A Fast Encoder of Frame-compatible Format Based on Content Similarity for 3D Distribution

Xin Jin¹, Zhuoying Zeng², Satoshi Goto³ and Qionghai Dai¹

¹Shenzhen Key Lab of Broadband Network & Multimedia, Graduate School at Shenzhen, Tsinghua Univ., China

²China Academy of Electronics and Information Technology, Beijing, China

³Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan

jin.xin@sz.tsinghua.edu.cn, stacey.zeng@gmail.com, goto@waseda.jp, qhdai@tsinghua.edu.cn

ABSTRACT

Frame compatible format (FCF), which multiplexes the left and right eye views into one frame spatially, has been considered as one of the most promising solutions for 3D video broadcasting. In this paper, a fast algorithm exploiting the content similarity is proposed to reduce the computational complexity for coding FCF. A low complexity average shift estimation method is proposed to find the matching blocks in the two interleaved views, based on which mode candidate sets are designed for accelerating intra and inter coding of the second interleaved view (the interleaved view at the bottom/right in FCF) by exploiting inter-view correlations. The proposed algorithm achieves an average of 72.45% reduction in Intra coding complexity and an average of 81.08% reduction in Inter coding complexity for the second interleaved view, respectively, with negligible loss in compression efficiency. By combining with conventional fast algorithms, that is applying an conventional fast algorithm to code the view on the top/left and the proposed one to code the view on the bottom/right, the whole encoder provides up to 5.44 times higher encoding speed relative to applying the conventional algorithm solely to the whole frame with negligible performance degradation.

KEY WORDS

Frame-compatible format (FCF) coding, fast video encoding, content similarity, fast average shift estimation, 3D video coding

1 INTRODUCTION

Through imitating the real world precisely, 3D technologies have been widely employed to provide impressive visual experiences. As the 3D market matures, the development of production, delivery and display is steadily proceeding. However, the bit rate for solely coding the neighboring views is rather high, while Multi-view Video Coding (MVC) technique is not compatible with the existing decoding devices at the user side. Consequently, frame compatible format (FCF) [1], which interleaves the two neighboring views into a single frame, has been introduced and adopted in advanced video coding standards, such as H.264/AVC [2][3].

FCF refers to a class of formats designed for stereoscopic 3D video, in which the left eye view and the right eye view are sub-sampled and spatially interleaved together in each frame [4]. Fig. 1 [3] depicts the generation process of this format by taking top-bottom format as an example. Some spatial multiplexing formats are shown in Fig. 2 [1]. Among them, the top-bottom and side-by-side format are the most widely used.

Figure. 1. Generation process of top-bottom frame-compatible format from left and right views.

Figure. 2. Common FCF: (a) top-bottom; (b) side-by-side; (c) column interleaved; (d) row interleaved; and (e) checkerboard, where “x” and “o” represent the pixels from one view and another, respectively.

FCF provides a solution for 3D video content distribution by fully utilizing the existing infrastructure and equipments. Encoding FCF using the existing video coding standards, e.g. H.264/AVC [2], and decoding it using the existing video decoders with up-sampling to reconstruct the two neighboring views relieve the application providers from updating the current infrastructure. The compressed data can be directly decoded by the existing set-top boxes

which are widely installed at the customer side. The advantages promote this format to be an advanced near-term approach for 3D program on current existing equipments. The signaling for FCF has been standardized in H.264/AVC as SEI [1][3] and its compatible compression techniques have been standardized as multi-resolution frame compatible stereo coding extension of AVC [5]. HDMI version 1.4, the interface of TV, has also announced their overall support for FCF [6].

Considering not only the applications using hardware-based encoders, e.g. satellite and terrestrial broadcasting, but also the applications using software-based or programmable processor based encoders, e.g. cable and IPTV, have started deploying real-time stereoscopic 3D video services using FCF, fast encoding schemes especially benefiting FCF coding are desired to improve the computational efficiency. Currently, one of the most popular ways of encoding FCF is using H.264/AVC. A number of accelerating techniques have been proposed for H.264/AVC for coding 2D videos, while none of them is dedicated to FCF coding. Fast algorithms designed for H.264/AVC Intra encoding focus on accelerating the rate distortion optimization (RDO) process, e.g. the work proposed by [6], and reducing the complexity in prediction by exploiting spatial and context correlations [8]-[17]. Among them, [8] [9] proposed partial sampling prediction; [10] proposed enhanced sum of absolute transformed differences; [11]-[17] exploited the spatial correlations in predicted pixels to perform a hierarchical mode decision or mode skipping. Fast algorithms designed for H.264/AVC Inter coding mainly focus on three aspects: fast mode decision (FMD) [18]-[25], fast motion estimation (FME) [19][26]-[28], and novel RDO methods [29]-[31]. The approaches of FMD mainly exploit the spatial homogeneity information to shrink the candidate mode set for early termination of prediction. [18] is mainly for Intra mode decision while [19]-[21] focus on the Inter modes, and [22]-[25] accelerate both Intra and Inter mode decision. FME approaches are proposed in [19][26]-[28] to reduce the complexity in ME by fast integer-pel and fractional-pel searching with different steps and searching patterns. [29] and [30] developed novel rate distortion models for H.264/AVC coding. [31] tries to classify the Intra and Inter coding modes into several clusters according to rate-distortion feature vectors. Also, some approaches based on machine learning are proposed like the mode decision scheme using particle swarm optimization [32]. Although most of the existing fast algorithms can be applied to FCF coding, none of them makes a good use of the content similarities between the two interleaved views for accelerating, which is the most distinctive characteristic of FCF.

It is observed that Inter-view correlations are always stronger than spatial correlations and more stable than temporal

1 correlations for FCF format. Due to the short baseline between the left eye view and the right eye view, the two views
 2 multiplexed in FCF are usually similar except a tiny shift. Fig. 3 shows an example of top-bottom FCF. As shown in the
 3 figure, the two macroblocks (MBs), squared in red, are visually the same with a horizontal shift. Statistical data have
 4 been retrieved for sequences packed in FCF format to compare the spatial, temporal and Inter-view correlations. The
 5 spatial correlation is calculated by the maximum correlation between the current block in Plane_1, as shown in Fig. 3,
 6 and its spatially adjacent blocks (the block above, left to, top-left to and top-right to the current block). The temporal
 7 correlation is calculated between the current block in Plane_1 and its collocated block in the previous frame. The
 8 Inter-view correlation is calculated between the current block in Plane_1 and its corresponding block in Plane_0, like
 9 the two blocks shown in Fig.3. By testing the sequences with representative features, statistical data showed that
 10 MB-averaged Inter-view correlation is 0.813, while, the MB-averaged spatial correlation is only 0.759, which is much
 11 weaker than the former. For the videos with complex motions and big moving objects, Inter-view correlation is 0.1
 12 stronger than that of spatial correlation. Although Inter-view correlation is 0.04 lower than temporal correlation on
 13 average, it is much stronger than temporal correlation for the sequence with vast motion and captured by moving
 14 cameras and it is more stable to the variations in video content. Since areas with similar content always tend to use
 15 similar prediction mode during encoding, exploiting such inter-view correlation in FCF enables us to reduce the
 16 prediction mode candidates for Plane_1 (the interleaved view on the bottom/right in top-bottom/side-by-side FCF, as
 17 shown in Fig. 3) according to the encoded Plane_0 (the interleaved view on the top/left in FCF, as shown in Fig. 3).

Figure. 3. An example of top-bottom FCF, where the blocks in red represent the blocks with the same content.

18 Consequently, in this paper, taking the top-bottom FCF as an example, a fast algorithm both for Intra and Inter
 19 encoding of FCF using H.264/AVC is proposed. First, a low-cost and efficient average shift estimation method is
 20 proposed to find the two corresponding blocks with the same content in the interleaved views. Based on the proposed
 21 average shift estimation algorithm, the prediction correlations between the corresponding blocks are investigated for

both Intra and Inter coding. Prediction mode candidate sets are designed by fully exploiting the Inter-view prediction correlations to save a big amount of computations for Plane_1 encoding. Some preliminary results have been published in [33][34]. As a result, the proposed approach provides an average of 72.45% computational reduction for Plane_1 in Intra encoding and an average of 81.08% computational reduction for Plane_1 in Inter encoding, respectively, with negligible loss in compression efficiency. Compared with the existing approaches, the proposed method presents the following advantages:

- 1) targeting fast encoding of FCF by exploiting the inter-view prediction correlations;
- 2) more efficient in reducing the computational complexity compared with the existing fast algorithms;
- 3) stable in reduction of computational complexity to video sequences captured by either moving cameras or static cameras and stable to videos with different motion features;
- 4) independent from depth maps and camera parameters to facilitate the stereoscopic coding.

The remainder of this paper is organized as follows. The architecture of the encoding system is presented in Section 2 with the proposed low-cost average shift estimation method. The prediction candidate mode sets are described in detail in Section 3. The encoding scheme is introduced in Section 4. The experimental results and discussions are given in Section 5. Section 6 concludes the paper.

2 ARCHITECTURE AND AVERAGE SHIFT ESTIMATION

2.1 System Architecture

The proposed FCF fast encoding system diagram is shown in Fig. 4. As shown in the figure, P_{n_0} and P_{n_1} represent the Plane_0 and Plane_1 of frame n , respectively. For Plane_0, all the MBs are encoded according to the coding process defined in an H.264/AVC encoder. No special processing is designed except that the prediction modes of each block in Plane_0 will be recorded by *Mode Recording*. For Intra coding, partition size and prediction direction will be recorded for each 4×4 block. For Inter coding, prediction mode will be recorded for each 8×8 block. The recorded modes will be used as reference modes for the MBs in Plane_1.

For Plane_1, the process for the first frame is a bit different from that of the latter frames. For the first frame, P_{0_1} , MBs will go through *Average Shift Estimation* to estimate the relative shifting, corresponding to the disparity, between Plane_0 and Plane_1 at block accuracy. Block-wise searching is performed for the MBs in the central row of Plane_1 to

1 find their best matches in Plane_0. The block in Plane_1 and its best match in Plane_0 are called corresponding block
 2 one to the other. The horizontal distance between the two matched blocks is accumulated and averaged, which will be
 3 exploited by *Mode Retrieval* to get the coding mode from Plane_0 for an MB in Plane_1. *Average Shift Estimation* will
 4 be described in detail in the next subsection.

5 Then, each MB in Plane_1 will go through *Mode Retrieval* to obtain the prediction mode of its corresponding block
 6 in Plane_0 according to the average shift. *Fast Intra/Inter Prediction* further maps the reference mode to the candidate
 7 modes according to the designed candidate mode sets to accelerate the mode decision process for the block in Plane_1.
 8 The candidate sets are designed based on the statistical correlations in prediction modes between Plane_0 and Plane_1,
 9 which will be described in detail in Section III. Finally, the residue between the original block and the predicted block
 10 will be coded according to H.264/AVC [2].

Figure. 4. The diagram of the proposed FCF encoding system.

11 2.2 Average Shift Estimation

12 As one of the key modules of the proposed FCF encoding system, *Average Shift Estimation* retrieves the average
 13 shift between the two interleaved views. The shift is defined to be the disparity at block accuracy. If the depth map and
 14 the camera parameters are available, the disparity can be calculated using camera geometry and depth information [35].
 15 However, since no depth information is available for FCF and generating the depth maps from the two views is really
 16 time consuming, we focus on retrieving the disparity directly from the texture information. Global disparity estimation
 17 (GDE) approaches, such as that applied in JMVM [36], the edge based solution [37] and the SIFT based approach [38]

belong to this type, which retrieve the disparity for two views without using the depth or camera information. However, GDE is always “frame matching” scheme, which involves all the pixels in the two views into estimation. Two-dimensional search is always performed for frame matching, which is computationally expensive. Considering a block is the basic coding unit in generally used video coding standards, the disparity at block accuracy between the two views is required. A block-based disparity vector derivation using the neighboring blocks has been developed in [39]. It requires motion vector information of all the spatially adjacent blocks and the blocks in the candidate regions from all the reference pictures, which is tightly bounded with encoding process. Thus, it is not suitable to a pre-decision before encoding. Consequently, a low complexity block-wise average shift estimation method is proposed based on the analysis of representative video contents.

Several multi-view sequences with representative motion features and capturing features have been analyzed to detect where the main objects would probably occur. The main object is defined as the moving foreground with area larger than 1% to the whole image, which draws the greatest attention from the viewers. The main objects in the sequence are manually notated using a macroblock as the basic unit. Three MB rows at the top, in the middle and at the bottom of the image, which locate at $1/8$, $1/2$, and $7/8$ height of one frame, have been analyzed. If the maximum width of the successive macroblocks belonging to the main object on that line is more than 20% of the maximum width of the main object in manual notations, the main object is determined as appearing on that line and the appearance frequency of that line is added by one. By averaging the appearance frequency over the five tested sequences covering indoor and outdoor contents captured by static and moving cameras, the probabilities of the main object appearing in the top, middle and bottom MB row are 12.5%, 93.75% and 68.75% on average, respectively. It is highly consistent with intuitive understanding that: the objects, which the photographer wants to show, are always located in the center of the image. With the highest probability in presenting the main content, the middle row of blocks in each plane is exploited by *Average Shift Estimation*.

Using the stereo video captured by two horizontally arranged cameras as an instance, the proposed average shift estimation algorithm is shown in Fig. 5.

Figure. 5. Searching scheme of top-bottom FCF for the views captured by horizontally arranged cameras.

As shown in the figure, a block in the middle block row of Plane_1 is denoted by B_i , $i \in [0, I_w/B_w - 1]$, in which I_w and B_w represent the width of the image and the width of a block, respectively. To find the corresponding block in Plane_0 for B_i (the block having the same content with B_i), the middle block row of Plane_0 is searched using the horizontal search range SR centered at $i \times B_w$. SR is set to 48 (i equals to 3), which has been verified to be usually wide enough for estimation because of the short baseline between the left eye view and right eye view. The search is performed using block searching step SS . Considering the minimum processing unit during H.264/AVC encoding is a 4×4 block and to facilitate integrating the proposed approach into an existing encoder architecture, SS is usually set as the width of the minimum prediction unit, like 4, during searching process. The blocks whose top-left corners located between $(i \times B_w - SR)/SS$ and $((i+1) \times B_w + SR)/SS$ are searched, which are denoted by B_t in Fig. 5. Among B_t , the block with the minimum sum of absolute difference (SAD) value comparing with B_i is set as the corresponding block, B_i' . The searching process is given by

$$B_i' = \arg \min_{t \in \left[\frac{(i \times B_w - SR)}{SS}, \frac{((i+1) \times B_w + SR)}{SS} \right]} \left(\sum_{j,k=0}^{B_w-1} |p_{i,j,k} - p_{t,j,k}| \right), \quad (1)$$

where $p_{i,j,k}$ and $p_{t,j,k}$ are pixel (j, k) in B_i and B_t , respectively. After retrieving (x_i, y_i) and (x_i', y_i') , the top-left corner locations of B_i and B_i' , respectively, Δx_i between B_i and B_i' is given by

$$\Delta x_i = x_i' - x_i. \quad (2)$$

For the blocks at the corner of the row, since they may be outside of the view packed in Plane_0, they will be skipped for average shift estimation. If Plane_1 packs the view on the right/left, empirically, four blocks at the right/left corner

of the row will be skipped, because of the short baseline between the two packed views. Finally, the average shift for Plane_1, $\overline{\Delta x}$, is given by:

$$\overline{\Delta x} = \left(\frac{B_w}{I_w} - 4 \right) \sum_{i=0}^{(I_w/B_w-5)} \Delta x_i. \quad (3)$$

The accuracy of the proposed average shift estimation algorithm is evaluated by comparing four approaches shown in Fig. 6. *M1* is the proposed average shift estimation algorithm, which only performs searching for the middle block row in P_{0_1} , retrieves the averaged shift by Eq. (3), and applies it to all the blocks in Plane_1 in each frame of the sequence, P_{n_1} , for fast prediction. *M3* performs searching and calculating the average shift for the middle block row in P_{n_1} instead of only in P_{0_1} , and applies the average shift to the blocks in each P_{n_1} accordingly. *M2* and *M4* retrieve a shift for each block in Plane_1 instead of calculating the averaged value. *M2* performs searching only for P_{0_1} , while *M4* performs it for P_{n_1} .

Figure 6. Four shift estimation methods.

Through integrating the four approaches into the proposed fast encoding system for FCF, two video sequences: “Alt Moabit” [40] (1024×768, taken by static cameras), and “Poznan_Hall” [41] (1920×1088, taken by moving cameras), are Intra encoded by setting B_w to 16, SS to 4 and SR to 48 in *Average Shift Estimation*. For each sequence, 100 frames have been encoded at four QP values, 22, 27, 32 and 37, by H.264/AVC high profile. Compared with the original H.264/AVC encoding using reference software JM16.2 [42] at high profile, the compression efficiency evaluated by BD-PSNR [43] of the four shift estimation methods together with the percentage of computations consumed by shift estimation (denoted by “Shift Computing”) are listed in Table I. *Shift Computing* is measured by the execution time consumed by shift estimation relative to that consumed by the whole encoding process.

TABLE I. EXPERIMENTAL RESULTS OF SHIFT ESTIMATION METHODS.

Seq.	Method	BD-PSNR[43] (dB)	Shift Computing (%)
Alt Moabit (1024×768)	M1	-0.163	0.00102
	M2	-0.160	0.00663
	M3	-0.162	0.10193

	M4	-0.150	0.66256
	GDE[36]	-0.239	0.19275
Poznan_Hall (1920×1088)	M1	-0.151	0.00064
	M2	-0.154	0.00503
	M3	-0.150	0.06331
	M4	-0.141	0.55493
	GDE[36]	-0.163	0.22116

As shown in the table, $M4$ provides the highest compression efficiency since it has the highest accuracy in the shift. However, its complexity is hundreds of times higher than others. The average shift obtained by $M3$ can adapt to the motion features of each frame, which results in better performance compared with $M1$. But the performance improvement is too small if taking more than one hundred times increment in complexity into consideration. $M2$ performs better than $M1$ when the video is captured by static cameras, e.g. Alt Moabit, while performs worse for the video capture by moving cameras, like Poznan_Hall. It is because that the exact shift value of each block in the first frame has much lower tolerance of variations, which may happen in the following, compared with an averaged shift value. $M1$ demonstrates its best trade-off between the computational complexity and the compression efficiency with more stable performance to videos taken by either static cameras or moving cameras. The performance of applying GDE [36] is also listed in TABLE I, which demonstrates that the proposed approach outperforms it both in terms of computational complexity and compression efficiency. Consequently, $M1$ is finally selected to retrieve the average shift for fast prediction in our encoding system.

3 MODE CANDIDATE SETS DESIGN

Based on the proposed *Average Shift Estimation*, each block which needs to be coded in P_{n-1} retrieves its reference block in P_{n-0} using the averaged shift value, gets the prediction mode of the reference block which has been recorded during Plane_0 encoding and applies fast coding flow according to the candidate mode sets designed based on the correlations in prediction between the two corresponding blocks. The candidate mode sets which are designed for Intra and Inter coding are introduced in detail in this section.

The candidate mode sets are designed based on prediction correlations between the two corresponding blocks. Statistical data are retrieved through counting the conditional probability between the prediction mode of the block in Plane_1 and that of its reference block in Plane_0. For Intra frame, the minimum block size of prediction mode

recording and comparing is 4×4 , which is the same with the minimum Intra prediction unit. For Inter frame, statistical data are retrieved for MB types like PSKIP, 16×16 , 16×8 , 8×16 , 8×8 , Intra_4x4, Intra_8x8 and Intra_16x16. While, for the partition types smaller than 8×8 , no statistical data are retrieved because of the low utilization rate.

Five sequences, listed in Table II, are used for statistical data retrieval and analysis by multiplexing two neighboring views into top-bottom format. The view numbers are listed in Table II. 100 frames of each sequence are encoded using JM16.2 [42] with brute-force RDO based mode decision at four QP values, 22, 27, 32 and 37, by H.264 high profile. The prediction mode selected by each block is recorded for analysis.

TABLE II. TRAINING SEQUENCES.

FCF Seq.	Resolution	Original Multiview Seq.	View No.
Book	1024×768	Book_Arrival [40]	1,2
Leave		Leaving_taptop [40]	10,11
Alt56		Alt Moabit [40]	5,6
Lovebird		Lovebird1[44]	3,4
News		Newspaper[39]	5,6

3.1 I Frame Prediction Correlation and Candidate Sets

First, the correlations in Intra prediction partition types, that are Intra_16x16, Intra_8x8 and Intra_4x4 [2], are analyzed. Letting PT_1 and PT_0 represent the Intra prediction partition type (PT) of the block in Plane_1 and that of its reference block in Plane_0, respectively, the conditional probability of PT_1 given PT_0 , $P(P_{T_1}|P_{T_0})$, is analyzed. Fig. 7 (a) to Fig. 7 (c) depict $P(P_{T_1}|P_{T_0})$ given that PT_0 equals to Intra_16x16, Intra_8x8 and Intra_4x4, respectively. Each $P(P_{T_1}|P_{T_0})$ is an averaged value over five sequences and four QP values.

Figure. 7. Averaged prediction correlation of Intra partition type as PT_0 equals to: (a) Intra_16x16; (b) Intra_8x8 and (c) Intra_4x4.

Based on the data in Fig. 7, it can be calculated that the MB in Plane_1 takes an average of 67.29% to choose the same partition type with that of its reference MB in Plane_0. The sum of the highest two probabilities averaged over

three types of PT_0 can reach 88.17%.

Furthermore, $P(P_{T_1}|P_{T_0})$ averaged over different sequences at different QPs are also retrieved to analyze the variations with QP. As shown in Fig. 8, as PT_0 equals to Intra_4×4 and Intra_16×16, the mode with the highest and the second highest probability to be selected by PT_1 keeps the same with the increase in QP. While, as PT_0 equals to Intra_8×8, the mode which takes the second highest probability to be selected by PT_1 switches from Intra_4×4 to Intra_16×16 as QP increases to more than 27. Thus, the partition type candidate set, denoted by “Candi_PT”, for I frame is designed in Table III.

Figure 8. Variations in $P(P_{T_1}|P_{T_0})$ as P_{T_0} equals to: (a) Intra_4×4; (b) Intra_8×8 and (c) Intra_16×16.

TABLE III. I FRAME PARTITION CANDIDATE SET.

PT_0	Candi_PT
Intra_4×4	Intra_4×4 & Intra_8×8
Intra_8×8	Intra_8×8 & Intra_4×4 (QP ≤ 27) Intra_8×8 & Intra_16×16 (QP > 27)
Intra_16×16	Intra_16×16 & Intra_8×8

Similarly, the correlations in Intra prediction directions are analyzed by taking 4×4 as the minimum unit. Denoting the prediction direction (PD) of the block in Plane_1 by PD_1 and that of its reference block in Plane_0 by PD_0 , the averaged conditional probabilities of PD_1 given PD_0 are shown in Fig. 9. As shown in the figure, the prediction directions are numbered according to H.264/AVC [2]: vertical, horizontal, DC, diagonal down-left, diagonal down-right, vertical-right, horizontal-down, vertical-left, horizontal-up prediction direction are numbered from 0 to 8 both for Intra_4×4 (Fig.9 (a) to (i)) and Intra_8×8 (Fig. 9 (j) to (r)), respectively, while vertical, horizontal, DC and plane direction correspond to 0 to 3 for Intra_16×16 (Fig.9 (s) to (v)), respectively. It can be observed that PD_1 prefers to choose the same direction as PD_0 . Meanwhile, analysis on variations in $P(P_{D_1}|P_{D_0})$ under different QPs is also

performed, which indicates that no obvious dependency exist between $P(PD_1|PD_0)$ and QP. Consequently, PD_0 and the direction with the highest conditional probability among all the possible directions, excluding PD_0 , are selected as the candidate prediction directions for the block in Plane_1. Thus, Intra prediction direction candidate sets, denoted by “Candi_PD”, are designed and listed in Table IV. For a block in Plane_1, a candidate set will be selected according to the prediction direction of its reference block in Plane_0.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

(m)

Figure 9. Prediction correlations of Intra prediction directions: (a) to (i) represent PD_0 equals to Intra_4×4 vertical, horizontal, DC, diagonal down-left, diagonal down-right, vertical-right, horizontal-down, vertical-left, horizontal-up, respectively; (j) to (r) corresponds to PD_0 equals to Intra_8×8 from vertical to horizontal-up, respectively; and (s) to (v) represents PD_0 equals to Intra_16×16 vertical, horizontal, DC and Plane, respectively.

TABLE IV. I FRAME DIRECTION CANDIDATE SET.

Intra_4×4		Intra_8×8		Intra_16×16	
PD_0	Candi_PD	PD_0	Candi_PD	PD_0	Candi_PD
0	0,1	0	0,1	0	0,3
1	1,0	1	1,2	1	1,3
2	2,1	2	2,1	2	2,1
3	3,1	3	3,1	3	3,1
4	4,1	4	4,1		
5	5,1	5	5,1		
6	6,1	6	6,1		
7	7,1	7	7,1		
8	8,1	8	8,1		

3.2 Inter Frame Prediction Correlation and Candidate Set

The correlations of Inter frame prediction can be retrieved similarly to that of Intra frame, except that the minimum unit is altered to 8×8 and the analysis solely focuses on the MB partition level. Targeting real-time 3D video applications, only the scheme for P frame is designed for Inter coding, which can be extended to B frame easily.

Fig. 10 shows the statistical results by taking PT_0 equal to PSKIP (Fig.10 (a)) and $P16 \times 16$ (Fig.10 (b)) as instances. As shown in Fig. 10 (a), PT_1 has more than a 90% chance of selecting PSKIP when PT_0 is PSKIP. As PT_0 is $P16 \times 16$, as shown in Fig. 10 (b), the probability of selecting PSKIP as PT_1 is still higher than that of selecting $P16 \times 16$.

Figure. 10. Prediction correlations of partition type in Inter frame as PT_0 equals to: (a) PSKIP; (b) $P16 \times 16$.

Moreover, if PT_0 is $P8 \times 8$, the two partition types with the highest conditional probability vary with QP. Fig. 11 illustrates the variation in the three highest probability in $P(PT_1|PT_0)$ for PT_0 equals to $P8 \times 8$ under different QPs using two sequences as instances. As shown in the figure, when QP drops, the two partition types with the highest probability are $P16 \times 16$ and $P8 \times 8$. As QP increases, the two partition types turn to be $P8 \times 8$ and $P16 \times 8$ or even further to be $P16 \times 16$ and $P16 \times 8$. Consequently, the partition type candidate set, $Candi_PT$, varying with QP at $PT_0 = P8 \times 8$ for Inter frame is designed in Table V.

Figure. 11. Variations in $P(P_{T1}|P_{T0})$ as P_{T0} is 8×8 for: (a) Race; (b) News.

TABLE V. P FRAME PARTITION CANDIDATE SET

PT_0	Candi_ PT
PSKIP	PSKIP & 16×16
16×16	16×16 & PSKIP
16×8	16×8 & 16×16
8×16	PSKIP & 16×16
8×8	$QP \leq 30$: 16×16 & 8×8
	$30 < QP \leq 38$: 8×8 & 16×8
	$QP > 38$: 16×16 & 16×8
Intra_ 4×4	Intra_ 4×4 & Intra_ 16×16
Intra_ 8×8	Intra_ 8×8 & PSKIP
Intra_ 16×16	Intra_ 16×16 & PSKIP

4 FAST ENCODING FLOW

Finally, a fast encoding scheme is designed to encode the blocks in FCF according to the flow in Fig. 12. As shown in the figure, PT_0 and PD_0 , selected by the original encoding flow, e. g. brute-force RDO based mode decision, are recorded during Plane_0 encoding. They are retrieved by the blocks in Plane_1 according to the average shift to accelerate Plane_1 encoding using the candidate sets designed above.

Figure. 12. The flowchart of the proposed fast encoding.

In I frame encoding, the retrieved average shift is integer multiples of 4 (SS is 4 pixels). It benefits PD_0 retrieving. When the shift is not integer multiples of the width of an MB (16 pixels), PT_0 is retrieved from the MB with the largest overlapped area. If the overlapped area is just half of one MB, PT_0 retrieves the partition types both from the left and right MB, adds them into the candidate set and performs mode decision for the maximum three candidate partition types.

Furthermore, for Intra prediction directions, as the candidate partition type applied to the block in Plane_1 is the same with that of the reference block in Plane_0, Table IV is directly applied. When the partition type is different, like the cases shown in Table VI, further rules are designed.

TABLE VI. REFERENCE AND CANDIDATE PARTITIONS CASES

Case	(PT_0 , Candi_PT)
1	(Intra_4×4, Intra_8×8) or (Intra_8×8, Intra_4×4)
2	(Intra_16×16, Intra_4×4) or (Intra_16×16, Intra_8×8)
3	(Intra_4×4, Intra_16×16) or (Intra_8×8, Intra_16×16)

For Case 1, as shown in Table VI, since the directions defined for Intra_8×8 are the same with those of Intra_4×4 in H.264/AVC, the direction candidate set got from the reference partition and reference direction according to Table IV is directly applied despite that $Candi_PT$ is different from PT_0 .

For Case 2, the 9 directions defined for Intra_4×4 or Intra_8×8 can cover most of the directions of Intra_16×16. And also, $Candi_PDs$ designed as $PT_0 = \text{Intra_16} \times 16$ are similar to those designed as $PT_0 = \text{Intra_4} \times 4$ and Intra_8×8. Thus, according to PD_0 of Intra_16×16, it is directly mapped to PD_0 of Intra_4×4 or Intra_8×8 (column 1 or column 3 of Table IV) to retrieve $Candi_PD$ for Intra_4×4 or Intra_8×8 partition.

For Case 3, the prediction directions available for Intra_4×4 or Intra_8×8 are more than those of Intra_16×16. Statistical data illustrate that the horizontal and plane direction of Intra_16×16 occupy higher usage. So, when PD_0 is one of {diagonal down-right, vertical-right, horizontal-down, vertical-left, horizontal-up} (numbered from 4 to 8) in Intra_4×4 or Intra_8×8, the proposal applies horizontal and plane prediction direction in Intra_16×16 as candidates.

In P frame encoding, PT_0 is also retrieved from the reference MB with the largest overlapped area similarly to that in I frame, except that the recording unit is 8×8 block.

5 EXPERIMENTAL RESULTS

The proposed fast algorithm has been integrated into JM16.2 [42], the reference software of H.264/AVC, for performance evaluation. The testing sequences are listed in Table VII, which are generated by multiplexing the neighboring two views into top-bottom FCF. The sequences are also categorized according to capturing condition, camera motion, baseline distance and content features in Table VIII. The average shift values retrieved during encoding (after performing *Average Shift Estimation*) are also provided. The negative shift value represents moving leftward from the collocated position in Plane₀ to get the corresponding block for the current block in Plane₁. Targeting real-time 3D applications, the low delay coding structures are tested. The major coding parameters are shown in Table IX.

TABLE VII. TEST SEQUENCES

FCF seq	Resolution	Multiview seq	View No.
Race	640×480	Race1 [46]	6,7
Vassar		Vassar [47]	6,7
Alt34	1024×768	Alt Moabit [40]	3,4
Flower		Door_flower[40]	7,8
Book		Book_Arrival [40]	1,2
News		Newspaper[45]	5,6
CP	1920×1088	Carparker [41]	5,6
Hall		Hall [41]	0,1

TABLE VIII. CONTENT FEATURES AND AVERAGE SHIFT VALUE

Seq.	Indoor/ Outdoor	Camera Movement	Baseline (cm)	Content Features	Average Shift (4-pixel)
Race	Outdoor	Rotating Cams	20	Vast motion with global motion	-1
Vassar	Outdoor	Static	19.5	Noisy video with fast motion	-6
CP	Outdoor	Static	13.75	Most Static	-6
Hall	Indoor	Horizontal Moving	13.75	Global Motion	-10
Alt34	Outdoor	Static	6.5	Fast motion	-2
Flower	Indoor	Static	6.5	Most Static	-3
Book	Indoor	Static	6.5	Most Static	-2
News	Indoor	Static	5	Most Static	-6

TABLE IX. MAJOR CODING PARAMETERS

Parameters	Value
Profile	High
Frames to be encoded	100
Number of reference frames	4
GOP	III...(Intra Only)
	IPPP...(Inter)

Rate distortion optimization	on
Search range	64
CABAC	On
Quantization parameter(I/P)	22/23, 27/28, 32/33, 37/38

By being applied to Plane_1 encoding solely, two representative existing fast algorithms designed for coding conventional 2D video contents are implemented for comparison: one is intensity gradient filter based fast Intra prediction proposed by Tsai *et al.* [12], which is used for Intra coding comparison; the other is early SKIP mode decision and selective Intra prediction proposed by Jeon *et al.* [22], which has been adopted in reference encoder JM, for Inter coding comparison.

The compression efficiency is measured by BD-PSNR and BD-BR [43] using the original JM16.2, which encodes both Plane_0 and Plane_1 using brute-force RDO mode decision, as the benchmark. The reduction in encoding time is measured by two methods: the reduction in total execution time, denoted by ΔT ; and the reduction in encoding time of Plane_1, denoted by ΔT_{p1} . ΔT is given by

$$\Delta T = \frac{T_{JM} - T_{fast}}{T_{JM}} \times 100\%, \quad (6)$$

where T_{JM} and T_{fast} represent the total execution time of the original JM16.2 and that of the fast encoder, respectively. T_{fast} includes the encoding time of the two planes and the time consumed by average shift estimation for the proposed encoder. ΔT_{p1} is given by

$$\Delta T_{p1} = \frac{T_{p1.JM} - T_{p1.fast}}{T_{p1.JM}} \times 100\%, \quad (7)$$

where $T_{p1.JM}$ and $T_{p1.fast}$ represent the encoding time of Plane_1 consumed by the original JM and the fast algorithm, respectively.

The comparison results of Intra encoding are shown in Table X. As shown in the table, the proposed algorithm provides an average of 72.45% reduction in the encoding time of Plane_1 and 36.19% reduction in the total execution time, respectively, with much less performance loss in comparison with Tsai *et al.*[12]. The coding bitrate of the proposed algorithm is 2.4% lower than that of Tsai's approach with around 4 times higher encoding speed on average. The approach proposed by Tsai *et al.* [12] only reduces the number of candidate prediction directions under each prediction partition type by retrieving the spatial gradient features from the image. While, the proposed approach follows top-down scheme to first reduce the number of candidate partition types, e.g. Intra_4×4, Intra_8×8 and

Intra_16×16, according to Table III, and then reduces the number of candidate prediction directions according to Table IV. Thus, all the prediction directions under the disabled partition type will be skipped, which contributes a lot to the encoding speed. Up to 87.90% reduction in the encoding time of Plane_1 can be achieved by the proposed algorithm, which indicates that Plane_1 encoding can be 8.26 times faster than the original JM with only 2.32% increase in the bit rate.

During Intra encoding, the average number of candidate modes processed during encoding is also retrieved for the two approaches. For Tsai's approach, all the MBs need to process the three partition candidates (Intra_4×4, Intra_8×8 and Intra_16×16). Under Intra_4×4, an average of 2.03 candidate prediction directions are processed and compared. Under Intra_8×8 and Intra_16×16, an average of 2.02 and 2 candidate prediction directions are processed, respectively. For the proposed fast algorithm, although all the MBs will enable Intra_8×8, on average, only 65.17% of the MBs and only 34.27% of the MBs will process Intra_4×4 and Intra_16×16, respectively. So, averaged over the total number of macroblocks, only 1.07, 2.14 and 0.76 prediction direction candidates are enabled for Intra_4×4, Intra_8×8 and Intra_16×16, respectively, which are much lower than that of Tsai *et al.* [12]. The maximum number of candidate prediction directions enabled by the proposed algorithm is 6 among the available 22 prediction types defined in the standard, which reduces the encoding complexity obviously.

TABLE X. INTRA ENCODING COMPARISON

Seq.	Tsai's [12]				Proposed			
	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	ΔT_{p1} (%)	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	ΔT_{p1} (%)
Alt34	-0.34	6.20	18.55	36.34	-0.15	2.10	36.22	70.39
Flower	-0.33	6.97	19.84	41.76	-0.22	3.65	36.15	72.09
Book	-0.26	5.37	20.32	36.24	-0.17	2.71	36.44	69.99
News	-0.19	3.31	20.88	36.61	-0.15	2.64	32.61	66.13
Race	-0.14	2.30	24.99	42.11	-0.12	2.00	38.97	87.90
CP	-0.15	2.89	19.72	37.98	-0.12	2.19	37.09	72.16
Vassar	-0.16	3.11	20.34	37.87	-0.11	1.52	36.91	71.23
Hall	-0.39	7.72	18.15	35.14	-0.14	1.78	35.14	69.69
Avg.	-0.25	4.73	20.35	38.01	-0.15	2.32	36.19	72.45

Fig. 13 compares ΔT_{p1} and BD-PSNR for the proposed algorithm and those of Tsai *et al.* [12] visually. As shown in Fig. 13(a), ΔT_{p1} provided by the proposed encoder is much larger than that of Tsai *et al.* [12] and is stable to all the sequences. The compression performance, shown in Fig. 13 (b), provided by the proposed algorithm is also much more stable than that of Tsai *et al.* [12] both for videos captured by static cameras and videos captured by moving cameras.

Figure 13. (a) ΔT_{p1} comparison; and (b) BD-PSNR comparison.

The results of Inter encoding are compared in Table XI. As shown in the table, compared with Jeon *et al.* [22], the proposed algorithm achieves an average of 81.08% reduction in Plane_1 encoding time relative to 61.52% by Jeon *et al.* [22]. Although the average bitrate increase introduced by the proposed algorithm is about 1.8% higher than that of Jeon's, its improvement in encoding speed is 2 times higher than that of Jeon *et al.* [22]. Jeon *et al.* [22] early terminates the encoding process by detecting skip-mode and selective Intra prediction. So, it performs well with negligible PSNR loss for sequences which are captured by static cameras and have big static background. However, for the videos with vast motions and captured by moving cameras, like sequence "Race" and "Hall", which have low probability to be coded by Skip-mode, the performance of Jeon's algorithm fluctuates obviously. For "Hall", the PSNR loss together with bit rate increase is much larger than those of other sequences. While, for "Race", both ΔT and ΔT_{p1} drop a lot. In comparison, the proposed fast algorithm of FCF reduces the number of prediction modes for all the possible Inter and Intra modes according to Table V. It can provide more stable compression performance both in terms of encoding time reduction and compression efficiency under different capturing conditions. For sequence "Race", it provides 3.26 times higher encoding speed relative to that of Jeon *et al.* [22].

TABLE XI. INTER ENCODING COMPARISON

Seq.	Jeon's [22]				Proposed			
	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	ΔT_{p1} (%)	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	ΔT_{p1} (%)
Alt34	-0.05	1.34	36.59	71.96	-0.29	4.20	41.93	81.26
Flower	-0.05	1.17	35.69	68.47	-0.27	4.90	41.93	81.28
Book	-0.07	1.62	30.88	61.59	-0.24	2.92	40.50	79.93
News	-0.01	0.18	31.54	62.66	-0.17	2.65	41.48	80.79
Race	-0.01	0.35	14.75	31.08	-0.24	3.21	40.00	78.86
CP	-0.002	0.08	38.71	75.43	-0.20	3.35	43.02	84.37
Vassar	-0.02	0.49	34.13	61.86	-0.20	3.43	42.07	82.26
Hall	-0.19	9.90	34.43	59.14	-0.27	5.21	41.52	79.93

Avg.	-0.05	1.89	32.09	61.52	-0.24	3.73	41.56	81.08
------	-------	------	-------	-------	-------	------	-------	-------

1 During Inter coding, the average number of candidate modes enabled is also retrieved for the two approaches. On
2 average, 3.88 candidate modes are enabled for each MB coded by Jeon *et al.* [22]. In contrast, only 2.11 candidate
3 modes are enabled by the proposed approach, which is almost consistent with the execution time reduction.

4 Fig. 14 compares the RD performance for sequence “News” and “Race”. “News” is a representative sequence with
5 slow motions and is captured by static cameras. “Race” is a representative sequence with vast and global motions, and
6 is captured by rotating cameras. As shown in the figure, the compression efficiency of the proposed algorithm is robust
7 to variations in bit rate. No increase in the performance gap is observed with the increase/decrease in the bit rate.

Figure 14. Compression performance comparison for: (a) News and (b) Race.

8 Fig. 15 summarizes the acceleration in encoding Plane_1 by the proposed approach relative to JM, Tsai *et al.* [12]
9 and Jeon *et al.* [22], respectively. As shown in the figure, the proposed approach can be up to 8.26 times faster than JM,
10 4.78 times faster than Tsai *et al.* [12] and 3.26 times faster than Jeon *et al.* [22] for Plane_1 encoding. On average, it is
11 2.44 times faster than Tsai *et al.* [12] and 2.00 times faster than Jeon *et al.* [22].

Figure 15. Encoding acceleration of the proposed approach relative to other approaches.

Furthermore, the proposed fast algorithm can work together with conventional fast algorithms to further accelerate encoding speed for the whole encoder. That is to apply a conventional fast algorithm to Plane_0's encoding and the proposed one to Plane_1's encoding. Table XII shows an instance by applying Jeon *et al.* [22] to Plane_0 and the proposed algorithm to Plane_1 (denoted by *Jeon+Prop*), and compares its coding performance with applying Jeon *et al.* [22] solely to both Plane_0 and Plane_1 (denoted by *JeonAll*). The combination achieves up to 88.29% reduction in total execution time, which lets the whole encoder be 8.53 times faster relative to the original JM for FCF Inter encoding. Compared with applying Jeon *et al.* [22] to the whole frame (*JeonAll*), *Jeon+Prop* drastically accelerates encoding speed to be up to 5.44 times faster and provides stable acceleration. It lets the encoder be at least two times faster than applying Jeon *et al.* [22] solely to the whole frame, which benefits encoding of FCF obviously.

TABLE XII. PERFORMANCE COMPARISON BETWEEN JEON+PROP AND JEONALL.

Seq.	JeonAll		Jeon+Prop		Jeon+Prop vs. JeonAll	
	BD-PSNR (dB)	ΔT (%)	BD-PSNR (dB)	ΔT (%)	ΔT (%)	Speed-up (times)
Race	-0.03	36.24	-0.24	88.29	81.62	5.44
Alt34	0.01	69.57	-0.31	87.37	58.49	2.41
Book	-0.12	74.84	-0.24	87.46	49.91	2.01
Flower	-0.10	74.17	-0.27	87.47	51.34	2.06

By analyzing the performance of the proposed algorithm among the testing sequences across Table VIII, Table X and Table XI, it is observed that: for Intra coding, the compression efficiency of the proposed algorithm mainly depends on the baseline distance. Although for FCF, the baseline between the two packed views is close to eye to eye distance, different capturing systems will still introduce differences in baseline for different sequences. The wider baseline leads to slightly higher compression efficiency and to some extent results in a lower reduction in encoding complexity. While, the efficiency in accelerating encoding speed is influenced jointly by the baseline distance and whether the average shift value is rounded to be a half-integer multiples of MB width. As shown in Table X, for the sequences with wide baseline, like sequence “Vassar”, “CP” and “Hall”, since more blocks at the boundary of Plane_1 cannot retrieve their reference blocks in Plane_0, brute-force mode decision using RDO will be applied to the blocks, which results in a bit higher compression efficiency relative to that of others and to some extent results in lower reduction in encoding complexity. Among the sequences with wide baseline, “Race” is an exception. Although its baseline is wide, the camera rotation compensates the disparity between the neighboring views, which results in small

shift and accordingly high reduction in encoding complexity. On the other hand, short baseline does not always contribute to the reduction of encoding complexity. When the shift value of short-baseline captured sequences, like “Book” and “News”, is rounded to be half-integer multiples of MB width, it lets the corresponding block in Plane_0 have higher probability to be overlapped equally by two MBs. According to the solution described above, the partition types both from the left and right MB will be enabled according to the candidate set, and mode decision will be performed for the maximum three candidate partition types. So, the efficiency in reducing encoding complexity will be a bit lower than other sequences. Similar phenomena exist in sequence “CP” and “Hall”. Although a slight fluctuation in compression efficiency and encoding complexity reduction exists, the performance of the proposed approach is still much more stable than that of Tsai *et al.* [12].

For Inter coding, the compression efficiency of the proposed algorithm not only depends on the baseline distance but also depends on the camera motion, while, the efficiency in accelerating encoding speed is mainly influenced by the camera motion and the motion features of the content. Camera motion, corresponding to global motion in the video, like that in sequence “Race” and “Hall”, degrades temporal correlations in the video, which results in the motion compensated data more sensitive to the selection of prediction mode. While, larger static regions in the video, like that in sequence “CP” and “Vassar”, let the candidate sets “PSKIP & P16×16” and “P16×16 & PSKIP”, listed in Table V, have higher probability to be selected. PSKIP and P16×16 have lower encoding complexity in comparison with other candidate partitions especially for motion estimation, which results in higher reduction in encoding complexity. On the contrary, sequences with vast motions and camera motions will increase the probability of enabling small candidate partition types, which results in a bit reduction in encoding speed acceleration.

Furthermore, performance of the proposed algorithm is compared with a fast mode decision algorithm proposed by Yeh *et al.* [48] for multi-view video coding, which skips a certain coding mode for the current MB based on updated thresholds predicted from the coded view. The algorithm is directly transferred to the testing platform according to the processing flow. The coding modes and the RD cost values of all the modes are recorded for the MBs in Plane_0. The MBs in Plane_1 are fast coded according to the processing flow introduced in [48]. The comparison results are summarized in Table XIII. As shown in the table, the proposed algorithm outperforms Yeh *et al.* both in terms of compression efficiency and encoding complexity. The approach proposed by Yeh *et al.* prefers the videos captured by static cameras with slow motions, like “Book” and “Flower”. While, for the sequences with vast motions or captured by

moving cameras, the compression efficiency fluctuates obviously. Meanwhile, since the algorithm proposed by Yeh *et al.* determines whether to skip a certain coding mode based on the updated thresholds, it cannot solve the worst case that: some MBs will still go through the longest path to find the best coding mode. So, the encoding complexity reduction varies with video contents. Compared with it, the proposed algorithm provides robust performance both in terms of compression efficiency and encoding complexity reduction.

TABLE XIII. ENCODING COMPARISON WITH YEH ET AL. [48]

Seq.	Yeh's [48]				Proposed			
	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	ΔT_{pl} (%)	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	ΔT_{pl} (%)
Alt34	-0.15	2.13	34.09	68.18	-0.29	4.20	41.93	81.26
Flower	-0.16	1.95	35.33	70.66	-0.27	4.90	41.93	81.28
Book	-0.10	2.15	40.10	80.19	-0.24	2.92	40.50	79.93
News	-0.18	3.50	40.05	80.11	-0.17	2.65	41.48	80.79
Race	-0.37	7.52	41.95	83.90	-0.24	3.21	40.00	78.86
CP	-0.15	3.10	31.62	63.24	-0.20	3.35	43.02	84.37
Vassar	-0.36	8.22	42.20	84.40	-0.20	3.43	42.07	82.26
Hall	-0.23	6.94	33.58	67.16	-0.27	5.21	41.52	79.93
Avg.	-0.21	4.43	37.37	74.73	-0.24	3.73	41.56	81.08

The section is concluded by analyzing the storage complexity overhead introduced by the proposed algorithm for encoding implementation. First, for the line buffer of Plane_0: since shift estimation is performed before encoding process and only for the first frame, it can be treated as a pre-processing module, which can reuse the buffer of the encoder. Since video encoding always has line buffers for Intra prediction and motion estimation, the average shift estimation can reuse the buffer space for buffering the block line of Plane_0 without introducing additional overhead to the encoder. Then, for the modes that are saved for Plane_0: for I frame, 2-bit per MB are needed to store the partition mode, e.g. Intra_4×4, Intra_8×8 and Intra_16×16. The MB using Intra_4×4 consumes the highest storage in saving the prediction directions, since each 4×4 block within the MB may choose different prediction directions. So, the worst case of storing prediction modes for I frame consumes 12.672 KB (4-bit per 4×4 block using Plane_0 with the resolution of 1024×384 as an example); for Inter frame, only MB types are stored, which will consume 4 bits per MB and totally 768 B for Plane_0 with the resolution of 1024×384. Furthermore, during encoding of Plane_1, the buffer saving the modes of Plane_0 can be released line by line according to the progress. Consequently, the storage overhead introduced by the proposed algorithm is acceptable both for hardware and software implementations.

6 CONCLUSIONS

In this paper, a fast encoding algorithm is proposed for FCF by exploiting the content similarities between the two interleaved views to reduce the computational complexity. Relying on the proposed low complexity average shift estimation and the candidate mode sets designed based on prediction correlations between the two views, the proposed fast encoding algorithm provides an average of 3.30 and 4.98 times higher encoding speed for Plane_1 Intra and Inter encoding, respectively. In comparison with existing fast Intra and Inter coding algorithms, the proposed algorithm demonstrates its stability in complexity reduction and compression efficiency for videos captured either by static cameras or moving cameras. Besides, this proposal can work together with conventional fast algorithms. The encoder using the conventional algorithm for Plane_0 coding and the proposed one for Plane_1 coding accelerates the whole FCF encoder up to 8.53 times faster, which is also up to 5.44 times faster compared with applying the conventional approach to the whole frame solely. With the development in 3D video capturing and representation, the algorithm will be further tuned according to new features in the new contents to let it be efficient to new types of data. It is possible for the proposed algorithm to exploit additional correlations like spatial correlations and temporal correlations. Applying them to the blocks at the left and right boundary of the image, and combining them with the inter-view correlations can further improve the compression efficiency and accelerating capability. While, how these multi-dimensional correlations can work together efficiently and how to limit the computational and storage overhead for an optimized balance between compression efficiency and encoding complexity need to be further investigated. Also, since the content similarities generally exist in the neighboring views, it is possible to apply the proposed fast algorithm to other standard based stereoscopic coding, like HEVC based 3D video coding and multi-view video coding, which will be investigated further.

7 ACKNOWLEDGEMENT

This work was supported in part by the NSFC-Guangdong Joint Foundation Key Project (U1201255), project of NSFC 61371138, China and a grant from the Core Research for Evolution Science and Technology (CREST), Japan Science and Technology Agency (JST), JAPAN.

REFERENCES

- [1] A. Vetro, "Frame Compatible Formats for 3D Video Distribution", in *Proc. Int. Conf. Image Process.*, Sept. 2010, pp.2405-2408.

- [2] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification", ITU-T Rec.H.264/ISO/IEC, 2003.
- [3] G. J. Sullivan, A. M. Tourapis, T. Yamakage, and C. S. Lim, "Draft AVC amendment text to specify Constrained Baseline profile, Stereo High profile, and frame packing SEI message", in Joint Video Team, Doc. JVT-AE204, London, UK, June 2009.
- [4] "Use cases and requirements for the enhancement of MPEG frame-compatible stereoscopic 3D video", ISO/IEC JTC1/SC29/WG11 MPEG 2010/N11681, Oct 2010, Guangzhou, China.
- [5] P. Yin, Y. Wang, G. J. Sullivan, "Editorial Improvement of Multi-resolution Frame Compatible Stereo Coding," JCT-3V, Doc. JCT3V-F0096, 6th meeting, Geneva, Nov. 2013.
- [6] "HDMI Licensing, LLC Announces Features of the Upcoming HDMI Specification Version 1.4", HDMI Licensing, LLC, 2009.
- [7] C.H. Tseng, H.M. Wang, and J.F. Yang, "Enhanced intra-4x4 mode decision for H.264/AVC coder," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 8, pp. 1027–1032, Aug. 2006.
- [8] Y.W. Huang, B.Y. Hsieh, T.C. Chen, and L.G. Chen, "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 3, pp. 378–401, Mar. 2005.
- [9] C.H. Hsia, J.S. Chiang, Y.H. Wang, T.Y. Teng, "Fast Intra Prediction Mode Decision Algorithm for H.264/AVC Standard", 3rd Conf. on IIHMSP, pp.535–538, 2007.
- [10] Y.Y. Lin, Y.M. Lee, C.D. Wu, "Efficient Algorithm for H.264/AVC Intra Frame Video Coding," Trans. IEEE, Circuits Syst. Video Technol., vol. 20, no.6, pp. 1367-1372, 2010.
- [11] J.C. Wang, J.F. Wang, J.F. Yang, and J.T. Chen, "A Fast Mode Decision Algorithm and its VLSI Design for H.264/AVC Intra-Prediction," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 10, pp. 1414–1422, Oct. 2007.
- [12] A.C. Tsai, A. Paul, J.C. Wang, and J.F. Wang, "Intensity gradient technique for efficient intra-prediction in H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 5, pp. 694–698, May 2008.
- [13] H. Li, K. Ngan, and Z. Wei, "Fast and efficient method for block edge classification and its application in H.264/AVC video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 6, pp. 756–768, Jun. 2008.
- [14] A.C. Tsai, J.F. Wang, etc al., "Effective Subblock-Based and Pixel-Based Fast Direction Detections for H.264 Intra Prediction", IEEE Trans. Circuits Syst. Video Technol., vol.18, No.7, pp. 975-982, July.2008.
- [15] H. Zeng, K.K. Ma, C. Cai, 'Hierarchical Intra Mode Decision for H.264/AVC', IEEE Trans. Circuits Syst. Video Technol., vol. 20, pp. 907-912, June. 2010.
- [16] K. Bharanitharan, B.D Liu, etc al., "A Low Complexity Detection of Discrete Cross Differences for Fast H.264/AVC Intra Prediction", IEEE Trans. Multimedia, vol.10, pp. 1250-1260, Oct. 2008.
- [17] F. Pan, X. Lin, etc al., "A Directional Field based Fast Intra Mode Decision Algorithm for H.264 Video Coding", in Proc. Int. Conf. Multimedia and Expo, vol. 2, pp. 1147-1150, 2004.
- [18] F. Pan, X. Lin, etc al., "Fast Mode Decision for Intra Prediction", ISO/IEC JTC1/SC29/WG 11 and ITU-T SG16 Q.6, JVT-G013.doc., 7th meeting Pattaya II, Thailand, Mar. 2003.
- [19] P. Yin, H.C.Tourapis, etc al., "Fast Mode Decision and Motion Estimation for JVT/H.264", in Proc. Int. Conf. Image Process., vol.2, pp. III-853-6, 2003
- [20] D.Wu, F. Pan, K.P.Lim, S.Wu, etc al., "Fast Inter mode Decision in H.264/AVC Video Coding", IEEE Trans. Circuits Syst. Video Technol., vol. 15, pp. 953-958, July. 2005.
- [21] A. C. Yu, "Efficient Block-size Selection Algorithm for Inter-frame Coding in H.264/MPEG-4 AVC", in Proc. Int. Conf. Acoustics, Speech and Signal Proces., vol. 3, pp. III-169-72, 2004.
- [22] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate distortion optimization for MPEG-4. Part 10: AVC/H.264," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 12, pp. 1557–1561, Dec. 2006.

- [23] C. Kim and C. J. Kuo, "Feature-based intra/inter coding mode selection for H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 4, pp. 441–453, Apr. 2007.
- [24] B. Kim, "Fast selective intra-mode search algorithm based on adaptive thresholding scheme for H.264/AVC encoding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 1, pp. 127–133, Jan. 2008.
- [25] D. S. Turaga, T. Chen, "Classification base Mode Decisions for Video over Networks", IEEE Trans. Multimedia, vol. 3, pp. 41-52. Mar. 2001.
- [26] J. You, W. Kim, J. Jeong, "16x16 Macroblock Partition Size Prediction for H.264 P Slices", IEEE Trans. on Consumer Electronics, vol. 4, pp. 1377-1383, Novm. 2006.
- [27] X. Li, G. Wu, "Fast Integer Pixel Motion Estimation", ISO/IEC JTC1/SC29/WG 11 and ITU-T SG16 Q.6, JVT-F011.doc, 6th meeting, Awaji, Island, JP, Dec. 2002.
- [28] Z. Chen, J. Xu, Y. He, J. Zheng, "Fast Integer-pel and Fractional-pel Motion Estimation for H.264/AVC", Journal of Visual Communication and Image Representation, vol. 17, no. 2, pp. 264-290, 2006.
- [29] D-K Kwon, M-Y Shen, C.-C. Jay Kuo, "Rate Control for H.264 Video With Enhanced Rate and Distortion Models", IEEE Trans. Circuits Syst. Video Technol., vol.17, no.5, pp.517-529, May. 2005.
- [30] Siwei Ma, Wen Gao, Yan Lu, "Rate-Distortion Analysis for H.264/AVC Video Coding and its Application to Rate Control", IEEE Trans. Circuits Syst. Video Technol., vol. 15, no.12, Dec. 2005.
- [31] Y. Sung and J. Wang, "Fast Mode Decision for H.264/AVC Based on Rate-Distortion Clustering," IEEE Trans. Multimedia, vol. 14, no. 3, pp. 693-702, June, 2012.
- [32] J. Wang and Y. Sung, "A Novel Fast Mode Decision Algorithm for H.264/AVC Using Particle Swarm Optimization," IEICE Tran. Fundamentals, vol. E96-A, no. 11, pp. 2154-2160, Nov. 2013.
- [33] Z. Zeng, X. Jin, S. Goto, "A Fast Intra Encoder of Frame-compatible Format Based on Content Similarity for 3D Distribution", in Proc. 19th European Signal Process. Conf., Aug. 2011, pp.1362-1366.
- [34] Z. Zeng, X. Jin, S. Goto, "A Fast Encoder of Frame-compatible Format based on Content Similarity for 3-D Delivery", in Proc. Visual Communications and Image Process., Nov. 2011, pp.1-4.
- [35] D. Tian, P. L. Lai, P. Lopez, C. Gomila, "View synthesis techniques for 3D video," in Proc. SPIE, vol. 7443, pp. 74430T-1-11, Sep. 2009.
- [36] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y.-K. Wang, "Joint Multiview Video Model (JMVM) 8.0," JVTAA207. doc, JVT of ISO/IEC MPEG & ITU-T VCEG 27th Meeting, Geneva, CH, 23-29 April, 2008.
- [37] T. Y. Kuo, C. K. Yeh, H. Y. Tsai, "A novel method for global disparity vector estimation in multiview video coding," in Proc. Int. Symposium Circuits Syst., pp. 864-867, 2009.
- [38] K. H. Liu, T. J. Liu, H. H. Liu, "A SIFT descriptor based method for global disparity vector estimation in multiview video coding," in Proc. Int. Conf. Multimedia and Expo, pp. 1214-1218, 2010.
- [39] L. Zhang, Y. Chen, M. Karczewicz, "3D-CE5.h related: Disparity vector derivation for multiview video and 3DV," m24937, ISO/IEC JTC1/SC29/WG11, 100th MPEG Meeting, Geneva, Switzerland, April-May 2012.
- [40] I. Feldmann, M. Mueller, etc al., "HHI Test Material for 3D Video", ISO/IEC JTC1/SC29/WG11 MPEG2008/M15413, April.2008. France.
- [41] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, K. Wegner "Poznań Multiview Video Test Sequences and Camera Parameters", ISO/IEC JTC1/SC29/WG11 MPEG 2009/M17050, Xian, China, Oct. 2009.
- [42] JM reference software 16.2, download at <http://iphone.hhi.de/suehring/>.
- [43] G. Bjontegaard, "Improvement of the BD-PSNR model", ITU-T SC16/Q6, 35th VCEG Meeting, Doc. VCEG-A111, Berlin, German, 2008.
- [44] G.M. Um, G. Bang, N. Hur, J. Kim and Y. S. Ho, "3D Video Test Material of Outdoor Scene," ISO/IEC JTC1/SC29/WG11 MPEG2008/m15371, April.2008. France.

- 1 [45] Y. S. Ho, E. K. Lee, C. Lee, "Multiview Video Sequence and Camera Parameters", ISO/IEC JTC1/SC29/WG11
2 MPEG2008/m15419, April.2008. France.
- 3 [46] Y. Su, A. Vetro, A. Smolic, "Common Test Conditions for Multiview Video Coding", ISO/IEC JTC1/SC29/WG11 and
4 ITU-T SG16 Q.6, JVT-T207.
- 5 [47] A. Vetro, et al, "Multiview Video Test Sequences from MERL", ISO/IEC JT1/SC29/WG11 MPEG05/m12077,
6 April.2005.Korea.
- 7 [48] C. Yeh, M. Li, M. Chen, M. Chi, X. Huang and H. Chi, "Fast Mode Decision Algorithm Through Inter-View Rate-Distortion
8 Prediction for Multiview Video Coding System," IEEE Trans. Industrial Informatics, vol. 10, no. 1, pp. 594-603, 2014.