Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Video coding using geometry based block partitioning and reordering discrete cosine transform*

Yi-xiong ZHANG[†1], Jiang-hong SHI[1], Wei-dong WANG[2]
([1]*Department of Communication Engineering, Xiamen University, Xiamen 361005, China*)
([2]*Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)
[†]E-mail: zyx@xmu.edu.cn

Received July 27, 2011; Revision accepted Nov. 28, 2011; Crosschecked Dec. 8, 2011

Abstract: Geometry based block partitioning (GBP) has been shown to achieve better performance than the tree structure based block partitioning (TSBP) of H.264. However, the residual blocks of GBP mode after motion compensation still present some non-vertical/non-horizontal orientations, and the conventional discrete cosine transform (DCT) may generate many high-frequency coefficients. To solve this problem, in this paper we propose a video coding approach by using GBP and reordering DCT (RDCT) techniques. In the proposed approach, GBP is first applied to partition the macroblocks. Then, before performing DCT, a reordering operation is used to adjust the pixel positions of the residual macroblocks based on the partition information. In this way, the partition information of GBP can be used to represent the reordering information of RDCT, and the bitrate can be reduced. Experimental results show that, compared to H.264/AVC, the proposed method achieves on average 6.38% and 5.69% bitrate reductions at low and high bitrates, respectively.

Key words: Geometry based block partitioning, Reordering DCT, Motion compensation, Video coding
doi:10.1631/jzus.C1100218 **Document code:** A **CLC number:** TN919.8

1 Introduction

In video coding, motion compensation is used to reduce the temporal redundancies between frames in video sequences. The latest video coding standard H.264/AVC uses a tree-structure based block partitioning (TSBP) method (Schwarz and Wiegand, 2001) for motion compensation. In TSBP, the macroblock (MB) can be divided hierarchically into rectangular or square blocks of a minimum of 4×4 pixels. However, the shape of the square or rectangular block does not normally correspond to the contour shape of a moving object. In the case that various motions coexist in one MB, smaller blocks tend to accumulate around the motion boundary and will lead to unnecessary overhead for the motion vector (MV).

To reduce the overhead caused by small blocks, one approach is to create arbitrarily-shaped blocks with a single MV by merging several neighboring blocks or sub-blocks based on the similarity of MVs (Marpe *et al.*, 2010; Mathew and Taubman, 2010). Alternatively, researchers have proposed geometry based block partitioning (GBP) to tackle this problem (Kondo and Sasai, 2005; Hung *et al.*, 2006; Divorra Escoda *et al.*, 2007; Muhit *et al.*, 2009). GBP attempts to divide the blocks into two regions, using an arbitrary line segment influenced by the motion boundaries in the scene. For the blocks with motion discontinuities, GBP can obtain better motion compensation accuracy than TSBP. To approximate the contour of moving objects more accurately, we proposed a motion compensation technique using polyline based block partitioning (PLBP) (Zhang *et al.*, 2009). In PLBP, the polyline shape can correspond as closely as

* Project supported by the National Natural Science Foundation of China (No. 61102135), the Fundamental Research Funds for the Central Universities, China (No. 2010121063), and the Science and Technique Commission Foundation of Fujian Province, China (No. 2011H6028)

possible to the outline shape of the moving object, and motion compensation accuracy can be improved. However, the coding efficiency improvement is shadowed by the high bitrate cost of the polyline, and the computational complexity of searching for the best partition polyline is significantly higher than that of GBP.

In inter-frame coding of most video compression schemes, two-dimensional discrete cosine transform (2D-DCT) is performed after motion compensation. A serious drawback of 2D-DCT is that it is ill-suited to approximate image features with an arbitrary orientation that is neither horizontal nor vertical. In these cases, it results in large-magnitude high-frequency coefficients. From the view point of transform size, a larger-size transform can provide better energy compaction and better preservation of details in a quantized signal. Therefore, adaptive block-size transform (ABT) was adopted in both H.264 Fidelity Range Extensions (FRExt) (Wien, 2003) and High Efficient Video Coding (HEVC) (Wiegand *et al.*, 2010). Moreover, by varying the position of the transform block and its size, better coding efficiency can be achieved using the spatially varying transform (SVT) algorithm (Zhang *et al.*, 2011).

To improve the compression efficiency of 2D-DCT, another effective way is to incorporate directional information into the transform. In the literature, some directional transform methods have been proposed. In image coding, shape adaptive DCT (Zeng and Fu, 2006; 2008; Fu and Zeng, 2007) and lifting based directional DCT (Xu *et al.*, 2007) were used to adapt the transform to the signal features. In video coding, by utilizing the directional information of intra prediction, the coding efficiency of 2D-DCT can be improved by mode dependent directional transform (MDDT) (Ye and Karczewicz, 2007; Budagavi and Zhou, 2010) and rotational transform (ROT) (Fernandes, 2010; Han *et al.*, 2010). In these methods, new transforms should be designed and are suitable only for image or intra frame coding. Robert *et al.* (2008) proposed a block oriented transform based on block pre-processing for intra and inter frame coding. However, the method sends directional information for each predicted sub-block, and the performances of many test sequences are instead worse than those using H.264. To reduce the bitrate of the directional information, we proposed a reordering DCT (RDCT) method (Zhang *et al.*, 2010). RDCT is applied only to the MBs instead of sub-blocks, and the performance improvement is obtained from low to high bitrates.

Compared to TSBP, the coding efficiency is improved by GBP. However, in our experiments, it is observed that the residual blocks of GBP mode still show some orientations following the contours of the moving objects. The objective of the GBP partition line is to approximate the object edge in the coding block. When applying directional transform to the GBP residual block, it is reasonable that the direction of the partition line can be used to represent the directional information of the transform. This is the concept of our approach in this paper. In the proposed coding approach, GBP is combined with RDCT, and the reordering direction of RDCT is represented by the direction of the GBP partition line. In this way, the coded bits for representing the directional information of RDCT can be saved, and the performance of GBP can be improved.

2 Geometry based block partitioning for motion compensation

In general, actual video sequences contain a lot of moving objects with curved edges. An MB with a fixed size (i.e., 16×16) may contain two or more regions with different motions. To code an MB containing two different regions separated by a curved contour, an ideal way is to code the contour shape precisely and to code the predicted error of each region (Fig. 1). However, this is difficult to implement. Moreover, the distortion reduction may be offset by the bitrate cost of the contour shape. To tackle this problem, a compromised partitioning approach, geometry based block partitioning (GBP), was proposed (Kondo and Sasai, 2005; Hung *et al.*, 2006; Divorra Escoda *et al.*, 2007). A comparison between TSBP and GBP is shown in Fig. 1. In GBP, a straight line segment is used to divide an MB into two regions, as shown in the right section of Fig. 1. The angle and the location of the partition line *ab* can be adjusted to capture the local geometrical features of predictable regions. To represent the object's contour accurately, about 10 sub-blocks should be generated for TSBP. Thus, the MV number of TSBP is much larger than that of GBP (10 versus 2 in the example of Fig. 1). Moreover, the residual data of TSBP after motion compensation may be greater than that of GBP, as the rectangular boundary of TSBP cannot represent the contour shape as accurately as the arbitrary line segment of GBP.

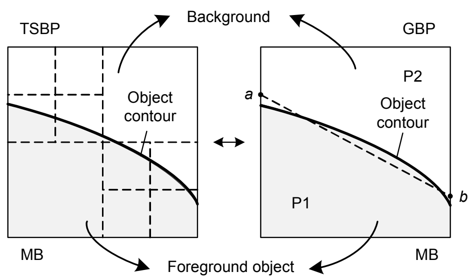

Fig. 1 Comparison between tree structure based block partitioning (TSBP) and geometry based block partitioning (GBP)

In GBP, the partition line segment ab splits an MB into two partitions, P1 and P2. A cross product based rule can be defined to calculate two masking matrices MASKP1 and MASKP2, which specify the shapes of P1 and P2, respectively. Motion compensation is performed for each partition. When the partition line passes through a pixel, a blending rule is used to combine the prediction of both partitions. This can minimize the aliasing effect. The predicted block of the current MB is defined as

$$\begin{aligned}\mathrm{refP}(x) = &\ \mathrm{refP1}(x)\cdot\mathrm{MASKP1}(x)\cdot w \\ &+ \mathrm{refP2}(x)\cdot\mathrm{MASKP2}(x)\cdot(1-w),\end{aligned} \tag{1}$$

where x denotes the pixel in the MB, refP1 and refP2 are the predicted blocks of P1 and P2, respectively, and w is a weighted factor, defined as

$$w=\begin{cases}0.5, & \text{if } (\mathrm{MASKP1}(x)=1\,\&\&\,\mathrm{MASKP2}(x)=1),\\ 1, & \text{if } (\mathrm{MASKP1}(x)=1\,\&\&\,\mathrm{MASKP2}(x)=0),\\ 0, & \text{else.}\end{cases} \tag{2}$$

In our implementation, GBP is added to the H.264 P-picture coding modes. A list of coding modes is shown in Table 1. For GBP mode, the information of the partition line should be sent to the decoder side. Kondo and Sasai (2005) used a prediction plus fixed-length coding method to code the coordinates of the partition points, while Hung *et al.* (2006) and Divorra Escoda *et al.* (2007) used angle and distance parameters to code the line information. These two methods are equivalent in efficiency. In this study, we adopt the former approach. Six submodes are defined for the GBP mode, depending on the location of the two partition points a and b (Table 1). For example, for submode 0, 'top_bot' indicates that a locates on the top MB border and b on the bottom border. With submode coding, only the horizontal or vertical coordinate need to be coded for each partition point.

Table 1 List of coding modes in GBP experiments

No.	Macroblock mode	P8×8 submode	GBP mode	
			Submode	Code
0	Skip	P8×8	top_bot	00
1	P16×16	P8×4	left_right	01
2	P16×8	P4×8	top_right	100
3	P8×16	P4×4	top_left	101
4	P8×8		bot_right	110
5	GBP		bot_left	111
6	Intra			

In general, the partition line can pass through any position in the MB, but this will bring two problems. First, the computational complexity required to select the most suitable partition line will increase significantly. Second, a large number of bits are required to describe the partition lines. To solve these problems, in this study we impose a limitation that all the partition points exist at an interval of two pixels on the MB boundary. Thus, three bits are needed to code the horizontal or vertical coordinate of each partition point. Coordinate prediction is performed under the assumption of line continuity. For example, suppose that the current MB has a partition point (a_1) on its left boundary in Fig. 2. The left MB has been encoded using GBP mode and has a partition point (b_2) on its right boundary. A predicted flag (Pflag) of one bit is used to indicate whether the two points have the same position. If Pflag=1, the coordinate of a_1 is the same as that of b_2. Otherwise, three additional bits should be used to encode the vertical coordinate of a_1. Prediction is also applied for the partition points on the top boundary. For other cases, the coordinate of the partition point is encoded using three bits.

The encoding mode and the best partition line of GBP mode are determined by the rate distortion optimization described below. Fig. 3 shows an example of GBP+H.264 coding. Fig. 3a illustrates how the coding modes are selected for each MB. Fig. 3b shows the residual frame of Fig. 3a, and Fig. 3c shows the enlarged upper right part of Fig. 3b.

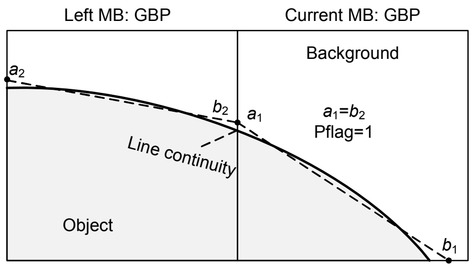

Fig. 2 Prediction of the partition point

It can be seen that the GBP mode is selected mostly by the MBs that contain two or more different motions separated by non-horizontal/non-vertical edges. This demonstrates that GBP mode can fit well to the boundaries of moving objects, such as the face and the background. We can also see that, for the MBs encoded with GBP mode, the residual blocks still present similar orientations to the textures or the contour shapes of the moving objects. One reason is that, when using the translational block matching algorithm (BMA) in motion estimation, the accuracy of motion compensation at the boundary regions is usually lower than that at the inner regions. The residual blocks may contain some non-vertical/non-horizontal edges. In these cases, many high-frequency coefficients will be generated by the conventional 2D-DCT. To tackle this problem, we propose a coding approach incorporating GBP and the reordering DCT method (Zhang *et al.*, 2010).

3 Video coding using GBP and RDCT

Until now, GBP and directional DCT have been applied separately. However, it can be observed that they have a common purpose—to exploit the directional information of video features, i.e., textures or object edges. Inspired by this observation, we propose a video coding approach using GBP and RDCT techniques. In the proposed coding approach, the direction of the GBP partition line is used to represent the reordering direction of RDCT. In this way, the bits for representing the directional information of RDCT can be saved, and the coding efficiency of GBP can be improved. In the following subsections, RDCT is first presented, and then we propose how to combine GBP with RDCT in video coding.

3.1 Structure of reordering DCT

The RDCT method consists of two stages: pixel reordering and the conventional 2D-DCT. The pixel reordering operation is a reversible transformation. The flow of video codec with RDCT is shown in Fig. 4. In the encoder, the residual block after motion compensation is first reordered based on some orientations. Then the conventional 2D-DCT is performed for the reordered block. The flow of the decoder is carried out in the opposite direction, as shown in the lower part of Fig. 4.

Let $\boldsymbol{X}$ be the residual block to be coded. $\boldsymbol{Y}$ denotes the block of transformed coefficients, and $\boldsymbol{A}$ is the transform matrix of DCT. The expression of RDCT can be defined as

$$\boldsymbol{Y} = \boldsymbol{A}F(\boldsymbol{X})\boldsymbol{A}^{\mathrm{T}}, \tag{3}$$

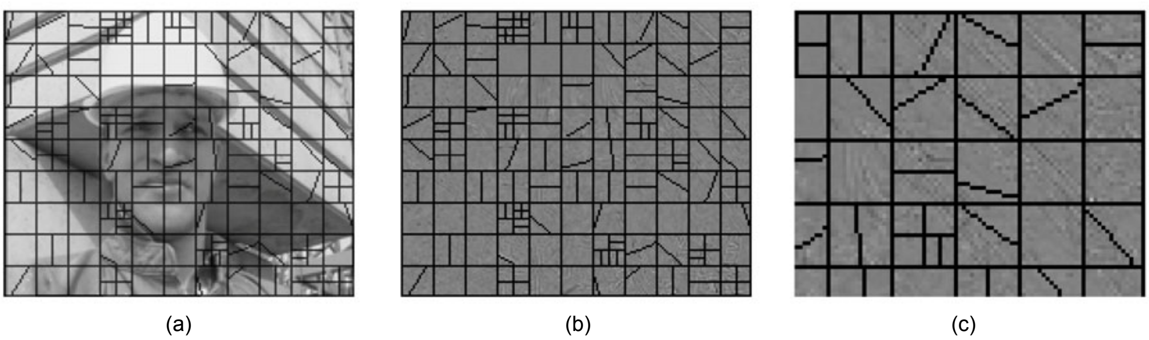

Fig. 3 The 14th frame of Foreman with geometry based block partitioning (GBP) coding mode
(a) Map of coding modes; (b) Residual frame of (a); (c) Enlarged upper right part of (b), with a scale factor of 1:2

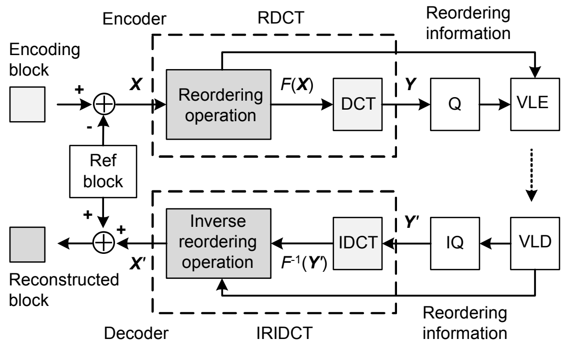

Fig. 4 Block structure of video codec with reordering DCT (RDCT)

Q: quantization; IQ: inverse quantization; VLE: variable length encoding; VLD: variable length decoding; IRIDCT: inverse reordering with inverse DCT

where $F(\cdot)$ denotes the reordering function, and $F(\boldsymbol{X})$ is the block after pixel reordering. Similarly, the inverse reordering operation with inverse DCT (IRIDCT) in the decoder can be expressed as

$$\boldsymbol{X}' = F^{-1}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}'\boldsymbol{A}), \tag{4}$$

where $F^{-1}(\cdot)$ denotes the inverse reordering transform, $\boldsymbol{Y}'$ denotes the transform coefficient block after de-quantization at the decoder side, and $\boldsymbol{X}'$ is the reconstructed residual block after IRIDCT.

$F(\cdot)$ is a reversible mapping, i.e., one-to-one mapping. For each pixel $X_{i,j}$ in block $\boldsymbol{X}$, there is only one corresponding pixel $F(X)_{k,l}=F(X_{i,j})$ in the reordered block $F(\boldsymbol{X})$, and vice versa. In other words, the reordering operation is equivalent to the coordinate transformation of the block pixels. $F(\cdot)$ can be implemented by various coordinate transformation methods, such as the look up table and circular shift. In this study we adopt the circular shift method. To perform the inverse reordering operation correctly, the reordering information (the index of the look up table, or the index of the reordering direction) should be sent to the decoder (Fig. 4).

3.2 Pixel reordering by circular shift

Circular shift comes from the idea of pixel shears (Taubman and Zakhor, 1994). Take the 8×8 block as an example. Figs. 5a and 5b show the horizontal and vertical circular shifts, respectively. The digital line l is used to represent the edge of the block. Its slant angle θ referred to the x axis indicates the reordering direction of the block. The gray squares denote the edge pixels in the block. The horizontal shift is used for $\pi/4<|\theta|\leq\pi/2$, while the vertical shift is used for $|\theta|\leq\pi/4$.

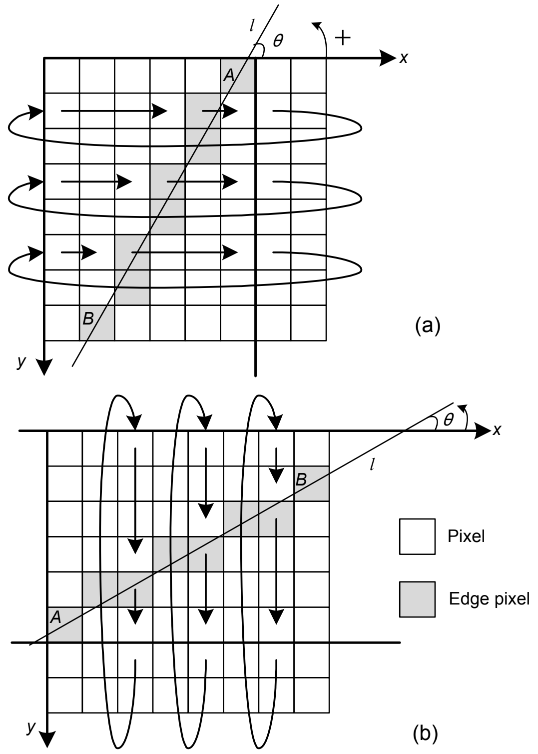

Fig. 5 Pixel reordering of the 8×8 block

(a) Horizontal shift ($\pi/4<|\theta|\leq\pi/2$); (b) Vertical shift ($|\theta|\leq\pi/4$)

Suppose the pixel coordinate of block $\boldsymbol{X}$ is $\boldsymbol{n}=(i, j)^{\mathrm{T}}$. The coordinate of the corresponding pixel in the reordered block $\boldsymbol{X}'$ is $\boldsymbol{m}=F(\boldsymbol{n})=(k, l)^{\mathrm{T}}$. The relationship between $\boldsymbol{n}$ and $\boldsymbol{m}$ can be formulated as

$$\boldsymbol{m} = (\lceil \boldsymbol{F}(\theta)\cdot\boldsymbol{n} \rceil + N) \bmod N, \tag{5}$$

$$\boldsymbol{F}(\theta) = \begin{cases} \boldsymbol{F}_{\mathrm{V}}(\theta), & |\theta| \leq \pi/4, \\ \boldsymbol{F}_{\mathrm{H}}(\theta), & \pi/4<|\theta| \leq \pi/2, \end{cases} \tag{6}$$

where N denotes the block size (N=8 in Fig. 5), mod is the module operation to implement circular shift, and $\boldsymbol{F}_{\mathrm{V}}(\theta)$ and $\boldsymbol{F}_{\mathrm{H}}(\theta)$ are defined as

$$\boldsymbol{F}_{\mathrm{V}}(\theta) = \begin{bmatrix} 1 & 0 \\ \tan\theta & 1 \end{bmatrix}, \tag{7}$$

$$\boldsymbol{F}_{\mathrm{H}}(\theta) = \begin{bmatrix} 1 & 1/\tan\theta \\ 0 & 1 \end{bmatrix}. \tag{8}$$

By circular shift, the edge pixels can be moved toward the positions along the horizontal or vertical direction. In the decoder side, the pixel coordinates can be recovered by inverse circular shift.

In our previous work (Zhang *et al.*, 2010), RDCT was incorporated into the H.264 encoder and applied to three inter-coding modes including 16×16, 16×8, and 8×16. Experimental results showed that RDCT achieved about 5% performance improvement without sending the reordering information. However, when considering the reordering information, the performance improvement reduces to 1.51% at low bitrates and 2.46% at high bitrates. These results imply that RDCT has the potential in residual coding. However, to harness the full benefits of using RDCT, it is of primary importance to reduce the bitrate of the reordering information.

3.3 Integrating GBP and RDCT into the video coder

Similar to GBP, the objective of RDCT is also to utilize the directional information of image textures or edges in the coding block. It is expected that the compression ratio of the GBP residual blocks can be improved when using RDCT.

Fig. 6 illustrates an example of how to combine GBP and RDCT techniques. Using GBP mode, the MB was partitioned by line *AB* (Fig. 6a). The slant angle of *AB* is θ. Fig. 6b shows the reordering operation of the GBP residual block, and Fig. 6c illustrates the residual block after reordering. The slant angle of *AB* is used to indicate the reordering direction of the GBP residual block. Therefore, it is not necessary to send the reordering information to the decoder side for the GBP MBs, and the bits can be saved.

The proposed technique is implemented and tested in the H.264/AVC framework. GBP and RDCT are evaluated for P frames. The coding modes of the P frame are listed in Table 1. RDCT is performed only for the MBs that have been selected as GBP mode. With RDCT, the rate distortion optimization of the GBP MB includes four steps: (1) determine the MVs of the two partitions for each candidate partition line, (2) select the best partition line, (3) select the best prediction mode for the encoding MB, and (4) choose to use either RDCT or conventional DCT for the GBP MB. Further details about each step are given below.

Step 1: The set of the candidate partition lines is defined as $\Lambda=\{l_i: 0<i<L\}$. To find the best partition line, a full search method is adopted and L is 384. When performing motion compensation for GBP mode, the line segment l_i is selected in turn from Λ. l_i divides the MB into two partitions. For each partition, the MV $m_j(l_i)$ (where the partition number $j\in\{1, 2\}$) is estimated for a reference picture using Eqs. (9) and (10):

$$I_j(m,l_i)=D_{\mathrm{v}}(s_j,r_j(m)\,|\,l_i)+\lambda_{\mathrm{v}}R(m-p_j), \quad (9)$$

$$m_j(l_i)=\arg\min_m I_j(m,l_i), \quad (10)$$

where m is the MV of partition j and $I_j(m, l_i)$ denotes the cost function for partition j. The function $D_{\mathrm{v}}(\cdot)$ denotes the sum of the error between the region to be coded in the original frame s_j and the predicted region in the reference frame $r_j(m)$, when referring to MV m. The function $R(\cdot)$ denotes the amount of bits for the MV difference $m-p$. The MV prediction value p is calculated by the median MV filter of H.264. λ_{v} is the Lagrange multiplier in the motion estimation stage.

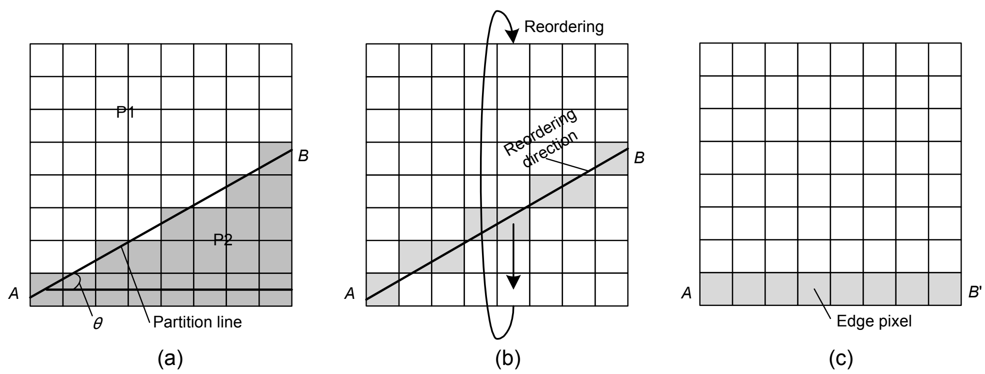

Fig. 6 Combination of geometry based block partitioning (GBP) and reordering DCT (RDCT)
(a) Coding block with GBP; (b) Residual block of (a); (c) Reordered residual block

Step 2: Next, the cost is calculated for the whole MB using the MVs $m_j(l_i)$ obtained for the two partitions, and the line l that gives the lowest cost value is selected as the partition line using

$$l = \arg\min_{l_i \in \Lambda} \left(\sum_j I_j(m_j(l_i), l_i) + \lambda_{\mathrm{v}}(R(l_i)) \right), \quad (11)$$

where $R(l_i)$ denotes the amount of bits for partition line l_i, including the bits for the residual data and the partition information.

Step 3: For each MB mode (Table 1), a cost function value J is calculated using

$$J(M) = \sum_k \{D(s_k, d_k \mid M, m_k(M)) + \lambda_m \{R((s_k - r_k) \mid M, m_k(M)) + R(M, m_k(M) - p_k)\}\}, \quad (12)$$

where k denotes the number of blocks in the MB when the coding mode is M, and λ_m is the Lagrange multiplier in the mode selection stage. d_k denotes the reconstructed block. The function $D(\cdot)$ denotes the sum of the squared error between the original block s_k and the reconstructed block d_k when the MV is $m_k(M)$. The function $R(\cdot)$ in the second term denotes the amount of bits for the residual signal s_k-r_k, and the function $R(\cdot)$ in the third term denotes the amount of bits for the MB coding mode M and the MV difference $m_k(M)-p_k$. The prediction mode giving the lowest cost J is selected as the final coding mode.

Step 4: As the error energy $D_{\mathrm{v}}(\cdot)$ may be affected by noise or brightness change, the selected partition line in GBP mode may not correspond to the object edge. In this case, the efficiency of residual coding with RDCT may become worse than that without RDCT. To prevent this, the coding cost J (Eq. (12)) of the GBP mode will be calculated twice, one with RDCT, the other with the conventional DCT. A one-bit flag (GBP_RDCT_EN) is used to indicate whether RDCT is applied. If GBP_RDCT_EN=0, the residual block will be coded using the conventional 2D-DCT; if GBP_RDCT_EN=1, the residual block will be coded using RDCT, and the direction of the GBP partition line will be used as the reordering direction. Therefore, only one bit is needed to represent the reordering information of the GBP MB.

In H.264, deblocking filters are used to reduce the blocking artifacts. To apply deblocking filters to the GBP partition edges, it is necessary to design a set of arbitrary directional filters. This non-trivial task, however, is out of the scope of this paper and will be studied in our future work. In this study, a sub-optimal approach is developed to be compatible with the deblocking filters of H.264. In the GBP MB, deblocking is not applied to the partition edge, but to the 4×4 block edges adjacent to the partition edge. The deblocking procedure includes two stages, boundary strength (BS) derivation and edge filtering. BS calculation is performed based on the 4×4 block unit. The BS value is dependent on the MVs and the nonzero residual coefficients of neighboring 4×4 blocks. Fig. 7 shows an example. Once a 4×4 block crosses the partition line, it will be divided by the partition line into two sub-partitions, a large one and a small one. In this case, the MV of the MB partition that contains the larger sub-partition will be assigned to the 4×4 block. In other cases, the MV of the 4×4 block is derived from the MB partition that contains it.

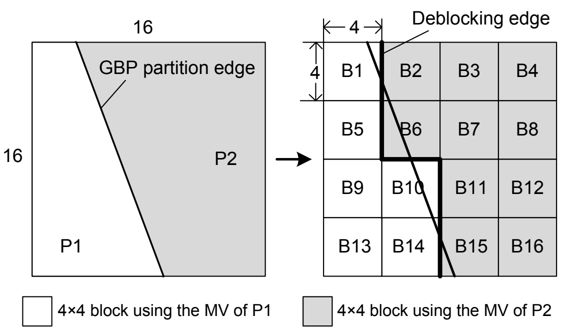

Fig. 7 Deblocking of the geometry based block partitioning (GBP) macroblock

In Fig. 7, the MV of partition P1 is assigned to the 4×4 blocks B1, B5, B9, B10, B13, and B14, while the MV of partition P2 is assigned to the other 4×4 blocks. As a result, deblocking is applied only to the 4×4 block edges adjacent to the partition line (the coarse line of the 4×4 block edge in Fig. 7). After the BS calculation is completed, the edge filtering procedure is performed as specified in H.264.

4 Results and analysis

We implemented the proposed algorithm in JM9.0 reference software. In the experiments, GBP and RDCT were applied to both the luminance and chrominance components of the residual frames. For the chrominance component, the geometric partition information, the MVs, and the reordering direction were derived from the luminance component. IPPP format was adopted and only the first frame was set as an intra coding frame. GBP and GBP+RDCT were integrated into H.264 P-picture coding and compared with the anchor experiment, respectively. To decrease the computational complexity, the unsymmetrical-cross multi-hexagon-grid search (UMHexagon) algorithm (Chen *et al.*, 2003) was used for integer pixel motion estimation. To evaluate the performances of the proposed technique at different bitrates, several standard sequences with fixed QP values ranging from 13 to 38 were tested. The experimental parameters are listed in Table 2.

The performances of the three coding schemes (H.264, GBP+H.264, GBP+RDCT+H.264) were calculated and compared using the BD-bitrate and BD-PSNR calculation method (Bjontegaard, 2001). Performance comparison between GBP+H.264 and H.264 is shown in Table 3. GBP achieves on average a 4.4% bitrate reduction at low bitrates and a 2.96% bitrate reduction at high bitrates. The gain of GBP is lower at high bitrates, because the mismatch between the regularity of the conventional DCT and the non-regular shape of the GBP partition is severer for smaller QPs. Also, it can be seen that the gains of Foreman and Carphone are larger than those of other sequences. The reason is that these two sequences have more clear foreground and background objects for which the GBP model is more suitable. For the Football sequence, the motion is much more complex so that BMA cannot work well and the least performance gain is obtained.

Table 2 Experimental parameters

Parameter	Value/Description
Image format	QCIF, CIF
Transform	QCIF: 4×4; CIF: 4×4 and 8×8
Frame rate	QCIF: 15 frames/s; CIF: 30 frames/s
Frame number	QCIF, CIF: 100 frames
Search range	QCIF: ±16 pixels; CIF: ±32 pixels
Search precision	1/4 pixel
Search algorithm	UMHexagon
Deblocking filter	On
QP	P: 38, 33, 28, 23, 18, 13 I: 37, 32, 27, 22, 17, 12

Table 3 Performance of GBP+H.264 compared to H.264

Sequence	PSNR (dB)		Rate (%)	
	Low bitrates	High bitrates	Low bitrates	High bitrates
Foreman_qcif	0.253	0.178	−6.16	−3.59
Mobile_qcif	0.156	0.104	−3.35	−1.34
Carphone_qcif	0.188	0.244	−4.12	−5.87
Football_qcif	0.044	0.043	−1.16	−0.67
Foreman_cif	0.266	0.165	−7.29	−4.19
Carphone_cif	0.214	0.134	−5.85	−3.97
Mobile_cif	0.123	0.082	−2.85	−1.10
Average	0.178	0.136	−4.40	−2.96

'Low bitrates' means that the QP of the P-picture ranges from 38 to 23; 'High bitrates' means that this QP ranges from 28 to 13

Table 4 shows the performance comparisons of the three methods. The gain of GBP+RDCT over GBP at high bitrates (2.48% on average) is higher than that at low bitrates (2.0% on average). As we know, the objective of RDCT is to re-organize the DCT coefficients. Nevertheless, many coefficients at high frequencies are quantized to zero at high QPs, and thus RDCT is less effective at lower bitrates. Table 4 also shows that the gain of the Mobile sequence is larger than those of other sequences. This shows that RDCT is more effective in processing residual data with more edge or texture features.

Table 4 Performances of GBP+RDCT+H.264 compared to GBP+H.264 and H.264

Sequence	PSNR* (dB)		Rate* (%)		PSNR** (dB)		Rate** (%)	
	Low bitrates	High bitrates	Low bitrates	High bitrates	Low bitrates	High bitrates	Low bitrates	High bitrates
Foreman_qcif	0.087	0.185	−2.10	−3.78	0.340	0.363	−8.23	−7.31
Mobile_qcif	0.155	0.253	−3.36	−3.32	0.311	0.357	−6.66	−4.63
Carphone_qcif	0.124	0.076	−2.76	−1.81	0.312	0.319	−6.82	−7.71
Football_qcif	0.036	0.100	−0.94	−1.48	0.080	0.143	−2.08	−2.15
Foreman_cif	0.050	0.108	−1.33	−2.82	0.316	0.273	−8.62	−8.95
Mobile_cif	0.101	0.147	−2.38	−2.09	0.224	0.229	−5.21	−3.09
Carphone_cif	0.044	0.068	−1.17	−2.07	0.259	0.202	−7.02	−5.99
Average	0.085	0.134	−2.00	−2.48	0.263	0.269	−6.38	−5.69

* GBP+RDCT+H.264 vs. GBP+H.264; ** GBP+RDCT+H.264 vs. H.264

The overall gains of GBP+RDCT over H.264 are shown in the right part of Table 4. The gain brought by GBP is higher at low bitrates, while the gain brought by RDCT is higher at high bitrates. As a result, the gains brought by GBP+RDCT reach the same level at both low and high bitrates. This implies that GBP and RDCT are complementary as expected. In general, the proposed GBP+RDCT technique achieves average bitrate reductions of 6.38% and 5.69% at low and high bitrates, respectively. For many cases, the bitrate reduction can reach 7%. Fig. 8 shows the rate-distortion curves for Foreman_cif and Mobile_cif sequences. GBP+RDCT exhibits stable performance improvements at different bitrates.

We further measured the percentages of the GBP mode and RDCT mode selected in the two experiments, GBP+H.264 and GBP+RDCT+H.264. The results are shown in Tables 5 and 6, respectively. The percentage of GBP increases as QP decreases. At high QPs, more high-frequency coefficients are quantized to zero and the texture information will not be well preserved. Hence, the percentage of GBP is smaller at high QPs. In Table 5, the GBP percentage at QP=13 is smaller than that at QP=18 instead. This can explain why the gain brought by GBP is relatively small at high bitrates. In contrast, in the GBP+RDCT+H.264 experiment, the percentage of GBP increases remarkably even from QP=18 to QP=13 (Table 6). This implies that the mismatch between the regularity of the conventional DCT and the non-regular shape of the GBP partition can be released by RDCT. This can also be approved by the percentage of RDCT in Table 6. Note that the MBs selected as RDCT mode are also GBP MBs. From QP=38 to QP=13, the proportion that RDCT takes up in GBP MBs increases gradually from 34% to 80%. In other words, RDCT is selected by most GBP MBs at high bitrates.

Comparison of the results in Tables 5 and 6 shows that the percentage of GBP with RDCT is much larger than that without RDCT; i.e., GBP benefits from the advantage of RDCT. Specifically, for some sequences such as Foreman and Mobile, the GBP percentages in Table 6 exceed 50% at higher bitrates. From the viewpoint of video resolution, low resolution videos have more MBs with geometric information. Thus, the percentages of GBP and RDCT of QCIF sequences are generally larger than those of CIF sequences.

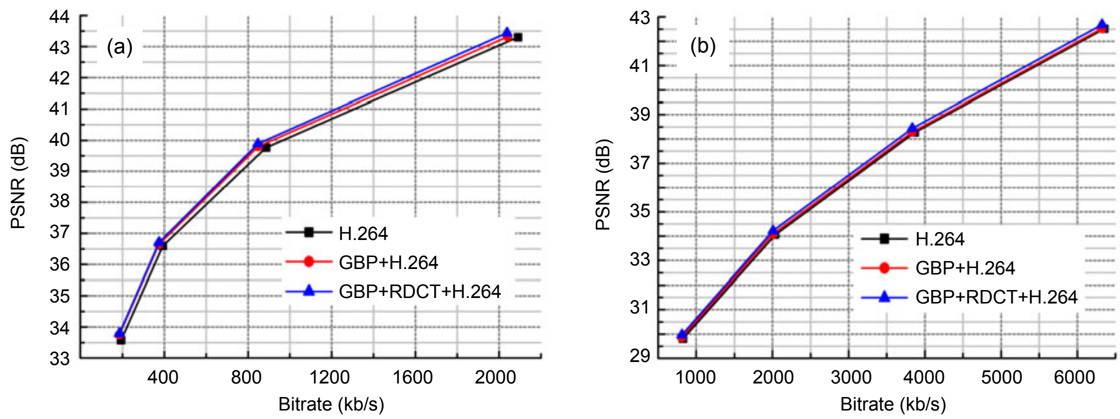

Fig. 8 Rate-distortion curves for Foreman_cif (a) and Mobile_cif (b)

Table 5 Percentage of GBP mode in GBP+H.264 experiments

QP	Percentage (%)							Average (%)
	Foreman_ qcif	Carphone_ qif	Mobile_ qcif	FootBall_ qcif	Foreman_ cif	Carphone_ cif	Mobile_ cif	
38	5.4	2.3	12.1	5.0	2.7	1.6	10.0	5.6
33	11.2	8.1	25.5	7.6	6.6	5.2	23.2	12.5
28	21.6	15.7	33.3	11.8	13.8	11.0	28.0	19.3
23	29.4	21.7	33.3	14.4	22.6	16.6	26.6	23.5
18	30.8	23.1	30.4	15.5	27.5	22.1	23.9	24.7
13	28.7	23.2	27.4	15.0	26.6	21.4	21.6	23.4

Figs. 9 and 10 show examples of coding results for Foreman_qcif and Carphone_qcif sequences. The MB partition modes in the GBP+RDCT+H.264 experiment are shown in (a), where the GBP MBs with white boundary lines are coded using RDCT. Figs. 9d and 10d show the residual frame of Figs. 9a and 10a, respectively. The images reconstructed using the proposed approach are shown in (b) and (e), while the images reconstructed using H.264 are shown in (c) and (f). The residual frames still present similar orientations to the geometric features of textures or object edges (e.g., the background, the shoulder, and the face edge). The GBP mode is selected mainly in these areas. Furthermore, RDCT is selected most probably by the MBs whose partition lines correspond to the geometric orientations. Comparison of (e) and (f) shows that the subjective quality is improved using the proposed approach.

Table 6 Percentage of GBP and RDCT modes in GBP+RDCT+H.264 experiments

QP	Mode	Percentage (%)							Average (%)
		Foreman_ qcif	Carphone_ qif	Mobile_ qcif	Football_ qcif	Foreman_ cif	Carphone_ cif	Mobile_ cif	
38	GBP	6.3	2.9	18.0	7.6	2.8	1.7	14.3	7.7
	RDCT	1.1	0.5	7.0	3.6	0.4	0.3	5.3	2.6
33	GBP	15.1	10.2	41.7	11.1	7.6	6.1	36.6	18.3
	RDCT	5.0	3.2	24.7	6.2	1.8	1.4	21.6	9.1
28	GBP	31.1	23.1	62.7	18.7	17.8	14.0	51.5	31.3
	RDCT	14.9	10.5	46.9	10.8	6.7	4.9	36.7	18.8
23	GBP	45.3	32.3	68.7	29.4	33.1	22.9	55.2	41.0
	RDCT	28.9	19.8	55.9	19.7	18.3	10.7	40.1	27.6
18	GBP	54.8	39.5	70.6	40.1	50.9	38.3	56.4	50.1
	RDCT	41.8	27.7	59.2	31.3	36.2	23.3	40.9	37.2
13	GBP	60.4	50.4	72.3	45.6	62.2	50.3	57.8	57.0
	RDCT	49.6	38.6	62.8	37.6	48.5	38.1	43.0	45.5

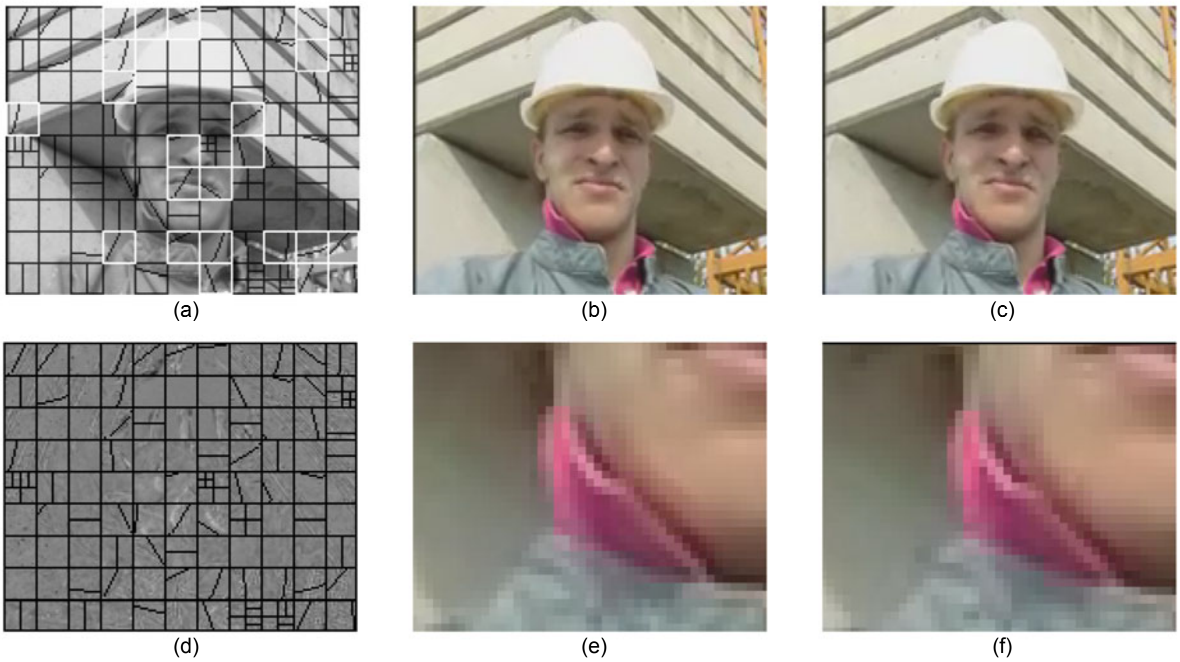

Fig. 9 The 19th frame of Foreman_qcif (QP=28)
(a) The map of coding modes in GBP+RDCT+H.264; (b) The frame reconstructed using GBP+RDCT+H.264; (c) The frame reconstructed using H.264; (d) The residual frame of (a); (e) The enlarged region of (b), with a scale factor of 1:4; (f) The enlarged region of (c), with a scale factor of 1:4

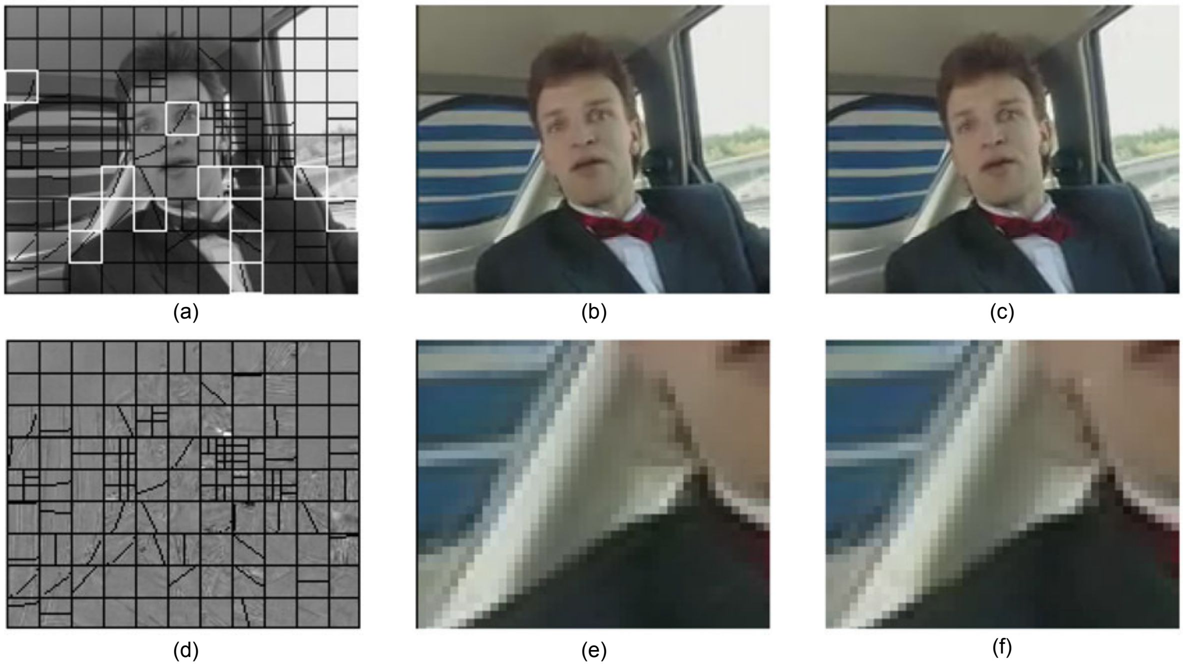

Fig. 10 The 29th frame of Carphone_qcif (QP=28)

(a) The map of coding modes in GBP+RDCT+H.264; (b) The frame reconstructed using GBP+RDCT+H.264; (c) The frame reconstructed using H.264; (d) The residual frame of (a); (e) The enlarged region of (b), with a scale factor of 1:4; (f) The enlarged region of (c), with a scale factor of 1:4

5 Conclusions

We have proposed a novel algorithm called GBP+RDCT to improve the coding efficiency of the video coder. The main concept of RDCT is to vary the pixel positions of the transform block, and to 'rotate' the block edge to the horizontal or vertical direction. The principle of GBP+RDCT is to utilize the partition information of GBP as the reordering information of RDCT. The proposed algorithm has been studied and implemented in the H.264/AVC framework. Compared to H.264, the proposed GBP+RDCT technique achieves on average 6.38% and 5.69% bitrate reductions at low and high bitrates, respectively. Specifically, for many cases, the bitrate reduction can reach 7%. The subjective quality is also improved using the proposed approach.

The decoding complexity of GBP+RDCT is slightly higher than that of H.264 because of the inverse reordering operation. However, the encoding complexity of GBP+RDCT is relatively high due to the need to search for the best partition line for the GBP mode. Fast searching algorithms (Hung *et al.*, 2006; Muhit *et al.*, 2009) can be employed to reduce the complexity of partition line selection. The proposed technique is also well-suited for the next generation video coding scheme, High Efficient Video Coding (HEVC), which combines super-macroblock coding structure and the larger-size transforms.

References

Bjontegaard, G., 2001. Calculation of Average PSNR Differences Between RD-Curves. VCEG-M33. 13th JVT Meeting.

Budagavi, M., Zhou, M., 2010. Orthogonal MDDT and Mode Dependent DCT. VCEG-AM20. 36th JVT Meeting.

Chen, Z.B., Xu, J.F., Zhou, P., He, Y., 2003. Hybrid Unsymmetrical-Cross Multi-Hexagon-Grid Search Strategy for Integer Pel Motion Estimation in H.264. Picture Coding Symp., p.17-21.

Divorra Escoda, O., Yin, P., Dai, C., Li, X., 2007. Geometry-Adaptive Block Partitioning for Video Coding. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, p.657-660. [doi:10.1109/ICASSP.2007.365993]

Fernandes, F.C., 2010. Low-Complexity Rotational Transform. JCT-VC-C096. 3rd JCT-VT Meeting.

Fu, J., Zeng, B., 2007. Directional Discrete Cosine Transforms: a Theoretical Analysis. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, **1**:1105-1108. [doi:10.

1109/ICASSP.2007.366105]

Han, W.J., Min, J., Kim, I.K., Alshina, E., Alshin, A., Lee, T., Chen, J., Seregin, V., Lee, S., Hong, Y.M., *et al.*, 2010. Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools. *IEEE Trans. Circ. Syst. Video Technol.*, **20**(12): 1709-1720. [doi:10.1109/TCSVT.2010.2092612]

Hung, E.M., de Queiroz, R.L., Mukherjee, D., 2006. On Macroblock Partition for Motion Compensation. Int. Conf. on Image Processing, p.1697-1700. [doi:10.1109/ICIP.2006.312686]

Kondo, S., Sasai, H., 2005. A Motion Compensation Technique Using Sliced Blocks in Hybrid Video Coding. IEEE Int. Conf. on Image Processing, **2**:305-308. [doi:10.1109/ICIP.2005.1530052]

Marpe, D., Schwarz, H., Bosse, S., Bross, B., Helle, P., Hinz, T., Kirchhoffer, H., Lakshman, H., Nguyen, T., Oudin, S., *et al.*, 2010. Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding. *IEEE Trans. Circ. Syst. Video Technol.*, **20**(12):1676-1687. [doi:10.1109/TCSVT.2010.2092615]

Mathew, R., Taubman, D.S., 2010. Quad-tree motion modeling with leaf merging. *IEEE Trans. Circ. Syst. Video Technol.*, **20**(10):1331-1345. [doi:10.1109/TCSVT.2010.2077480]

Muhit, A.A., Pickering, M.R., Frater, M.R., 2009. A Fast Approach for Geometry-Adaptive Block Partitioning. Picture Coding Symp., p.1-4. [doi:10.1109/PCS.2009.5167365]

Robert, A., Amonou, I., Pesquet-Popescu, B., 2008. Improving H.264 Video Coding Through Block Oriented Transforms. IEEE Int. Conf. on Multimedia and Expo, p.705-708. [doi:10.1109/ICME.2008.4607532]

Schwarz, H., Wiegand, T., 2001. Tree-Structured Macroblock Partition. VCEG-O17. 15th JVT Meeting.

Taubman, D., Zakhor, A., 1994. Orientation adaptive subband coding of images. *IEEE Trans. Image Process.*, **3**(4):421-437. [doi:10.1109/83.298396]

Wiegand, T., Ohm, J.R., Sullivan, G.J., Han, W.J., Joshi, R., Tan, T.K., Ugar, K., 2010. Special section on the joint call for proposals on High Efficiency Video Coding (HEVC) standardization. *IEEE Trans. Circ. Syst. Video Technol.*, **20**(12):1661-1666. [doi:10.1109/TCSVT.2010.2095692]

Wien, M., 2003. Variable block-size transforms for H.264/AVC. *IEEE Trans. Circ. Syst. Video Technol.*, **13**(7):604-613. [doi:10.1109/TCSVT.2003.815380]

Xu, H., Xu, J.Z., Wu, F., 2007. Lifting-based directional DCT-like transform for image coding. *IEEE Trans. Circ. Syst. Video Technol.*, **17**(10):1325-1335. [doi:10.1109/TCSVT.2007.903552]

Ye, Y., Karczewicz, M., 2007. Improved Intra Coding. VCEG-AG11. 33rd JVT Meeting.

Zeng, B., Fu, J.J., 2006. Directional Discrete Cosine Transforms for Image Coding. IEEE Int. Conf. on Multimedia and Expo, p.721-724. [doi:10.1109/ICME.2006.262400]

Zeng, B., Fu, J.J., 2008. Directional discrete cosine transforms—a new framework for image coding. *IEEE Trans. Circ. Syst. Video Technol.*, **18**(3):305-313. [doi:10.1109/TCSVT.2008.918455]

Zhang, C.X., Ugur, K., Lainema, J., Hallapuro, A., Gabbouj, M., 2011. Video coding using spatially varying transform. *IEEE Trans. Circ. Syst. Video Technol.*, **21**(2):127-140. [doi:10.1109/TCSVT.2011.2105595]

Zhang, Y.X., Wang, W.D., Zheng, L.X., Wu, M.X., 2009. Motion Compensation Using Polyline Based Block Partition. 2nd Int. Congress on Image and Signal Processing, **1**:290-294. [doi:10.1109/CISP.2009.5304312]

Zhang, Y.X., Chen, L.Y., Wang, W.D., 2010. Reorder discrete cosine transform method for video compression. *J. Zhejiang Univ. (Eng. Sci.)*, **44**(9):1705-1710 (in Chinese). [doi:10.3785/j.issn.1008-973X.2010.09.013]