Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A new all-zero 4×4 block determination rule for integer transform and quantization in AVS-M encoder*

LIU Dong-hua†, CHEN Yi-qiang

(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*)

†E-mail: dhliu@shict.cn

Received Dec. 2, 2005; revision accepted Feb. 17, 2006

Abstract: A new all-zero block determination rule was used to reduce the complexity of the AVS-M encoder. It reuses the sum of absolute difference of 4×4 block obtained from motion estimation or intra prediction as parameters so that the determination threshold need to be computed only once when quantization parameter (QP) is invariable for given video sequence. This method avoids a lot of computation for transform, quantization, inverse transform, inverse quantization and block reconstruction. Simulation results showed that it can save about 20%~50% computation without any video quality degradation.

Key words: AVS-M, Integer DCT transform, Quantization, All-zero 4×4 block, PSNR
doi:10.1631/jzus.2006.AS0089 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

AVS-M is the latest video coding standard established by China for mobile application. Its framework is similar to that of H.264 and has comparative performance with lower complexity. Like other video coding standards (such as MPEG-*x*, H.26L, etc.), a great deal of computations of AVS-M encoder concentrate on motion search, transform, quantization, inverse quantization and inverse transform. So how to reduce the complexity of these modules is very important.

AVS-M encoder performs motion estimation of the variable block sizes such as 16×16, 16×8, 8×16 and 8×8 blocks for every 16×16 macro-block and 8×4, 4×8 and 4×4 blocks for each 8×8 block. In AVS-M, transform is based on 4×4 block. It adopts rate distortion optimization as rule for choosing optimal block type. So for every block type, the encoder computes the codeword length and variance between original block and reconstructed block, which includes implementing transform, quantization for residual block and then inverse quantization and inverse transform to obtain reconstructed block. So computations for them increase greatly.

In fact, there are many residual blocks which become zero after transform and quantization in low bit rate video coding with higher QP (Zhou *et al*., 1998; Pao and Sun, 1999; Wang *et al*., 2004; Kim *et al*., 2004). So if we can find out whether a residual block becomes an all-zero block before actual transform and quantization, their computation and corresponding inverse quantization, inverse transform and block reconstruction can be omitted together. This paper proposes a new all-zero block determination rule aimed at reducing computation complexity.

The rest of this paper is organized as follows. In Section 2, a new all-zero block determination rule is presented. The validity of this rule is proved by simulation in Section 3. Finally, Section 4 concludes this paper.

PROPOSED ALL-ZERO BLOCK DETERMINATION RULE

The transform in AVS-M is based on 4×4 block,

* Project (No. 05R214207) supported by the Sustentation Fund Plan for Post Doctor of Shanghai, China

and adopts low complexity 16-bit 4×4 integer DCT transform (AVS Workgroup, 2005; Zheng *et al.*, 2004), transform matrix is simple and its elements are no more than 3 integers. During the encoding process, the residual block obtained in terms of the original block and prediction block that located in the reference frame by motion vector (inter prediction) or filled by boundary pixels of spatial neighboring block in the current frame (intra prediction), is divided into 4×4 blocks to implement transform and quantization. The transform procedure is done as shown below:

$$\boldsymbol{Y}=\boldsymbol{C}_{\mathrm{f}}\boldsymbol{X}\boldsymbol{C}_{\mathrm{f}}^{\mathrm{T}}\otimes\boldsymbol{E}_{\mathrm{f}}, \qquad (1)$$

here

$$\boldsymbol{C}_{\mathrm{f}}=\begin{bmatrix}2&2&2&2\\3&1&-1&-3\\2&-2&-2&2\\1&-3&3&-1\end{bmatrix}$$

is transform matrix and

$$\boldsymbol{E}_{\mathrm{f}}=\begin{bmatrix}a^2&ab&a^2&ab\\ab&b^2&ab&b^2\\a^2&ab&a^2&ab\\ab&b^2&ab&b^2\end{bmatrix},$$

is scale matrix.

In Eq.(1), $\boldsymbol{C}_{\mathrm{f}}\boldsymbol{X}\boldsymbol{C}_{\mathrm{f}}^{\mathrm{T}}$ is the core of the integer DCT transform, $\boldsymbol{E}_{\mathrm{f}}$ denotes scale matrix, operator ⊗ means coefficient after transform is multiplied by scale factor in the same position of $\boldsymbol{E}_{\mathrm{f}}$.

In practice, scalability combines with quantization, so transform is just to implement $\boldsymbol{C}_{\mathrm{f}}\boldsymbol{X}\boldsymbol{C}_{\mathrm{f}}^{\mathrm{T}}$.

If the element in residual 4×4 block is denoted as $res(m,n)$, $0\le m$, $n<4$, the result after transform is denoted as $trans(i,j)$, $0\le i$, $j<4$, then we have

$$trans(i,j)=\sum_{m=0}^{3}\sum_{n=0}^{3}a(m,n)\cdot res(m,n), \qquad (2)$$

here $a(m,n)$, $0\le m$, $n<4$ is the coefficient of the transform matrix and listed in Table 1.

Based on Eq.(2) and Table 1, we can get

$$\begin{aligned}\left|trans(i,j)\right| &\le \sum_{m=0}^{3}\sum_{n=0}^{3}\left|a(m,n)\right|\cdot\left|res(m,n)\right| \\ &\le 9\cdot\sum_{m=0}^{3}\sum_{n=0}^{3}\left|res(m,n)\right| \\ &= 9SAD_{4\times4}.\end{aligned} \qquad (3)$$

Table 1 Coefficient in Eq.(2)

m	*n*	*a*(*m*,*n*)
0	0	{{4,4,4,4},{4,4,4,4},{4,4,4,4},{4,4,4,4}}
0	1	{{6,2,−2,−6},{6,2,−2,−6},{6,2,−2,−6},{6,2,−2,−6}}
0	2	{{4,−4,−4,4},{4,−4,−4,4},{4,−4,−4,4},{4,−4,−4,4}}
0	3	{{2,−6,6,−2},{2,−6,6,−2},{2,−6,6,−2},{2,−6,6,−2}}
1	0	{{6,6,6,6},{2,2,2,2},{−2,−2,−2,−2},{−6,−6,−6,−6}}
1	1	{{9,3,−3,−9},{3,1,−1,−3},{−3,−1,1,3},{−9,−3,3,9}}
1	2	{{6,−6,−6,6},{2,−2,−2,2},{−2,2,2,−2},{−6,6,6,−6}}
1	3	{{3,−9,9,−3},{1,−3,3,−1},{−1,3,−3,1},{−3,9,−9,3}}
2	0	{{4,4,4,4},{−4,−4,−4,−4},{−4,−4,−4,−4},{4,4,4,4}}
2	1	{{6,2,−2,−6},{−6,−2,2,6},{−6,−2,2,6},{6,2,−2,−6}}
2	2	{{4,−4,−4,4},{−4,4,4,−4},{−4,4,4,−4},{4,−4,−4,4}}
2	3	{{2,−6,6,−2},{−2,6,−6,2},{−2,6,−6,2},{2,−6,6,−2}}
3	0	{{2,2,2,2},{−6,−6,−6,−6},{6,6,6,6},{−2,−2,−2,−2}}
3	1	{{3,1,−1,−3},{−9,−3,3,9},{9,3,−3,−9},{−3,−1,1,3}}
3	2	{{2,−2,−2,2},{−6,6,6,−6},{6,−6,−6,6},{−2,2,2,−2}}
3	3	{{1,−3,3,−1},{−3,9,−9,3},{3,−9,9,−3},{−1,3,−3,1}}

Here $SAD_{4\times4}$ is the sum of absolute difference of 4×4 block and can be obtained from the procedure of motion estimation or intra prediction.

Based on the principle of DCT transform, energy concentrates to the upper left corner of the matrix (e.g., $trans(0,0)$) after transform. From the first row of Table 1, we have

$$\begin{aligned}\left|trans(0,0)\right| &= 4\cdot\left|\sum_{m=0}^{3}\sum_{n=0}^{3}res(m,n)\right| \\ &\le 4\cdot\sum_{m=0}^{3}\sum_{n=0}^{3}\left|res(m,n)\right| = 4SAD_{4\times4}.\end{aligned}$$

$res(m,n)$ ($0\le m$, $n<4$) is the mean distribution in the range of (−256, 255) from the view of statistics. So for almost all $trans(i,j)$, $0\le i$, $j<4$, we can use mean of coefficients listed in Table 1 as the factor of $SAD_{4\times4}$. That is,

$$\begin{aligned}\left|trans(i,j)\right| &\le \frac{1}{16}\cdot\sum_{m=0}^{3}\sum_{n=0}^{3}\left|a(m,n)\right|\cdot\sum_{m=0}^{3}\sum_{n=0}^{3}\left|res(m,n)\right| \\ &= 4SAD_{4\times4}.\end{aligned} \qquad (4)$$

This condition will be proved to be appropriate by simulation result in the next section.

In AVS-M, the transform block is scaled before quantization. Corresponding scale table is

$$Scale(4,4)=\begin{bmatrix}32768 & 26214 & 32768 & 26214\\ 26214 & 20972 & 26214 & 20972\\ 32768 & 26214 & 32768 & 26214\\ 26214 & 20972 & 26214 & 20972\end{bmatrix}. \quad (5)$$

Quantization procedure is implemented with quantization table, whose elements are decided by QP. In AVS-M, the range of QP is [0, 63]. For low bit rate application, the range of QP is [25, 63] in general. For this range, the relation between element in quantization table and QP is,

$$Q\_TAB(qp)\approx round(2^{15-qp/8}). \quad (6)$$

Quantization procedure is implemented by operation denoted by the following equation

$$\begin{aligned}coef(i,j)=\{&[|trans(i,j)|\times Scale(i,j)+2^{14}]/2^{15}\\ &\times Q\_TAB(qp)+qp\_const\}/2^{19},\end{aligned} \quad (7)$$

here $coef(i,j)$ denotes the result after quantization, qp_const is a constant. For example, inter prediction is considered, so qp_const equals $2^{19}/6$.

Based on Eq.(6), we can have

$$\begin{aligned}coef(i,j)&\approx\{\{[|trans(i,j)|\times Scale(i,j)+2^{14}]/2^{15}\}\\ &\quad\times(2^{15}/2^{qp/8})+2^{19}/6\}/2^{19}\\ &=|trans(i,j)|\times Scale(i,j)/2^{19+(qp/8)}\\ &\quad+1/2^{5+(qp/8)}+1/6.\end{aligned} \quad (8)$$

For the range of QP mentioned above, we know $2^{5+(qp/8)}>>1$.

So Eq.(8) can be approximated as

$$coef(i,j)\approx\frac{|trans(i,j)|\times Scale(i,j)}{2^{19+(qp/8)}}+\frac{1}{6}. \quad (9)$$

Based on Eq.(5), we have

$$Scale(i,j)\le 2^{15}. \quad (10)$$

So Eq.(10) can be written as

$$coef(i,j)\le\frac{|trans(i,j)|}{2^{4+(qp/8)}}+\frac{1}{6}. \quad (11)$$

If the coefficient in quantization matrix becomes zero after quantization, it should satisfy

$$coef(i,j)<1. \quad (12)$$

Based on Eqs.(11) and (12), we have

$$\frac{|trans(i,j)|}{2^{4+(qp/8)}}+\frac{1}{6}<1. \quad (13)$$

That is

$$|trans(i,j)|<\frac{5}{6}\times 2^{4+(qp/8)}. \quad (14)$$

Based on Eq.(4), if

$$|trans(i,j)|\le 4SAD_{4\times4}<\frac{5}{6}\times 2^{4+(qp/8)}, \quad (15)$$

then

$$SAD_{4\times4}<\frac{5}{6}\times 2^{2+(qp/8)}. \quad (16)$$

That is to say, for a 4×4 residual block, if its $SAD_{4\times4}$ satisfies Eq.(16), corresponding quantization coefficient matrix must have all-zero elements.

Under this condition, all operation including transform, quantization, inverse quantization, inverse transform and reconstruction of this block for choice of block type and encoding can be omitted.

SIMULATION RESULT

In this section, AVS-M reference code, Version 3.1, is used for simulation. Five typical benchmark sequences with different degree of detail and motion are chosen for the test. They are Mother and Daughter, News, Foreman, Container and Salesman. The format of the sequence is 4:2:0 and size of CIF is 352×288. The number of frames of every sequence is set to 101, in which the first one is encoded by intra prediction (I frame), the following 100 frames are encoded by inter prediction (P frame, also can include intra prediction block chosen by rate distortion optimization cost).

To simplifying analysis, QP is fixed for every frame of sequence to avoid introducing biasing factors. Considering the application of low rate, range of QP is from 28 to 40, increment is 3.

In the simulation, the performance of AVS-M encoder adopting the proposed rule is compared with the original one.

Detecting probability

First, the validity of the proposed rule is tested. For original AVS-M encoder, the number of actual all-zero block for all block types is summed. Also, the number of all-zero block detected by the proposed rule is also summed for all block type.

Fig.1 shows the comparison chart for every test sequence. In the legend, "actual" denotes total number of actual all-zero block; "detected" denotes total number of detected all-zero block. Here we need to say, the result is normalized.

As shown in Fig.1, the proposed rule can find out most all-zero blocks for all test sequences. The actual number of all-zero blocks increases with increasing *qp*; detection rate (detected number of all-zero blocks vs actual number of all-zero blocks) increases with increasing *qp* too.

For example, for Foreman, all-zero block ratio is 80.42% if *qp* equals 28, and up to 86.96% for 31, 91.72% for 34, 94.71% for 37 and 96.63% for 40. The detection rate is 61.15% if *qp* equals 28, and is up to 71.24% for 31, 80.51% for 34, 88.20% for 37 and 92.55% for 40.

That's to say, false acceptation rate, which means ratio of non-zero block classified as all-zero block is very small for the proposed rule. For example, for Container and *qp* equal to 31, the number of total blocks detected is about 5368800 and that of actual all-zero block is 4669103 for 100 frames. By the proposed rule, there are 3326382 all-zero blocks that are classified correctly and no more than 100 blocks with non-zero elements are classified as all-zero blocks so the false acceptation rate is very small. Similar result could be obtained for other QP and test sequences. From all our simulation, the false acceptation rate did not exceed 0.1%.

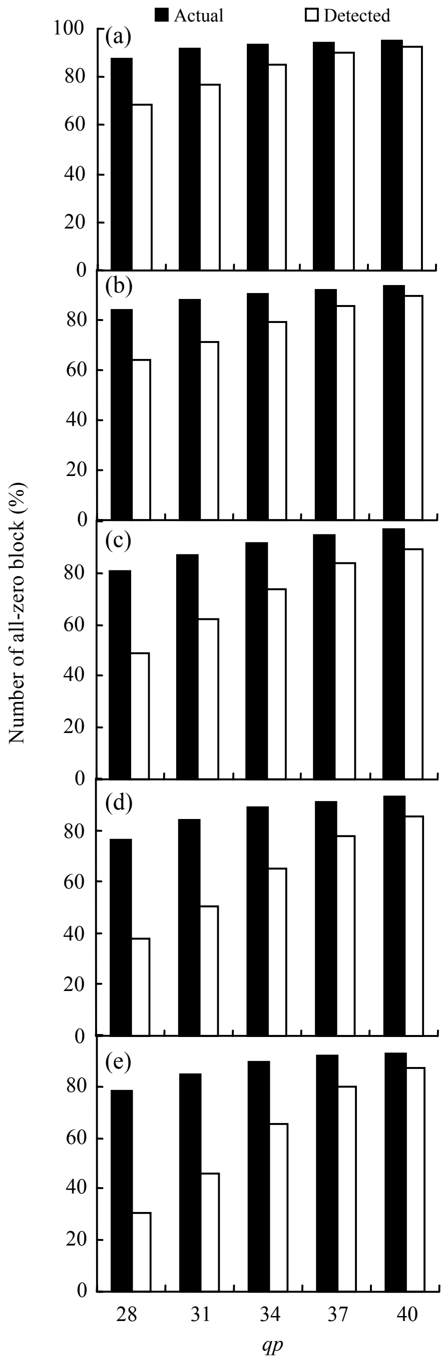

Fig.1 Number of actual and detected all-zero block. (a) Mother and Daughter; (b) News; (c) Foreman; (d) Container; (e) Salesman

Reduction of computation complexity

By pre-detecting all-zero block before transform and quantization using the proposed rule, some transform, quantization, inverse transform, inverse quantization and block reconstruction are omitted and computation complexity of the AVS-M encoder reduces. Here we only consider using the proposed rule for choice of block type in rate distortion process. Fig.2 shows the percent of saved computation, related to rate distortion computation of original AVS-M encoder, for every test sequence and *qp*.

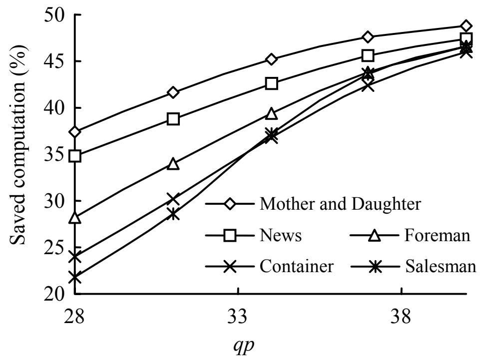

Fig.2 Saved computation by proposed rule

From the figure we can see that the saved computation nearly reaches 50% for larger *qp*. The minimum of the saved computation exceeds 20%. Also, saved computation increases with the rise of *qp*.

If the proposed rule can also be applied to motion search and intra prediction process more computation will be saved. This will be researched in our future work.

Video quality and coding rate

Effect of proposed rule on video quality, measured by PSNR and coding efficiency, measured as bit rate is reviewed. Define $PNSR_{\text{org}}$ as PSNR of coded by original AVS-M encoder and $PNSR_{\text{imp}}$ as that adopts proposed rule, the corresponding definitions for bit rate are $bitrate_{\text{org}}$ and $bitrate_{\text{imp}}$. Then

$$\Delta PSNR = \frac{PSNR_{\text{imp}} - PSNR_{\text{org}}}{PSNR_{\text{org}}} \times 100\%,$$

$$\Delta bitrate = \frac{bitrate_{\text{imp}} - bitrate_{\text{org}}}{bitrate_{\text{org}}} \times 100\%$$

are used as change of video quality and coding rate.

If we use the proposed rule only for the choice of block type in rate distortion process, there is nearly no change about PSNR and bit rate. So we extend the proposed rule to encoding process. That is, if one 4×4 block is determined as an all-zero block by the proposed rule, omit entropy coding for it. Corresponding simulation results are listed in Table 2.

The simulation results are also plotted in Fig.3. In the legend, "Original" denotes that of original AVS-M encoder; "Improved" denotes that of AVS-M encoder adopting proposed rule.

The conclusion is that by adopting the proposed rule, PSNR values reduce for all QP and sequences, but at the same time bit rates also reduce. The degree of reduction of bit rate is larger than that of PSNR, so we expect the performance of rate distortion will be a little better than the original one. At least, simulation results proved that the video quality and coding rate are not worse than the original one.

CONCLUSION

In this paper, a new rule is deduced to determine all-zero 4×4 blocks before transform and quantization based on sum of absolute difference of 4×4 block. The corresponding threshold, which varies with QP, was also carried out. By using this rule in AVS-M encoder, the times of implementing transform, quantization, inverse transform, inverse quantization and block reconstruction are reduced greatly although with a little change of video quality and bit rate. This is proved by simulation of five typical benchmark video

Table 2 Average PSNR and bit rate

Sequence	*qp*=28		*qp*=31		*qp*=34		*qp*=37		*qp*=40	
	Δ*PSNR* (%)	Δ*bitrate* (%)	Δ*PSNR* (%)	Δ*bitrate* (%)	Δ*PSNR* (%)	Δ*bitrate* (%)	Δ*PSNR* (%)	Δ*bitrate* (%)	Δ*PSNR* (%)	Δ*bitrate* (%)
Mother and Daughter	−0.518605	−5.609060	−0.539938	−4.750051	−0.553127	−4.031359	−0.555442	−3.012864	−0.529566	−2.997681
News	−0.373134	−2.692328	−0.316504	−3.061897	−0.382298	−2.707511	−0.387319	−2.967813	−0.371251	−2.455558
Foreman	−0.658588	−5.7179282	−0.659676	−5.8473134	−0.641101	−5.582914	−0.707891	−5.2181194	−0.722277	−4.969809
Container	−0.450814	−6.492766	−0.405249	−6.5101768	−0.318344	−3.796428	−0.259002	−0.0881173	−0.259002	−2.0881173
Salesman	−0.402449	−3.939636	−0.31344	−4.0623778	−0.299837	−2.907131	−0.316885	−3.312384	−0.366100	−2.213370

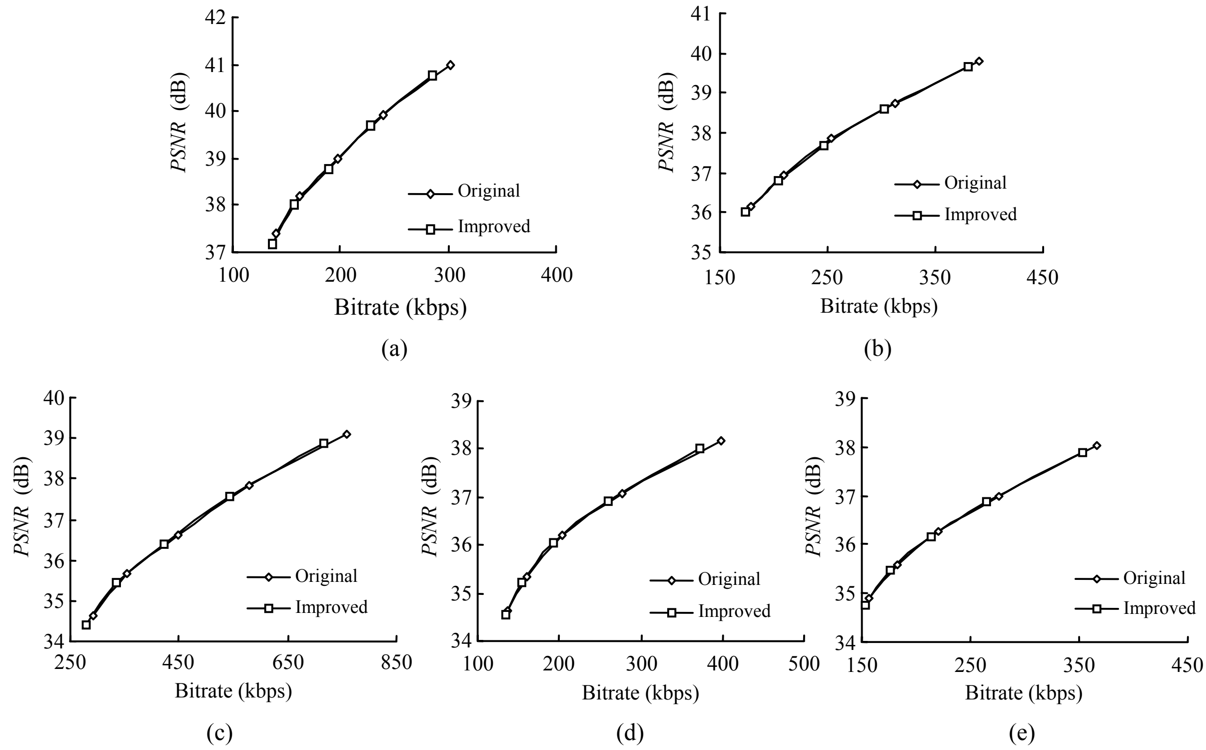

Fig.3 Performance comparison between original AVS-M Encoder and that adopting proposed rule. (a) Mother and Daughter; (b) News; (c) Foreman; (d) Container; (e) Salesman

sequences. For simplification, this paper considers luma block and inter prediction only and tests with AVS-M encoder. The rule may also be used for chroma block and intra prediction to reduce computation time by omitting more all-zero blocks. This is the work in the future.

References

AVS Workgroup, 2005. Information Technology—Advanced Coding of Audio and Video—Part 7: Mobile Video.

Kim, G.Y., Mon, Y.H., Kim, J.H., 2004. An early detection of all-zero DCT blocks in H.264. *Proc. IEEE Int. Conf. Image Processing*, p.453-456.

Pao, I.M., Sun, M.T., 1999. Modeling DCT coefficients for fast video coding. *IEEE Trans. Circuit Syst. Video Technol.*, **9**(4):608-616. [doi:10.1109/76.767126]

Wang, Y., Zhou, Y., Yang, H., 2004. Early detection method of all-zero integer transform coefficients. *IEEE Trans. Consumer Electron.*, **50**(3):923-928. [doi:10.1109/TCE.2004.1341701]

Zheng, Y., Gao, W., Zhao, D.B., Ma, S.W., Lu, H.Q., 2004. A Method for Low Complexity 16 Bit 4×4 Integer Transform and Quantization. AVS_M1331.

Zhou, X., Yu, Z., Yu, S., 1998. Method for detecting all-zero DCT coefficients ahead of discrete cosine transform and quantization. *Electro. Lett.*, **34**(19):1839-1840. [doi:10.1049/el:19981308]

Welcome visiting our journal website: ***http://www.zju.edu.cn/jzus***

Welcome contributions & subscription from all over the world

The editor would welcome your view or comments on any item in the journal, or related matters

Please write to: Helen Zhang, Managing Editor of *JZUS*

E-mail: **jzus@zju.edu.cn** Tel/Fax: 86-571-87952276/87952331