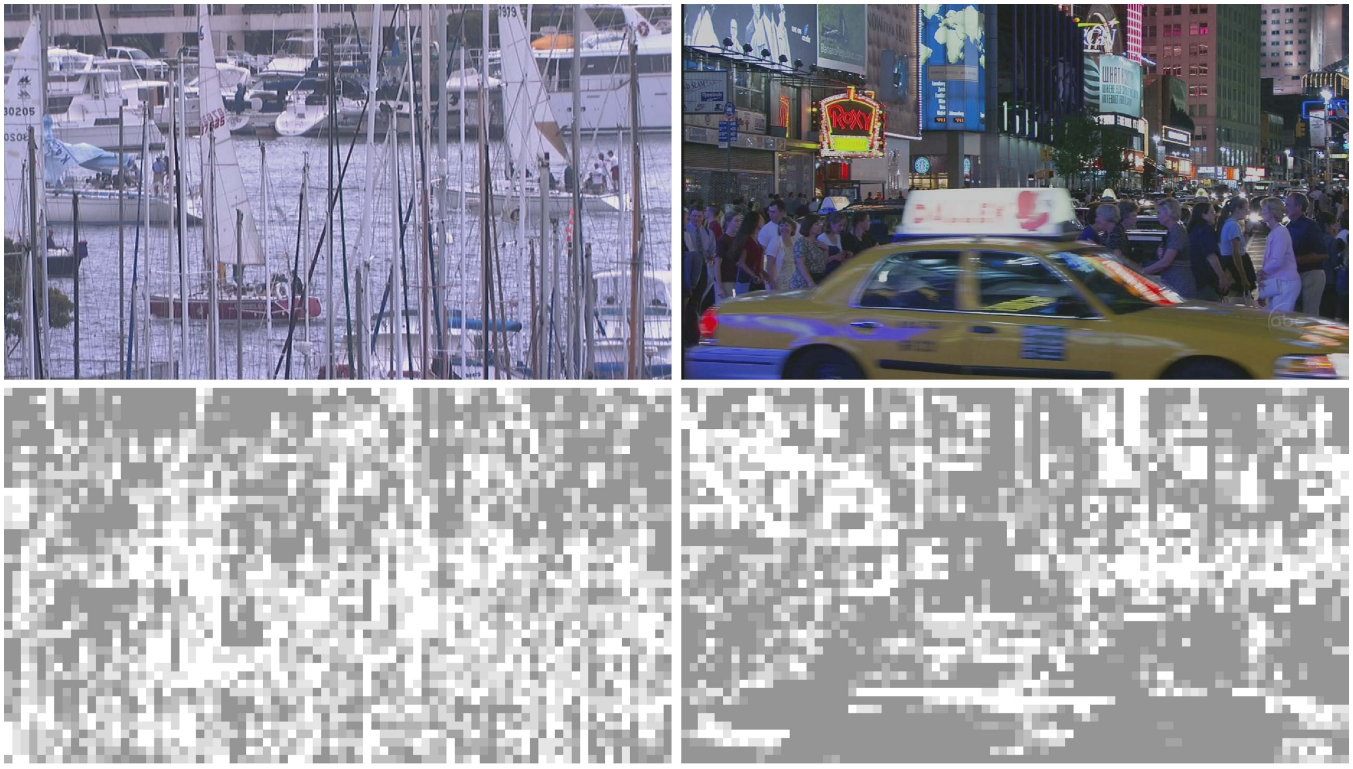
T_{basic} accounts for the visual sensitivity to spatial frequencies. Let θ_x and θ_y be the horizontal and vertical visual angles of a pixel, respectively, and N be the dimension of the DCT block. The spatial frequency of the (i,j)th DCT subband is [37]

$$\omega_{ij}=\frac{1}{2N}\sqrt{\left(i/\theta_x\right)^2+\left(j/\theta_y\right)^2}. \tag{29}$$

Then T_{basic} can be expressed as [37]

$$T_{\text{basic}}(n,i,j)=s\cdot\frac{1}{\phi_i\phi_j}\cdot\frac{\exp\left(c\cdot\omega_{ij}\right)/\left(a+b\cdot\omega_{ij}\right)}{r+(1-r)\cdot\cos^2\varphi_{ij}} \tag{30}$$

where ϕ_i and ϕ_j are DCT normalization factors, and φ_{ij} stands for the directional angle of the corresponding DCT component. Here we calculate the basic threshold in the $\times4$ DCT scenario ($N=4$; $0\le i,j\le3$), and similar psychophysical experiments as in [37] are conducted to get the fitted values $a=0.336$, $b=0.074$, and $c=0.238$. The setting of the other parameters can be found in [37].



(a) (b)

Fig. 3. From top to bottom: the example frame and its equivalent quantization step sizes (enhanced for visibility) for (a) 88th frame of *Harbor* and (b) 43rd frame of *Night* under QP=28.

F_{lum} describes the impact of luminance to human perception. Here we first compute the average intensity value $\overline{I}$ of every 4×4 block. Then F_{lum} is determined as [37]

$$F_{\text{lum}}(n) = \begin{cases} \left(60 - \overline{I}\right)/150 + 1 & \overline{I} \leq 60 \\ 1 & 60 < \overline{I} < 170 \\ \left(\overline{I} - 170\right)/425 + 1 & \overline{I} \geq 170 \end{cases} \tag{31}$$

which shows the visual sensitivity is low in the dark and light regions.

Usually distortion is easily observed in the smooth and edge areas but not in those with high texture energy, which is taken into account in the model via F_{contrast}. First, the Canny operator [38] is applied to detect the edge pixels of a given image. Then based on the percentage of edge pixels ρ_{edge} in the centered 8×8 block, every 4×4 block is classified into one of three types as [37]

$$\text{Block type} = \begin{cases} \text{Plane} & \rho_{\text{edge}} \leq 0.1 \\ \text{Edge} & 0.1 < \rho_{\text{edge}} \leq 0.2 \\ \text{Texture} & \rho_{\text{edge}} > 0.2 \end{cases}. \tag{32}$$

Finally with the elevation factor

$$\Psi(n, i, j) = \begin{cases} 1 & \text{for Plane and Edge block} \\ 2.25 & \text{for } \left(i^2 + j^2\right) \leq 4 \text{ in Texture block} \\ 1.25 & \text{for } \left(i^2 + j^2\right) > 4 \text{ in Texture block} \end{cases} \tag{33}$$

here F_{contrast} equals to Ψ for $\left(i^2 + j^2\right) \leq 4$ in plane and edge blocks and otherwise is calculated as [37]

$$\Psi(n, i, j) \times \min\left(4, \max\left(1, \left(\frac{C(n,i,j)}{T_{\text{basic}}(n,i,j) \times F_{\text{lum}}(n)}\right)^{0.36}\right)\right) \tag{34}$$

where $C(n, i, j)$ is the (i, j)th DCT coefficient of the nth block.

Video is often characterized by the motion information, and its effects on the visual sensitivity are reflected by F_{temporal}. Let f_s denote the spatial frequency and f_t the temporal frequency, which depends on the motion vectors, the frame rate, etc. Then F_{temporal} can be derived as [37]

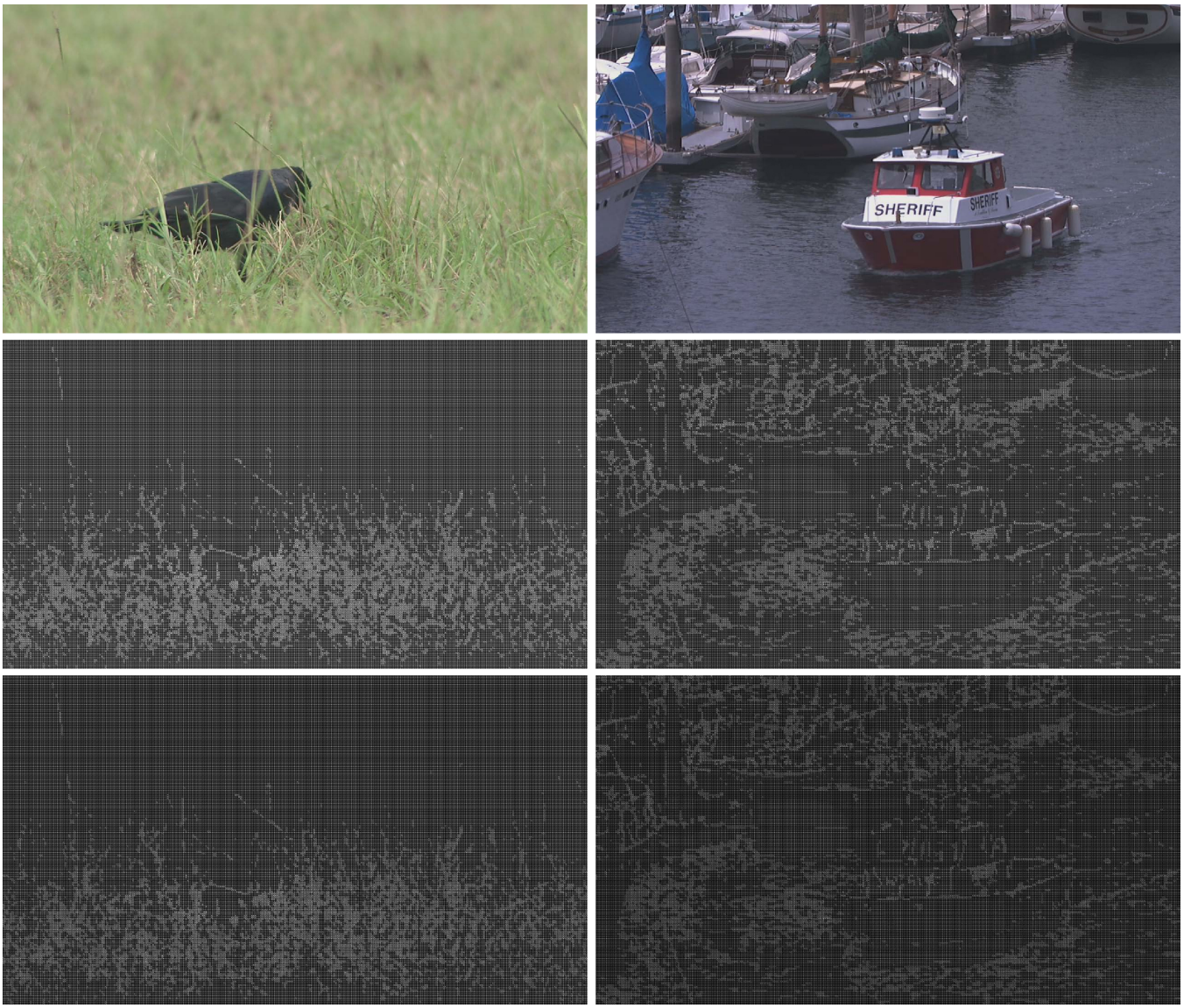
$$F_{\text{temporal}}(n, i, j) = \begin{cases} 1 & f_s < 5cpd \,\&\, f_t < 10\,\text{Hz} \\ 1.07^{(f_t - 10)} & f_s < 5cpd \,\&\, f_t \geq 10\,\text{Hz} \\ 1.07^{f_t} & f_s \geq 5cpd \end{cases}. \tag{35}$$

For more details the readers are referred to [37].

Finally, the JND threshold in the classic 4×4 DCT domain can be obtained by (28).

The above model applies to the luminance component of video. As for the chrominance components, similar behaviors as luminance are assumed [19], [39]. So in the case of the 4:2:0 format in this paper, we take the average of corresponding JND thresholds for luminance as that for chrominance at the 4×4 level.

2) *JND Translation for the H.264/AVC Integer Transform Domain:* The H.264/AVC transform is associated with the



(a) (b)

Fig. 4. From top to bottom: the example frame, and its JND thresholds in the classic DCT domain and the H.264/AVC transform domain (enhanced for visibility) for (a) 52nd frame of *Raven* and (b) 118th frame of *Sheriff*.

classic DCT transform approximately through element-wise scaling, of which the detailed derivation is provided in the appendix. Let $J_{4\times4}$ and $J^*_{4\times4}$ denote, respectively, the JND thresholds in the classic DCT domain and the H.264/AVC transform domain. Considering JND in different domains results from the same HVS mechanism, we have

$$J_{4\times4} \approx J^*_{4\times4} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \tag{36}$$

where $\otimes$ means element-wise multiplication, $a = 1/2$, $b = \sqrt{2/5}$ and $d = 1/2$ as defined in the appendix. So based on the JND thresholds in the classic DCT domain, the counterpart in the H.264/AVC transform domain can be approximated by

$$\begin{aligned} J^*_{4\times4} &\approx J_{4\times4} \otimes \begin{bmatrix} 1/a^2 & 2/ab & 1/a^2 & 2/ab \\ 2/ab & 4/b^2 & 2/ab & 4/b^2 \\ 1/a^2 & 2/ab & 1/a^2 & 2/ab \\ 2/ab & 4/b^2 & 2/ab & 4/b^2 \end{bmatrix} \\ &= J_{4\times4} \otimes \begin{bmatrix} 4 & 4\sqrt{5/2} & 4 & 4\sqrt{5/2} \\ 4\sqrt{5/2} & 10 & 4\sqrt{5/2} & 10 \\ 4 & 4\sqrt{5/2} & 4 & 4\sqrt{5/2} \\ 4\sqrt{5/2} & 10 & 4\sqrt{5/2} & 10 \end{bmatrix}. \end{aligned} \tag{37}$$

Under the experimental conditions in Section V, the JND thresholds of example frames are illustrated in Fig. 4, where high gray levels indicate high thresholds. It can be seen that the distribution corresponds well with the visual sensitivity.

(a) (b)

Fig. 5. From top to bottom: the example frame and its attention weight of 4×4 blocks for (a) first frame of *Cyclists* and (b) first frame of *SpinCalendar*.

B. Adjustment Threshold Distribution

It has been shown that the probability of detecting distortion by human perception increases monotonically with the JND-normalized error [26]. Thus the adjustment threshold, which controls the maximum JND-normalized error, determines the probability of error detection in images. But as we know, people do not pay equal attention to every part of an image. Hence in accordance with HVS the adjustment threshold can be regulated by visual saliency.

In this paper, a saliency algorithm is adopted to verify our method. The GBVS model [40], [41] is used to estimate the attention weight of every 4×4 block. The attention weight for representative frames is shown in Fig. 5, where higher gray levels indicate more salient blocks. Let $W_n \in [0, 1]$ denote the attention weight of the nth block by the model. Then by subjective experiments the block's adjustment threshold is determined as

$$T_n = \begin{cases} -4\,(T_{\max} - 1)\,W_n + T_{\max} & W_n < 0.25 \\ 1 & W_n \geq 0.25 \end{cases} \tag{38}$$

where $T_{\max}$ is the maximum adjustment threshold and is set to 2 here. In this way, residual adjustment is confined within the JND threshold in the most salient regions, while more residuals may be adjusted elsewhere.

IV. Implementation Issues

A. Complexity Consideration

Since both JND calculation and predictive coding builds upon motion vectors of video, it is preferred that motion estimation can be reused for reduced complexity. The differences between these two motion vectors have to be clarified first. Fig. 6(a) shows an exemplary structure of temporal predictive coding, which shows a representative reference relationship. It is obvious that the obtained motion vectors may not be relative to the immediate previous frames. But as video frames are displayed eventually in the original order, the motion vectors required by JND calculation shall be those relative to the immediate previous ones. To enable motion estimation reuse, motion vectors relative to the reference frame can be temporally scaled to approximate those required by JND as shown in Fig. 6(b). Specifically, if the motion vector obtained by forward motion estimation in the sense of minimum distortion is $\overrightarrow{MV}$, and the frame interval between the current frame and the reference frame is d, the motion vector for JND calculation can be approximated as

$$\overrightarrow{MV}_{JND} \approx \overrightarrow{MV} \Big/ d. \tag{39}$$

Here the motion vectors of I frames are assumed to be zero vectors in JND calculation for better practicality. Though this may lead to lower JND thresholds and less bit saving, it may not have a large impact as usually I frames do not account for a large percentage in coding.

As the method is fully compatible with the H.264/AVC standard, and many coefficients are suppressed toward zero, the computational complexity of the decoder is not increased. At the encoder side, from Fig. 1 we can see that extra complexity results mainly from the JND threshold calculation, the adjustment threshold calculation, the Lagrange multiplier adaptation, the adaptive coefficient suppression in coding, and

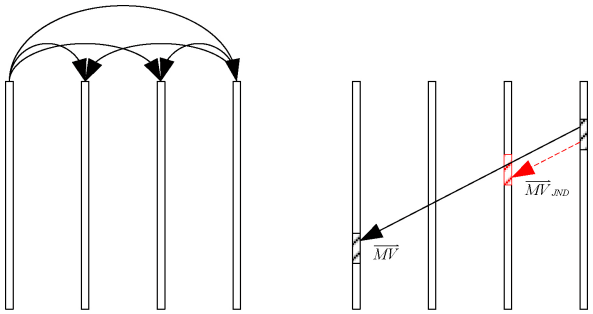


Fig. 6. Illustration of (a) representative reference relationship and (b) motion vector scaling.

the motion vector scaling. The computational complexity is analyzed briefly as follows.

1) The JND threshold is calculated for each transform coefficient. Except for the motion estimation, whose intermediate results may be shared by later coding, JND calculation involves floating-point computation. If c_{JND} denotes the computational cost for one transform coefficient, the complexity of JND calculation for a frame is $c_{JND} \cdot W \cdot H$ with W and H being, respectively, the frame width and height.
2) The adjustment threshold is calculated from the adopted attention model, which requires complex computation like feature extraction. But the attention weight can be calculated on downsampled images to reduce computation.
3) The Lagrange multiplier is adapted according to the equivalent quantization step size for each macroblock. As the distortion is formulated and the equivalent quantization step size is searched by closest distortion, computation is controllable in the adaptation. If c_λ denotes the computational cost for one Lagrange multiplier, the complexity of Lagrange multiplier adaptation for a frame is $c_\lambda \cdot M$ with M being the number of macroblocks in a frame.
4) After the JND threshold and the adjustment threshold are determined, coefficient suppression is easy to perform and needs only limited computation.
5) By motion estimation reuse, motion vector scaling also needs only limited computation.

Though the above steps may incur much extra computation, in practice, steps like JND threshold calculation can be carried out parallel to reduce the computation time. At present the method is more suitable for off-line video coding.

B. Discussion and Summary

In the salient regions, where the adjustment threshold is set to 1, it can be seen from (8) that if the coefficients are suppressed, the errors resulting from both suppression and quantization will not exceed the JND thresholds. Correspondingly, the errors of normal coding, which result from quantization only, can be thought not to exceed the JND thresholds either. Therefore, coding with coefficient suppression and normal coding are expected to have similar visual quality. In the nonsalient regions, though the adjustment threshold is slightly larger than 1 and slightly more errors may be caused, as their JND thresholds could be higher due to attention, similar effects can be expected.

Algorithm 1. Macroblock level encoding steps.

For *each macroblock of the sequence*

1 ***JND threshold calculation***
- 1.1 *Obtain the basic threshold T_{basic}, the luminance adaptation factor F_{lum} and the contrast masking factor $F_{contrast}$ of the JND model using (30), (31), (34), etc.*
- 1.2 *Search the motion vectors in the sense of minimum distortion.*
- 1.3 *Approximate the motion vectors $\overrightarrow{MV}_{JND}$ with respect to the immediate previous frame by temporal scaling using (39).*
- 1.4 *Calculate the temporal modulation factor $F_{temporal}$ and the final JND thresholds of the JND model using (35), (28), etc.*
- 1.5 *Translate the JND thresholds for the H.264/AVC transform domain using (37).*

2 ***Adjustment threshold calculation***
- 2.1 *Obtain the attention weight W_n based on the visual attention model.*
- 2.2 *Calculate the adjustment thresholds T_n using (38).*

3 ***Lagrange multiplier adaptation***
- 3.1 *Estimate the prediction residuals r_{uv} and the distribution parameters of the transform coefficients using (14), etc.*
- 3.2 *Estimate the average distortion ε^2_{pro} of the representative frequency component using (23), etc.*
- 3.3 *Calculate the equivalent quantization step size Q'_{step} in the distortion sense using (27), etc.*
- 3.4 *Calculate the adapted Lagrange multiplier $\lambda\left(Q'_{step}\right)$ using (11), etc.*

4 ***Specific macroblock encoding***
- 4.1 *Encode the macroblock with adaptive coefficient suppression based on the adapted Lagrange multiplier.*

End

TABLE I
SPECIFICATIONS OF THE USED LCD DISPLAY

Resolution	$1680 \times 1,050$
Viewing Angle (Horizontal/Vertical)	160°/160°
Colour Supported	16.7 Mil.
Brightness	300 cd/m^2
Contrast Ratio	DC 3000:1 (1000:1)
Response Time (G-to-G)	2 (GTG)

For better clarification of the proposed method, the method is summarized as a sequence of steps as shown in Algorithm 1. It should be noted that the intermediate results obtained during motion estimation in step 1.2 may be saved for later use in the residual estimation in step 3.1 and the specific coding in step 4.1 to avoid duplicate calculations.

V. EXPERIMENTAL RESULTS

The proposed method is implemented in the JM 14.2 reference software [23]. It is compared with the original reference software and the recently proposed method in [8], which implements perceptual coding based on the adjustment of quantization parameters and is compatible with the H.264/AVC standard. Here the evaluation is conducted with the first 151 frames of representative 1280×720 4:2:0 sequences—*Cyclists*, *Harbor*, *Night*, *Raven*, *Sheriff*, and *SpinCalendar* at 30 f/s. Group of pictures (GOP) of IBBPBBP... structure with one I frame inserted every 30 frames are considered. Two reference frames, 4×4 transform and CABAC are used during

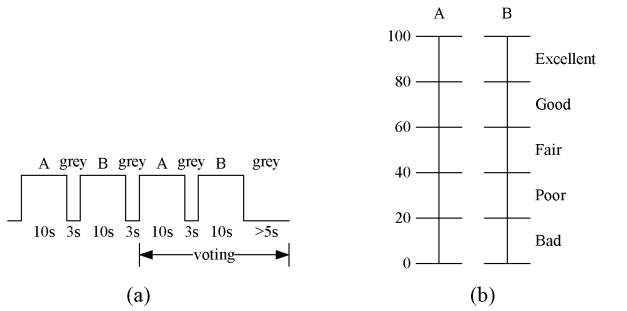


Fig. 7. Illustration of (a) procedure and (b) MOS scales of the DSCQS method.

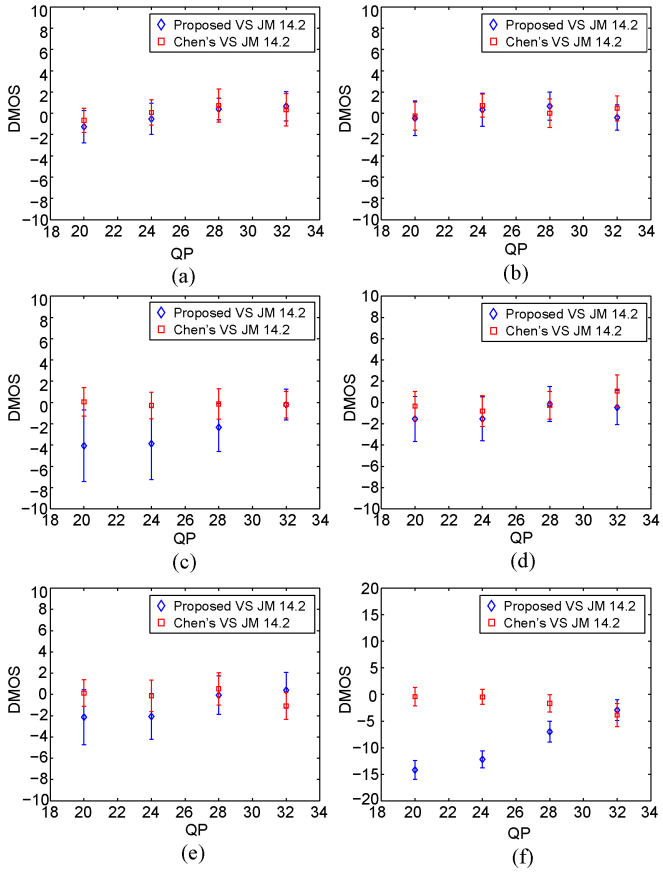


Fig. 8. DMOS scales of (a) *Cyclists*, (b) *Harbor*, (c) *Night*, (d) *Raven*, (e) *Sheriff*, and (f) *SpinCalendar* for the proposed method and Chen's method with respect to the reference software.

encoding. In the experiments, fixed quantization parameters $QP \in \{20, 24, 28, 32\}$ are enabled for comparison purposes.

A. Subjective Tests of the Proposed Method

First, quality of the encoded sequences is evaluated by subjective tests, which are conducted in a room illuminated by fluorescent lights. Fifteen observers, ten females and five females, are involved in the experiments over several days. The sequences are displayed on a 21″ display (SyncMaster 206BW), whose specifications are listed in Table I, and the viewing distance is about four times the image height.

First, video quality of the reference software and the proposed method is compared under the same QP using the double stimulus continuous quality scale (DSCQS) method [42], whose procedure is shown in Fig. 7(a). And the observers

TABLE II

COMPARISONS OF THE BITRATES FOR THE ENCODED SEQUENCES

Sequence	Preset QP	Bitrate (kb/s)			Bitrate Reduction Against JM 14.2 (%)	
		JM 14.2	Chen's	Proposed	Chen's	Proposed
Cyclists	20	7945.83	6889.50	5149.85	13.29	35.19
	24	3165.17	2660.42	2436.40	15.95	23.02
	28	1343.73	1103.82	1138.30	17.85	15.29
	32	658.92	543.16	612.40	17.57	7.06
Harbor	20	25104.43	23734.86	15822.41	5.46	36.97
	24	13496.66	12290.08	8843.39	8.94	34.48
	28	6054.17	5336.50	4557.15	11.85	24.73
	32	2909.30	2607.64	2588.25	10.37	11.04
Night	20	20306.64	18749.84	11330.19	7.67	44.20
	24	9688.57	8714.15	6239.72	10.06	35.60
	28	4507.60	4036.23	3430.19	10.46	23.90
	32	2311.90	2088.36	2050.42	9.67	11.31
Raven	20	7135.21	6568.93	4147.18	7.94	41.88
	24	3193.59	2850.05	2201.83	10.76	31.05
	28	1537.32	1346.20	1189.10	12.43	22.65
	32	803.07	705.19	710.89	12.19	11.48
Sheriff	20	13951.79	12986.99	7317.07	6.92	47.55
	24	6472.74	5838.45	3739.43	9.80	42.23
	28	2665.81	2361.96	1817.07	11.40	31.84
	32	1159.36	1032.24	963.12	10.96	16.93
SpinCalendar	20	25071.25	21394.72	11108.62	14.66	55.69
	24	7878.49	5930.58	4548.43	24.72	42.27
	28	2653.01	2194.53	2046.35	17.28	22.87
	32	1315.22	1129.24	1177.62	14.14	10.46
Average	–	–	–	–	12.18	28.32

are trained in advance to make them understand their tasks. Here every stimulus of DSCQS is a 10 s video constructed by composition of the encoded sequence of one method. And the displaying order of the two videos is random and unknown to the observers. During the voting time, the observers are asked to give their mean opinion score (MOS) scales from the continuous scales ranging from 0 to 100 as shown in Fig. 7(b). And the differential mean opinion score (DMOS) scales are calculated as the MOS scales of the stimuli of the proposed method minus those of the stimuli of the reference software.

The finally obtained average DMOS scales for the proposed method are shown in Fig. 8, where significantly distant data have been discarded from the statistics. It can be seen that the DMOS scales on the whole are quite close to 0, which shows the proposed method can produce visually similar video quality as the reference software. The reason why comparatively *SpinCalendar* has lower DMOS scales than the other sequences is that the characters and the dense but regular black stripes, which draw the attention of some observers for cognitive reasons, are mistaken for inconspicuous textures in the adopted JND and attention models and get impaired by multiplier adaptation and coefficient suppression during encoding. But this does not cause too much degradation from the test results and can be improved by adopting better JND and attention models. Similarly, that human faces are mistaken for inconspicuous textures explains the slightly lower DMOS scales of *Night*.

Subsequently, similar subjective tests are conducted for method in [8], and the results are also shown in Fig. 8. It can be seen that the method by Chen and Guillemot [8] produces

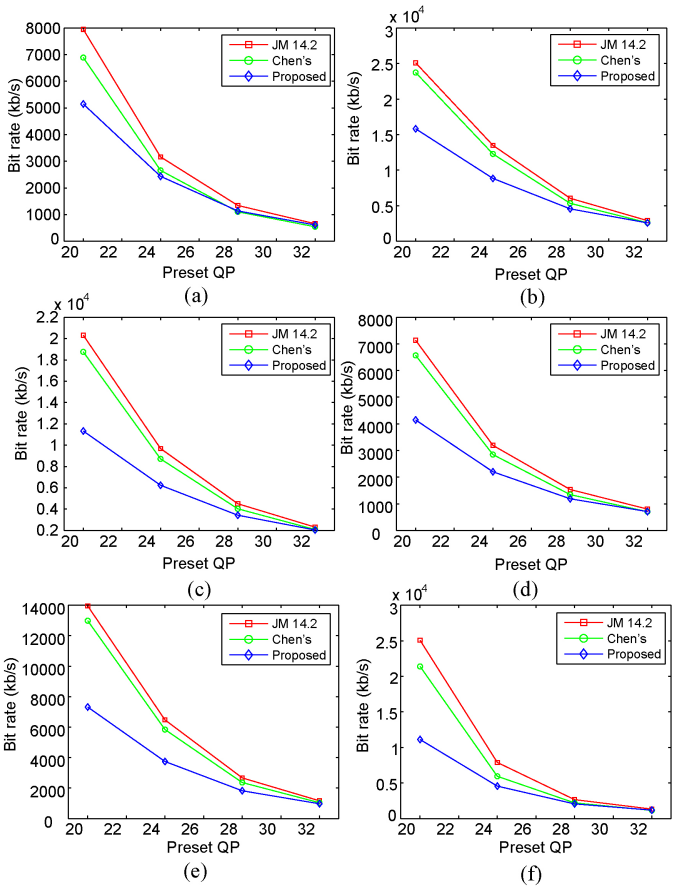


Fig. 9. Comparisons of the bitrates for (a) *Cyclists*, (b) *Harbor*, (c) *Night*, (d) *Raven*, (e) *Sheriff* and (f) *SpinCalendar*.

visually similar video quality as the reference software as well.

B. Bitrate Comparisons with Anchor Algorithms

Since both the proposed method and that proposed by Chen and Guillemot [8] have similar visual quality as the reference software, now we turn to the bitrate differences of the three methods. And the bitrates of the test sequences under different preset QPs are shown in Table II and Fig. 9.

It can be seen that compared with the reference software, the bitrates can be reduced a lot by our method. The reason why bitrate reduction decreases as QP increases is further explained. In the case of low QPs, many nonzero coefficients are available after quantization, which leaves a wide margin of bit saving via coefficient suppression. While under high QPs, there exist fewer nonzero quantized coefficients for suppression, which enables only limited bitrate reductions.

The bitrates and the bitrate reductions of the method in [8] are also shown in Table II and Fig. 9. It should be noted that in the experiments the adjusted QPs in Chen and Guillemot's [8] method have been limited to be no less than the preset QP to show its bit saving. It can be observed that compared with Chen and Guillemot's [8] method, more bit saving can be obtained by our method in most cases. The reason why our method does not perform as well as Chen and Guillemot's [8] method for some sequences under high QPs is as follows. In

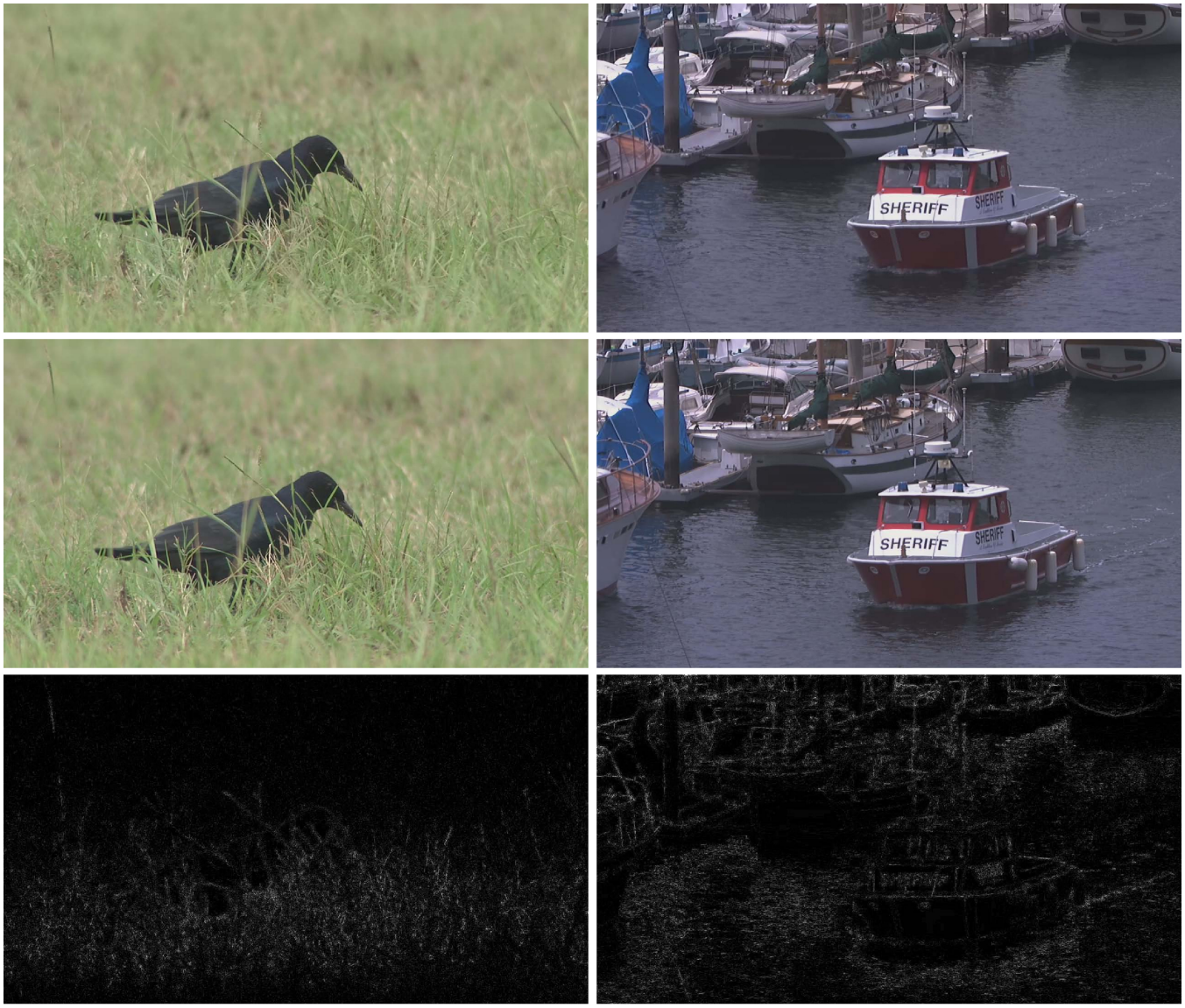
TABLE III
COMPARISONS OF THE SSIM METRICS AND PSNR FOR THE ENCODED SEQUENCES

Sequence	Preset QP	SSIM			PSNR (dB)		
		JM 14.2	Chen's	Proposed	JM 14.2	Chen's	Proposed
Cyclists	20	0.9693	0.9675	0.9614	42.83	42.44	40.99
	24	0.9542	0.9521	0.9474	40.73	40.33	39.52
	28	0.9354	0.9318	0.9279	38.84	38.32	37.99
	32	0.9143	0.9094	0.9077	37.05	36.45	36.55
Harbor	20	0.9892	0.9885	0.9763	41.92	41.51	36.29
	24	0.9797	0.9784	0.9664	38.80	38.29	35.01
	28	0.9635	0.9609	0.9516	35.78	35.21	33.67
	32	0.9375	0.9329	0.9304	33.20	32.65	32.43
Night	20	0.9803	0.9783	0.9632	42.29	41.84	36.12
	24	0.9666	0.9642	0.9514	39.28	38.85	35.25
	28	0.9491	0.9452	0.9347	36.73	36.26	34.20
	32	0.9221	0.9153	0.9124	34.33	33.82	33.26
Raven	20	0.9786	0.9781	0.9659	43.70	43.45	40.14
	24	0.9697	0.9690	0.9546	41.50	41.23	38.80
	28	0.9554	0.9541	0.9379	39.35	38.97	37.39
	32	0.9306	0.9282	0.9179	36.98	36.55	36.05
Sheriff	20	0.9781	0.9764	0.9532	42.81	42.44	37.78
	24	0.9606	0.9579	0.9362	39.96	39.57	36.54
	28	0.9323	0.9282	0.9087	37.33	36.92	35.21
	32	0.8944	0.8881	0.8778	35.07	34.65	34.12
SpinCalendar	20	0.9695	0.9652	0.9447	40.72	39.98	35.38
	24	0.9443	0.9399	0.9289	37.40	36.79	34.35
	28	0.9263	0.9203	0.9114	35.34	34.73	33.24
	32	0.9008	0.8912	0.8920	33.28	32.57	32.43

Chen and Guillemot's [8] method QP is restricted to increase further from the preset QP. But when the quantization distortion is large, the maximum coefficient adjustment amplitudes constrained by the original JND thresholds may be comparatively conservative for coefficient suppression in our method. In fact, videos with large quantization distortion usually have higher JND thresholds. Thus the performance of our method under high QPs can be improved by relaxing the adjustment amplitude constraint.

After the bitrates are compared, the popular SSIM [43] metrics along with the PSNR of the encoded sequences are presented in Table III for reference. It's known that PSNR does not reflect the visual perception effectively. Later we'll see that lower PSNR of our method is mainly caused by the quality degradation in the inconspicuous regions. Though the proposed method has slightly lower SSIM metrics, this does not necessarily mean lower visual quality as fewer visual sensitivity factors are taken into account in SSIM. In fact, such inaccuracy may also be observed for other similar perceptual quality metrics [19].

To demonstrate the bitrate reduction of the proposed method, the sequences are also compared on a frame-by-frame basis. Fig. 10 shows the selected frames of encoded sequences from the reference software and the proposed method along with their differences. It can be observed that the differences lie mainly in the inconspicuous regions, such as the weeds in *Raven* and the rippling water in *Sheriff*. Therefore, by adaptive suppression fewer bits are allocated to regions and frequency components that can bear more distortion; as a result similar visual quality can be maintained at lower bitrates.



(a) (b)

Fig. 10. From top to bottom: the reconstructed frame of the proposed method, the reconstructed frame of the reference software, and their differences (enhanced for visibility) for (a) 88th frame of *Raven* and (b) 102nd frame of *Sheriff* under QP=20.

VI. Conclusion

In this paper, a method of perceptual coding was proposed for the H.264/AVC standard. A JND-directed coefficient suppression tool was introduced to the coding framework to suppress the prediction residuals in an adaptive and normalized manner, and the Lagrange multiplier for rate distortion optimization was also adapted for the new tool. Besides, a JND translation formula was also derived for the H.264/AVC transform domain. Our method was fully compatible with the H.264/AVC standard, and significant bit saving can be obtained at similar visual quality to traditionally coded video. In the future, we plan to explore more accurate JND and attention models and related efficient computation to further improve the performance.

Appendix

To perform the translation of JND, we need to know about the connections between the classic DCT transform and the H.264/AVC transform. Let X and Y be the data block before and after transform. The classic 4×4 DCT transform can be expressed as [44], [45]

$$
\begin{aligned}
Y &= CXC^T \\
&= \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} X \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \\
&= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} X \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \\
&\otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}
\end{aligned} \tag{40}
$$

where

$$C = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \tag{41}$$

is the classic DCT transform matrix, $\otimes$ means element-wise multiplication, and related elements are

$$a = 1/2 \tag{42}$$

$$b = \sqrt{1/2}\cos\left(\pi/8\right) \approx 0.6533 \tag{43}$$

$$c = \sqrt{1/2}\cos\left(3\pi/8\right) \approx 0.2706 \tag{44}$$

$$d = c/b \approx 0.4142. \tag{45}$$

Now if we redefine [45]

$$b = \sqrt{2/5} \tag{46}$$

$$d = 1/2 \tag{47}$$

we will obtain [45]

$$\begin{aligned} Y &\approx \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix} X \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \right) \\ &\otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \\ &\otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}. \end{aligned} \tag{48}$$

Note that

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{49}$$

is the preferred H.264/AVC transform matrix [46], and from (48) we have

$$Y \approx \left(HXH^T\right) \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \tag{50}$$

so the H.264/AVC transform is associated with the classic DCT transform approximately through element-wise scaling.

ACKNOWLEDGMENT

The authors would like to thank J. Wang for his technical suggestions and R. Kaliski for polishing this paper.

REFERENCES

[1] *Advanced Video Coding for Generic Audiovisual Services*, Recommendation ITU-T H.264, Mar. 2010.

[2] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur, "Special section on the joint call for proposals on high efficiency video coding (HEVC) standardization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1661–1666, Dec. 2010.

[3] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Trans. Inform. Theory*, vol. 20, no. 4, pp. 525–536, Jul. 1974.

[4] C.-Y. Wang, S.-M. Lee, and L.-W. Chang, "Designing JPEG quantization tables based on human visual system," *Signal Process.: Image Commun.*, vol. 16, no. 5, pp. 501–506, Jan. 2001.

[5] Z. Yuan, H. Xiong, L. Song, and Y. F. Zheng, "Generic video coding with abstraction and detail completion," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Apr. 2009, pp. 901–904.

[6] E. Zhang, D. Zhao, Y. Zhang, H. Liu, S. Ma, and R. Wang, "A JND guided foveation video coding," in *Proc. Pacific-Rim Conf. Multimedia*, 2008, pp. 31–39.

[7] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Perceptual prefiltering for video coding," in *Proc. Int. Symp. Intell. Multimedia, Video Speech Process.*, Oct. 2004, pp. 510–513 .

[8] Z. Chen and C. Guillemot, "Perceptually-friendly H.264/AVC video coding based on foveated just-noticeable-distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 806–819, Jun. 2010.

[9] C.-W. Tang, "Spatiotemporal visual considerations for video coding," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 231–238, Feb. 2007.

[10] C.-W. Tang, C.-H. Chen, Y.-H. Yu, and C.-J. Tsai, "Visual sensitivity guided bit allocation for video coding," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 11–18, Feb. 2006.

[11] X. Yang, W. Lin, Z. Lu, X. Lin, S. Rahardja, E. Ong, and S. Yao, "Rate control for videophone using local perceptual cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 4, pp. 496–507, Apr. 2005.

[12] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 134–139, Jan. 2008.

[13] Z. Chen, J. Han, and K. N. Ngan, "Dynamic bit allocation for multiple video object coding," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1117–1124, Dec. 2006.

[14] X. Yang, W. Lin, Z. Lu, E. Ong, and S. Yao, "Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 742–752, Jun. 2005.

[15] H. Cheng, A. Kopansky, and M. A. Isnardi, "Reduced resolution residual coding for H.264-based compression system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May. 2006, pp. 3486–3489.

[16] B. Schuur, T. Wedi, S. Wittmann, and T. Palfner, "Frequency selective update for video coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 1709–1712.

[17] C.-M. Mak and K. N. Ngan, "Enhancing compression rate by just-noticeable distortion model for H.264/AVC," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 609–612.

[18] H. Chen, R. Hu, J. Hu, and Z. Wang, "Temporal color just noticeable distortion model and its application for video coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2010, pp. 713–718.

[19] M. Naccari and F. Pereira, "Advanced H.264/AVC-based perceptual video coding: Architecture, tools, and assessment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 766–782, Jun. 2011.

[20] M. Naccari and F. Pereira, "Integrating a spatial just noticeable distortion model in the under development HEVC codec," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, May 2011, pp. 817–820.

[21] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-inspired divisive normalization for perceptual video coding," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1657–1660.

[22] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Chichester, U.K.: John Wiley & Sons Ltd., 2003.

[23] *JM 14.2 Reference Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[24] *x264 rev. 602 Reference Software* [Online]. Available: http://www.videolan.org/developers/x264.html
[25] G. J. Sullivan and S. Sun, "On dead-zone plus uniform threshold scalar quantization," in *Proc. SPIE Int. Conf. Visual Commun. Image Process.*, 2005, pp. 1041–1052.
[26] I. Hontsch and L. J. Karam, "Adaptive image coding with perceptual distortion control," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 213–222, Mar. 2002.
[27] G. J. Sullivan and T. Wiegand, "Rate–distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
[28] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2001, pp. 542–545.
[29] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.
[30] X. Li, N. Oertel, A. Hutter, and A. Kaup, "Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 193–205, Feb. 2009.
[31] I-M. Pao and M.-T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 608–616, Jun. 1999.
[32] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[33] D. S. Turaga, Y. Chen, and J. Caviedes, "No reference PSNR estimation for compressed pictures," *Signal Process.: Image Commun.*, vol. 19, no. 2, pp. 173–184, 2004.
[34] M. Jiang and N. Ling, "On Lagrange multiplier and quantizer adjustment for H.264 frame-layer video rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 663–669, May 2006.
[35] X. H. Zhang, W. S. Lin, and P. Xue, "Improved estimation for just-noticeable visual distortion," *Signal Process.*, vol. 85, no. 4, pp. 795–808, 2005.
[36] Y. Jia, W. Lin, and A. A. Kassim, "Estimating just-noticeable distortion for video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 7, pp. 820–829, Jul. 2006.
[37] Z. Wei and K. N. Ngan, "Spatio-temporal just noticeable distortion profile for grey scale image/video in DCT domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 337–346, Mar. 2009.
[38] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
[39] H. R. Wu and K. R. Rao, *Digital Video Image Quality and Perceptual Coding*. Boca Raton, FL: CRC Press, 2006.
[40] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007.
[41] J. Harel, *GBVS Reference Codes* [Online]. Available: http://www.klab.caltech.edu/ harel/share/gbvs.php
[42] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, Recommendation ITU-R BT.500-12, Sept. 2009.
[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[44] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Boston: Academic Press, 1990.
[45] A. Hallapuro, M. Karczewicz, and H. Malvar, "Low complexity transform and quantization—part I: Basic implementation," *JVT-B038*, 2002.
[46] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.

Zhengyi Luo received the B.S. degree in information engineering from the Nanjing University of Posts and Telecommunications, Jiangsu, China, in 2004, and the M.S. degree in electronic engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2007. He is currently pursuing the Ph.D. degree at the Institute of Image Communication and Information Processing from the same university.

His current research interest includes video coding.

Li Song (M'08) received the B.Eng. and the M.S. degrees both from the Nanjin University of Science and Technology, Jiangsu, China, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, in 1997, 2000, and 2005, respectively.

He then joined SJTU as a Faculty in the Department of Electrical Engineering, and has been an Associate Professor since 2009. He was also a Visiting Associate Professor in Santa Clara University, CA, from 2011 to 2012.

Dr. Song has more than 100 publications in the fields of video coding, image processing, and pattern recognition, 15 issued patents and several MPEG/JVT/JCTVC international standard contributions. He has served as an Associate Editor for the *Journal of Springer Multidimensional Systems and Signal Processing* and a Publicity Chair of the 2013 IEEE International Conference on Multimedia and Expo, and a Session Chair of the 2010 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting. He is a TC member of the IEEE CAS Visual Signal Processing and Communications and the IEEE Communication Society Multimedia Communications Technical Committee (MMTC), and the Technical Program Committee for many international conferences. He received the 2010 International Conference on Wireless Communications and Signal Processing Best Paper Award and several research awards from SJTU.

Shibao Zheng (M'05) received both the B.S. and M.S. degrees in electronic engineering from Xidian University, Xi'an, China in 1983 and 1986, respectively.

He is currently a Professor and the Vice Director of Elderly Health Information and Technology Institute of Shanghai Jiao Tong University (SJTU), Shanghai, China. He is also the Professor Committee Member of Shanghai Key Laboratory of Digital Media Processing and Transmission, and Commissioner of Shanghai Communication Society Multi-media Division. His current research interests include urban image surveillance systems, intelligent video analysis, spatial information systems and elderly health technology.

Nam Ling (S'88–M'90–SM'99–F'08) received the B.Eng. degree from the National University of Singapore, Singapore, in 1981, and the M.S. and Ph.D. degrees from the University of Louisiana, Lafayette, LA, USA, in 1985 and 1989, respectively.

He is currently the Sanfilippo Family Chair Professor of Santa Clara University, CA, where he is also the Chair in the Department of Computer Engineering. From 2002 to 2010, he was an Associate Dean with the Santa Clara University School of Engineering, Santa Clara, CA, United States. Currently, he is also a Consulting Professor with the National University of Singapore, a Guest Professor for Shanghai Jiao Tong University, Shanghai, China, and a Tsuiying Chair Professor for Lanzhou University, Gansu, China. He has more than 160 publications and standard contributions, including a book, in the fields of video coding and systolic arrays.

Dr. Lin is an IET Fellow. He was named the IEEE Distinguished Lecturer twice and received the IEEE ICCE Best Paper Award (First Place). He was a recipient of six awards from Santa Clara University, four at the University level (Outstanding Achievement, Recent Achievement in Scholarship, President's Recognition, and Sustained Excellence in Scholarship) and two at the School/College level (Researcher of the Year and Teaching Excellence). He was a Keynote Speaker for IEEE APCCAS, VCVP, JCPC, IEEE ICAST, IEEE ICIEA, and IET FC & U-Media, as well as a Distinguished Speaker for IEEE ICIEA. He has served as a General Chair/Co-Chair for IEEE Hot Chips, VCVP, and IEEE ICME. He has also served as a Technical Program Co-Chair for IEEE ISCAS, APSIPA ASC, IEEE APCCAS, IEEE SiPS, DCV, and IEEE VCIP. He was a Technical Committee Chair for IEEE CASCOM TC and IEEE TCMM, and has served as a Guest Editor/Associate Editor for IEEE TCAS-I, IEEE J-STSP, JSPS, and Springer *Journal of MSSP* journals. He has delivered more than 110 invited colloquia worldwide and has served as a Visiting Professor/Consultant/Scientist/Scholar for many institutions and companies. He is an IEEE Fellow due to his contributions to video coding algorithms and architectures.