International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.11 (2015), pp.141-158
http://dx.doi.org/10.14257/ijsip.2015.8.11.14

A Fast Inter Prediction Algorithm Based on Rate-Distortion Cost in HEVC

Jianfu Wang[1], Lanfang Dong[2] and Yinlong Xu[3]

[1]*School of Computer Science and Technology, University of Science and Technology of China*
[2]*School of Computer Science and Technology, University of Science and Technology of China*
[3]*School of Computer Science and Technology, University of Science and Technology of China*
[2]*lfdong@ustc.edu.cn*

Abstract

As one of the most important video compression technologies, inter prediction coding is highly efficient in reducing the temporal redundancy of video sequence. However, complicated inter prediction for the latest High Efficiency Video Coding standard (HEVC) brings high computational complexity and seriously restricts the encoding speed. In this paper, a fast inter prediction algorithm based on Rate-Distortion (RD) cost is proposed to improve inter prediction of HEVC. First, the splitting of Largest Coding Unit (LCU) is determined according to the RD costs with best Coding Unit (CU) size being 64x64 in the reference picture. Then, for other CUs in lower depths, the comparable RD costs are selected from encoded CUs in the same depth at the same Coding Tree Unit (CTU) based on the local homogeneity in spatial domain. By comparing the RD cost of current CU with its corresponding RD threshold, the splitting is terminated in advance. In this way, the proposed fast inter prediction algorithm can avoid the traversal of all CUs in the coding tree structure and improve the encoding speed. Experimental results show that the algorithm can save about 30% encoding time on the basis of ensuring visual quality and compression ratio of videos. Therefore, the computational complexity can be reduced greatly.

Keywords: *HEVC, Inter Prediction, Rate-Distortion, CU partition*

1. Introduction

Represented as a series of still images with strong correlation, the raw videos are of tremendous amount of data and cannot be applied in most video services directly so that only after compression, the video data can be effectively stored and transmitted. The video compression is using modern coding techniques to reduce redundancy and then using the corresponding decoders to restore the raw video. Nowadays, the latest video coding standard HEVC was developed by the Joint Collaborative Team on Video Coding (JCT-VC) composed of ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The technical content of HEVC was finalized in January 2013 and the specification was formally ratified as a standard in April 2013 [1]. The emerging of HEVC conforms to the development tendency of digital videos towards high definition, high frame rate and high compression ratio.

The neighboring frames in one video contain high similarity and this correlation in sequence is the so-called temporal redundancy which is removed by inter prediction in HEVC. Briefly, with using the encoded frame as reference picture, the inter prediction searches the matching area from it for the block in current frame. The matching area is

ISSN: 2005-4254 IJSIP

set as prediction signals that are subtracted from current signals to get the residuals. Compared with the previous standards, the prediction of HEVC is of more prediction modes and more complex prediction units (PU) [2], which bring both high compression ratio and huge computational complexity. According to experiments, HEVC can double the compression ratio compared to H.264 with nearly tripled computational complexity. Of the whole encoding, the processing time occupied by inter prediction is up to 96% [3] so that how to reduce the calculation of inter prediction becomes one focus of research in HEVC.

According to the statistics of RD costs in reference picture and current CTU, we get the thresholds for CUs in different depths and use threshold comparison to eliminate the unnecessary CU splitting. In this way, the fast inter prediction can speed up the encoding. The results of the experiment validate the feasibility and effectiveness of the proposed method.

The remainder of this paper is organized as follows. Section 2 presents related work for HEVC and Inter coding. Section 3 describes the proposed fast inter prediction method. Experiments are performed in Section 4, while Section 5 concludes this paper.

2. Background Knowledge

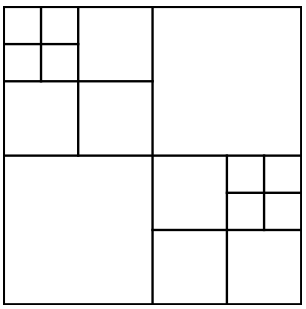
2.1. Overview of Inter Coding in HEVC

Inter prediction makes the best of temporal correlation to improve compression ratio. To minimize the temporal redundancy, HEVC searches the best matching area from reference picture for current block through motion estimation and motion compensation, and then determines the optimal coding mode by comparing the RD costs under different CU levels and PU modes. The rough definition of RD cost [4] is as (1):

$$RD\cos t(B) = D(B) + \lambda \cdot R(B) \tag{1}$$

B is current block, D is the distortion specifying the average loss between current CU and its matching block, λ is the Lagrange multiplier and R is the bit cost to be considered for CU size and mode decision.

Complicated coding structure is one cause of computational complexity in inter prediction. In HEVC, each frame is divided into basic block unit-CTU, and CTU can be further partitioned into smaller CUs in a recursive quadtree structure. The max depth of quadtree is 4 of which the root node is LCU with block size being 64x64. And the sizes of CUs in depths from 1 to 3 are 32x32, 16x16 and 8x8 respectively. Figure 1 shows one sample of CTU partition and its quadtree structure [5].



(a)

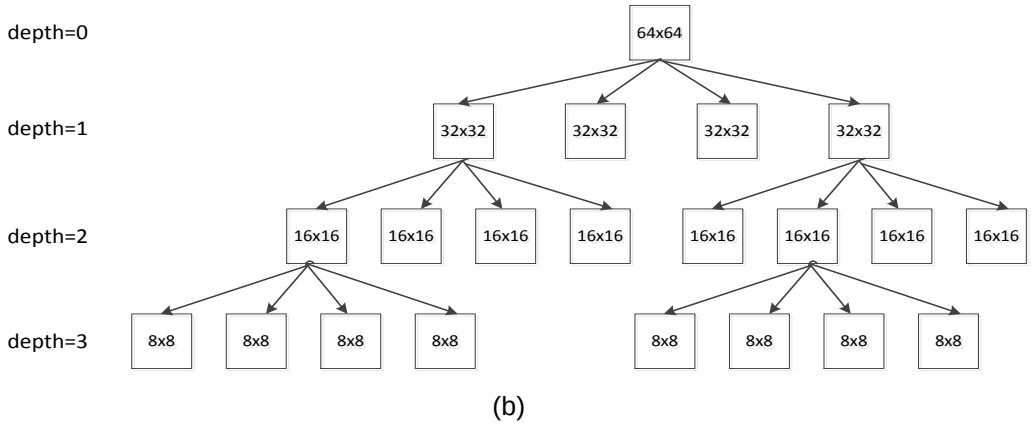


Figure 1. CU Structure in HEVC (a) One Sample of CTU (b) The Corresponding Quadtree

As the basic unit for coding, CU should be split into PUs as the basic unit for the intra/inter prediction. The PU modes for inter prediction are shown in Figure 2, including asymmetric partition and symmetric partition. One important task for inter prediction in HEVC is to select the best PU mode for each CU with RD cost comparison.

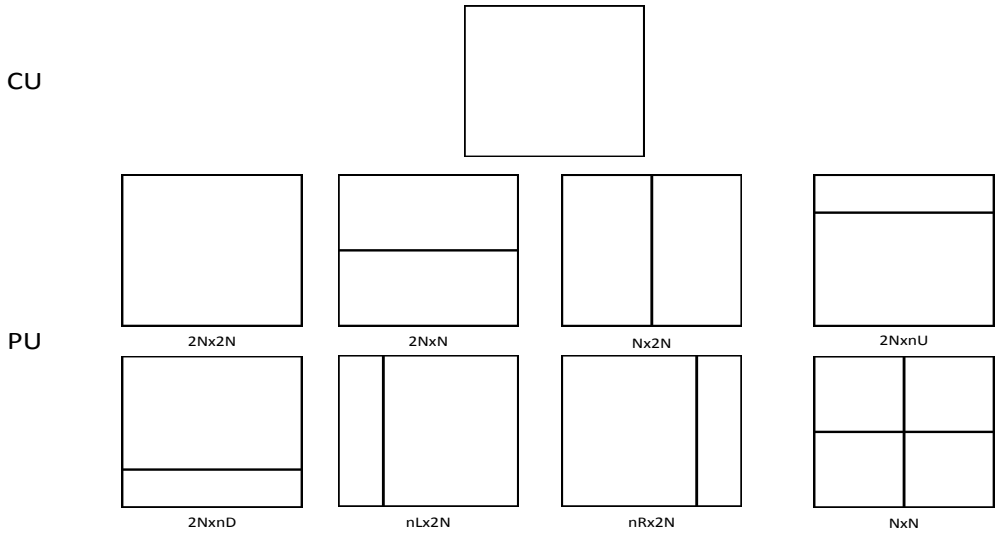


Figure 2. CU and its Corresponding PU Modes in Inter Prediction of HEVC

Another cause for the tremendous computation of inter prediction is the complex selection of optimal coding mode. The inter prediction in HEVC starts from root node LCU, and traverses all nodes of the quadtree in depth-first order. For every node (*i.e.*, CU), all PU modes of 2Nx2N, 2NxN, Nx2N, 2NxnU, 2NxnD, nLx2N, nRx2N and NxN (only used in CU with 8x8 size) are processed through motion estimation and motion compensation, and the PU mode with the minimal RD cost is recorded for every CU. When returning from the lower depth to upper depth, the optimal CU partition and PU modes of current node are achieved by comparing the RD cost of current CU and the total cost of its sub-CUs in lower depth. When reaching to the root node, the optimal coding parameters of CTU are achieved.

From the process, we can see that all PU modes of each CU in every depth require traversing. Therefore, the complex structures and mode selection lead to huge computation.

2.2. Related Work

Inter prediction can improve the compression ratio of videos effectively, which has been the emphasis of research since the emerging of HEVC. Nowadays, the recent researches focus on two aspects: one is to improve the compression ratio by reducing the bitrate. The paper [6] proposed a new block-adaptive skip mode based on higher-order parametric motion models for inter prediction with sophisticated parametric motion estimation. The new mode relied on the accurate description of camera motion while feature correspondences used for parametric motion model estimation referred to complex zoom, rotation, and perspective transformation. Paper [7] introduced the motion vector coding techniques for HEVC and proposed three coding tools for the motion vector predictor in the Inter, Skip and Merge modes to achieve average1.5% bitrate reductions. T. Michael *et. al.,* presented a novel parametric motion vector predictor based on higher-order motion models in paper [8]. The method could predict complex motion as rotation and zoom efficiently. However, with about 2% reduction of bitrate, the increase of encoding time was more obvious by complex parametric motion estimation. In papers [9-10], S. Oudin *et. al.,* made a special study on quadtree-based partitioning and introduced a simple but efficient block merging algorithm which generated a single motion parameter set for a whole region of contiguous motion-compensated blocks with BD rate saving from 6% to 8%. B. Philippe, F. Edouard and T. Dominique [11] proposed a fast encoding algorithm for geometry-adaptive block partitioning which used non-horizontal or non-vertical line to split a block. The method brought little increase of compression ratio but high increase of encoding time. Different from other methods, the paper [12] aimed at compound videos and proposed three approaches to exploit inter-frame correlations based on base color, index map and scalar quantization. Similarly, to save the bitrate of screen videos, M. Naccari *et. al.,* [13] proposed a Residual Differential Pulse Code Modulation (RDPCM) applied to inter predicted residuals by exploiting the spatial correlation present in blocks containing edges or text areas. With applying the inter RDPCM to CU, PU and Transform Unit (TU) and introducing two additional tools: Prediction Chunking and Hierarchical Prediction, this method achieved up to 8% average bitrate reduction. In general, such methods with aiming to the reduction of bitrate will bring high increase of computational complexity.

The other focus is on reducing the computation and saving encoding time. Such algorithms improved the performance mainly at PU level with optimizing the selection of prediction modes or speeding up motion estimation. For example, in paper [14], J. Kim *et. al,* proposed an efficient bi-prediction algorithm by finding the favorable condition of bi-prediction with comparing to forward and backward prediction to reduce encoding time of HEVC. However, the improvement of coding speed was not evident for uni-prediction. Another fast inter mode decision method was proposed in [15] by simplifying the inter PU mode decision process with saving 46.5% encoding time. On the CU level, when all PU modes shared the same motion information with 2Nx2N mode, the division for CU was terminated in early which led to 1.1% coding efficiency loss. By jointly using the inter-level correlation of quadtree structure and the spatiotemporal correlation, L. Shen *et. al.,* in paper [16] proposed a fast inter-mode decision algorithm for HEVC. Considering the prediction mode distribution at each depth level and the coding information correlation among the adjacent CUs, this paper proposed early Skip mode decision, prediction size correlation-based mode decision and RD cost correlation-based mode decision, and about 50% computational complexity on average was saved. Paper [3] presented a novel fast heuristic decision for motion vectors merging. The algorithm avoided several motion estimation calls during the inter prediction and reduced about 34% execution time in the overall encoding process but led to 2% increase of bitrate. Another fast motion estimation algorithm for HEVC was proposed by P. Nalluri, L. N. Alves and A. Navarro in paper [17] which used rotating hexagonal grid searching and adaptive threshold for early termination to save the time of motion estimation. The

algorithm can also be suitable for encoders of other codec standards like H.264. In paper [18], the authors introduced a mothed by checking the neighbor LCU edge motion vector and comparing the motion vector of current 2Nx2N CU with certain thresholds to terminate any further RD calculations for the current CU.

Another way to improve the speed of inter prediction is terminating CU splitting at CU level. A fast coding algorithm based on adaptive coding depth range selection for HEVC was presented in [5] which employed the mode information of the current CU and depth range selection mechanism to avoid unnecessary CU splitting that saved the encoding time significantly. The paper [19] proposed content based hierarchical fast CU decision algorithm with analyzing the utilization rate of CU in all depths on frame level and CU level respectively to skip several rarely used CUs in specified depth and reduce the computational complexity. In paper [20], an adaptive CU early termination algorithm was proposed with taking use of the average RD cost of previous skipped CUs to avoid the splitting of some CUs. More than 1.0% bitrate increase was brought by this algorithm. The paper [21] used Bayesian decision rule with collecting relevant and computational-friendly features to make a precise and fast selection on CU size that greatly reduced the complexity of HEVC while suffered from 2.0% loss on RD performance. A fast Pyramid Motion Divergence (PMD) based CU selection algorithm in paper [22] was presented for inter prediction. First, the PMD features were calculated with estimated optical flow of the down sampled frames. Then, a nearest neighboring like method was used to determine the CU splitting. The computational complexity was reduced significantly but the loss in bitrate should be decreased further.

Significant improvement in the reduction of complexity was achieved by the above algorithms, but the loss of compression ratio cannot be neglected. In this paper, under the guarantee of compression ratio and visual quality of reconstructed videos, we propose a fast inter prediction algorithm to improve the speed of encoding and reduce the computational complexity. The algorithm is realized at CU level, and once the splitting of CU is early terminated, the calculation of partition and selection of PU modes for all the other CUs in lower depths are avoided. The presented algorithm is easy and implemented with simple threshold comparison which can save the encoding time evidently.

3. Fast Inter Prediction Based on RD Cost

The presented scheme for inter prediction is implemented based on the HM14.0 by combining the RD costs of reference picture and current CTU to reduce the computational complexity and improve the encoding speed.

3.1. Analysis of RD Cost

RD cost is the determining factor for the selection of CU size. When LCU is chosen as optimal size, the distortion for current area is close to that encoded with small CUs but coding bits (R in formula (1)) are fewer. In this case, the whole RD cost of LCU is lower. On the contrary, the coding bits of image areas choosing small CUs as optimum may increase but the distortion must be reduced more effectively. By contrast, RD cost of using small CUs for these areas is lower than using a large CU. In a word, there is a close relationship among the CU sizes, RD costs and image contents.

The high similarity of neighboring images is exploited by inter prediction to remove the temporal redundancy of videos. In general, for the image areas of small change in the time domain, the distortion and coding bits are usually small when encoded with large CU. That is, the RD cost in this condition is low. On the other hand, with small CU, the distortion may decrease partly, but the increase of coding bits is likely to lead to higher RD cost compared with large CU. Therefore, the areas with small change in neighboring images are usually encoded with large CU, such as LCU. Therefore, this paper uses the RD cost information of areas encoded with LCU in the reference picture to assess the RD

cost produced during inter prediction of current frame. When the RD cost is low enough, the corresponding image areas are determined to be encoded with LCU without further splitting to save encoding time.

The image areas choosing LCU as the optimal size are usually of small change between video sequences and low RD costs on the whole. But for the areas of high fluctuations in the neighboring frames, both the distortion and RD cost are high when encoded with large CU. In this case, the CUs except LCU (Non-LCU) are more suitable. However, the Non-LCU contains multiple CU levels, which are 32x32, 16x16 and 8x8, and the range of RD costs for Non-LCU is wide on the whole. Because of local homogeneity, high similarity exists in the adjacent regions. As the increase of space distance, the similarity will drop. That is, in certain space scope, the Non-LCUs are of high correlation and close RD costs. For example, the coding information of the first 32x32 CU in one CTU could be used to decide whether its neighboring 32x32 CU is split or not. Therefore, for the Non-LCUs, we take CTU as the basic processing unit and the RD costs of encoded CUs are used to determine the partition of current CU while there are no dependencies in different CTUs.

3.2. Analysis of Proposed Algorithm

Based on the analysis of RD cost, this paper first divided the CUs into LCU and Non-LCU. The splitting of LCU is determined according to the RD costs of reference picture while the splitting of Non-LCU is determined with exploiting the spatial information in one CTU. In detail, we adopt the following different strategies to deal with LCU and Non-LCU respectively.

3.2.1. Strategy for LCU: Because inter prediction starts from the root node LCU to traverse the quadtree, the early determination of LCU as the optimal coding unit will avoid all CUs in lower depths and reduce the computational burden evidently.

For LCU, with the statistics of RD costs for CUs with 64x64 as the best size in reference picture, we get one threshold T_64. Based on the comparison result of T_64 and the RD cost of current LCU, whether or not the LCU should be split will be determined. When the RD cost is no more than T_64, the splitting and inter prediction will be terminated in early, otherwise, the strategy for Non-LCU needs to be considered. Figure 3 shows the flowchart of strategy for LCU.

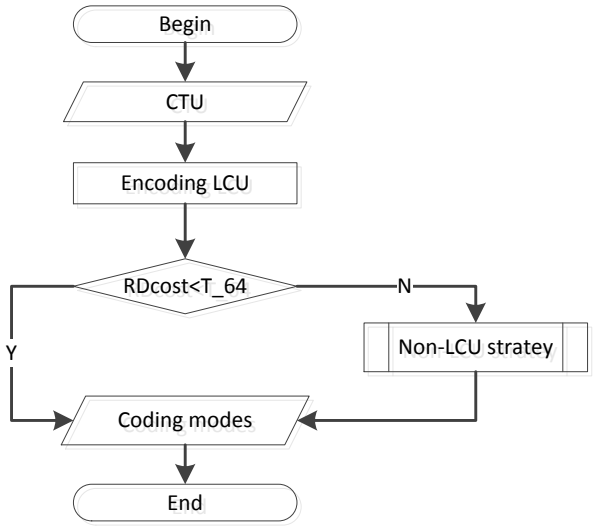


Figure 3. The Strategy for LCU

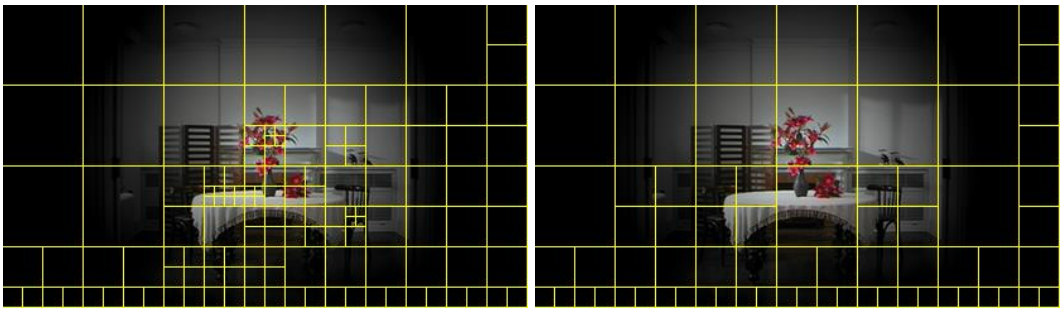
The following is the calculation of threshold T_64.

1. Statistic and sorting: record RD cost $Cost_i$ for CU with best coding size being 64x64 in reference picture, sort $Cost_i$ in the order ascendingly and get $CostInOrder_i$ ($1 \le i \le M$, M is the number of LCUs).
2. Interval division: divide $CostInOrder_i$ into 3 intervals and select the elements of middle interval $[CostInOrder_{N1+1}, CostInOrder_{N2}]$ as parameters to compute threshold where $N_1 = \lfloor 2 * M / 5 \rfloor$, $N_2 = \lfloor 3 * M / 5 \rfloor$.
3. Obtaining threshold: calculate the mean value of the elements in middle interval and obtain the threshold according to equation (2):

$$T\_64 = \lambda \frac{\sum_{i=N_1+1}^{i=N_2} CostInOrder_i}{N_2 - N_1} \qquad (2)$$

$$\lambda = \frac{QP2}{QP1}$$

In the equation, *QP1* and *QP2* are the Quantization Parameters (QP) of the reference picture and current picture respectively. Generally, the smaller QP brings higher visual quality and finer partition of CTU. In other words, for the same encoded picture, the number of 64x64 CUs with smaller QP is less than larger QP as well as the mean value of corresponding RD costs. As Figure 4 shows, compared to encoded image with *QP* being 35, the CU partition is finer with *QP*=29.



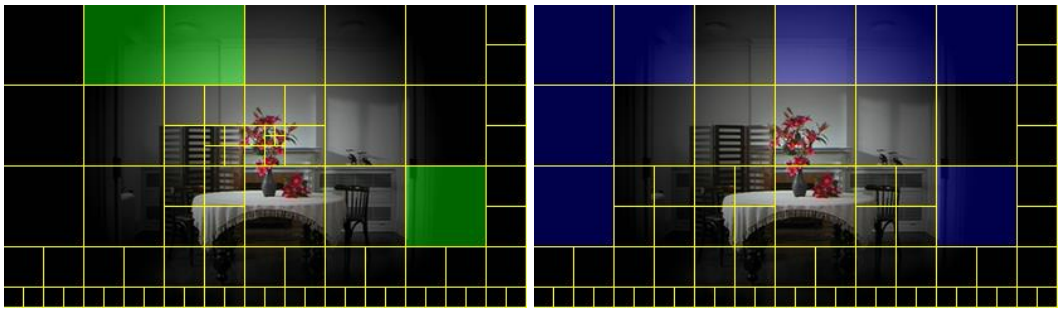
(a) (b)

Figure 4. CU Partition of One Image (a) QP=29 (b) QP=35

The value of elements in the first interval is relatively small. If they are referenced as parameters to compute the threshold for inter prediction, the achieved threshold is small that will lead to partition of most LCUs. It means that only few LCUs with RD costs less than *T_64*. Similarly, the elements in the third interval are always with high values, by using which the achieved high threshold will lead to early termination of most LCUs. Compared to the HM14.0, the accuracy of CTU partition is low which shall bring high loss in compression ratio and visual quality. From experiments, it is found that about 30-50% LCUs can be early terminated of which the accurate partition is up to 90% using the elements in the middle interval. For the other LCUs with failed termination, the correct partition of CTU also can be determined in the subsequent comparison with its sub-CUs.

The early determination of CTUs with coding size being 64x64 can avoid traversing lower depths of the quadtree which effectively reduce the computational expense. Consequently, it is necessary and important to terminate the partition of LCUs. Figure 5 is an example of strategy for LCU. The areas in green of Figure 5 (a) are the LCUs in

reference picture used to calculate the threshold while the blue areas in Figure 5 (b) are the LCUs whose splitting can be terminated in advance according to the proposed LCU strategy. Compared with the results of HM14.0 in Figure 4 (b), the partition of images has no difference.

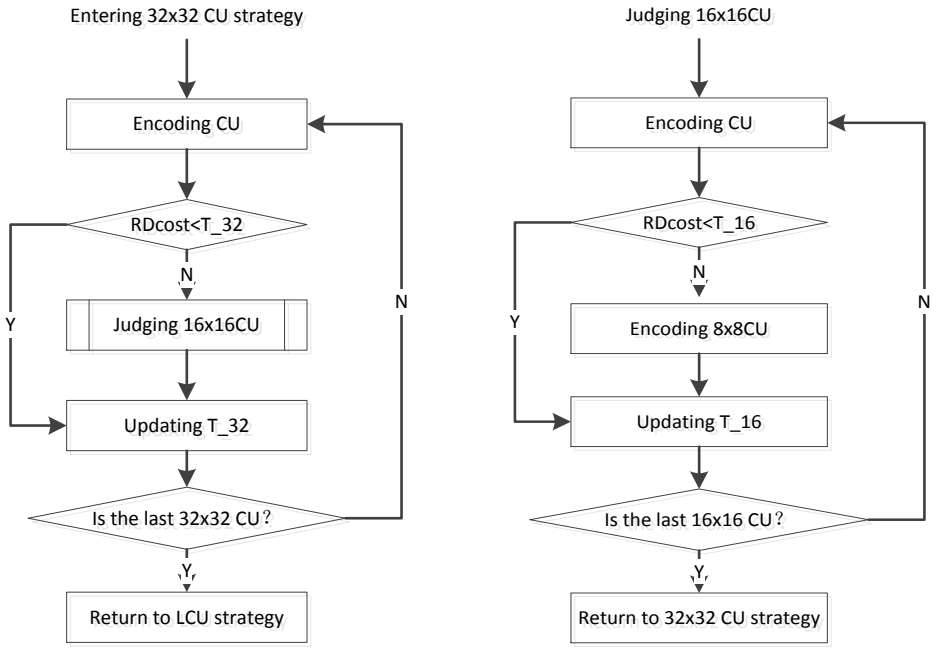


(a) (b)

Figure 5. CU sizes in (a) reference picture (b) current frame Green areas: LCUs used to calculate T_64; Blue areas: the LCUs with early splitting termination

3.2.2. Strategy for Non-LCU: When the RD cost of current LCU is more than the threshold T_64, it indicates that the distortion of CTU using 64x64 as coding size is high and the judgments of CUs in lower depths are necessary. In this case, the partition of LCU into smaller CUs can ensure the quality of pictures in video coding.

Different from LCUs using the reference picture as a criterion, the Non-LCUs use CTU as the basic processing unit and utilize the local spatial information of images. Within the traversal of quadtree, the minimal RD cost of each node is recorded and used to compute the thresholds *T_32* or *T_16* for the determination of the partition of 32x32 CU or 16x16 CU. At the 32x32 CU level, if the corresponding RD cost is less than *T_32*, the splitting should be terminated ahead. Otherwise, it goes to 16x16 CU level and the calculated RD cost of 16x16 CU is compared with *T_16*. Based on the result of comparison, this splitting of 16x16 CU is decided. The process is shown in Figure 6. 8x8 is the smallest allowed CU size and cannot be further split so that the judgments at 8x8 CU level are avoided.



(a) (b)

Figure 6. The Strategy for Non-LCU (a) Judgments at 32x32 CU Level (b) Judgments at 16x16 CU Level

The recorded RD cost of one CU in certain depth is achieved through comparison with the costs of its sub-CUs in lower depths. Based on the recorded RD cost, the thresholds *T_32* and *T_16* for early termination are computed as follows.

1. Initialization: set $T\_32 = 0$ as the threshold for CU with 32x32 size and set $T\_16 = 0$ for CU with 16x16 size.
2. Threshold comparison: get the RD cost of current CU without further partition. If the depth is 1 and its RD cost $Cost32_i$ is less than T_32, or if the depth is 2 and its RD cost $Cost16_j$ is less than T_16, the partition of current CU is early terminated. Or, further partition is necessary.
3. Threshold updating: when returning back from the low level of quadtree, if the CU with depth being 1 in the quadtree, its RD cost $Cost32_i$ is recorded, $1 \le i \le 3$. If the CU with depth being 2, its RD cost $Cost16_j$ is recorded, $1 \le j \le 15$. Then the following equations are used to update the thresholds.

$$T\_32 = \lambda_1 \frac{\sum_{k=1}^{k=i} Cost32_k}{i} \quad 0 < \lambda_1 < 1 \qquad (a)$$

$$T\_16 = \lambda_2 \frac{\sum_{k=1}^{k=j} Cost16_k}{j} \quad 0 < \lambda_2 < 1 \qquad (b) \qquad (3)$$

   In the equations, λ_1 and λ_2 are coefficient constants. To prevent wrong termination for CU partition, the value of λ_1 and λ_2 should not be too large. In the algorithm, we set $\lambda_1 = 0.6$ and $\lambda_2 = 0.7$.
4. Iteration: return to Step 2, and process the subsequent CU based on the thresholds after updating.

Using the 32x32 CUs as an example as shown in Figure 7, we set *T_32*=0 when starting to process CU1. The threshold *T_32* is updated to $T\_32=\lambda_1*Cost32_1$ after completing CU1. Likewise, when finishing CU2 and CU3, the *T_32* is updated to $\lambda_1*(Cost32_1+Cost32_2)/2$ and $\lambda_1*(Cost32_1+Cost32_2+Cost32_3)/3$ respectively. By that analogy, the threshold *T_16* for 16x16 CUs is achieved and updated.

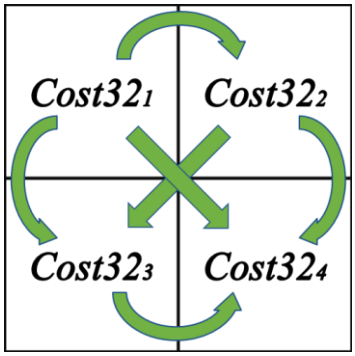
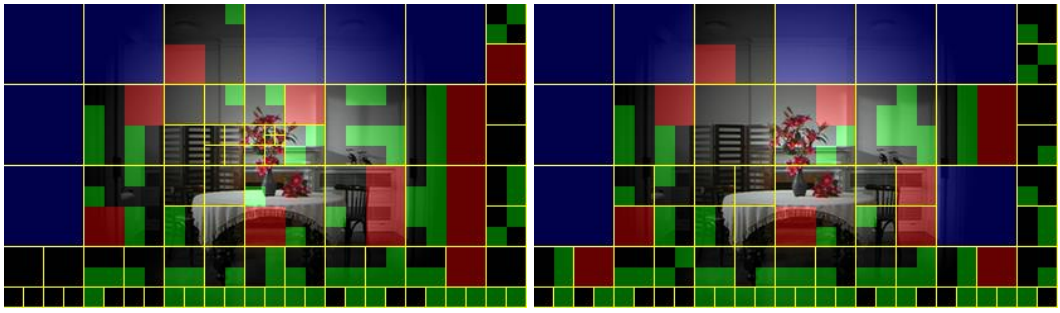


Figure 7. The Relationship of 32x32 CUs in the CTU

Figure 8 shows the achieved CU sizes for Figure 5 (a) and (b) after Non-LCU strategy. The blue areas are determined in LCU level. The splitting of red areas is terminated in early at 32x32 CU level while the early terminated areas marked with green color are for 16x16 CUs. From the figures, we can see that the CU splitting in large image areas can be terminated at LCU level and Non-LCU level in early.



(a) (b)

Figure 8. CU Sizes and the Early Determined Areas in (a) Reference Picture (b) Current Frame. Blue: LCUs; Red: 32x32 CUs; Green: 16x16 CUs

3.3. Flowchart of the Algorithm

Based on the above analysis and the flowcharts in Figure 3 and Figure 6, we get the improved inter prediction algorithm as follows:

Algorithm: Fast inter prediction algorithm based RD cost
Initialization: $T\_64=0$, $T\_32=0$, and $T\_16=0$
Input: $Cost64_i$ of the reference picture and Current frame
Output: Coding modes of Current frame
Process: Compress Picture

1. Sort the recorded $Cost64_i$ of reference picture and set them as $CostInOrder_i$, $1 \le i \le M$;

2. Divide $CostInOrder_i$ into three intervals and select the elements in $[CostInOrder_{N1+1}, CostInOrder_{N2}]$ as parameters to compute T_64;
3. Set *i*=0;divide the picture into CTUs;
4. For each CTU in current picture, set *j*=1, *k*=1, *depth*=0 , $T\_32=0$ and $T\_16=0$;
   - 4.1 If *depth*<=3, apply inter prediction for current CU in current depth and get the temporary RD cost $Cost\_T$;
     - A. If *depth*=0&&$Cost\_T < T\_64$,set $Cost\_T$ as $Cost$ and goto 4.5, or goto 4.2;
     - B. If *depth*=1&&$Cost\_T < T\_32$, goto 4.3, or goto 4.2;
     - C. If *depth*=2&&$Cost\_T < T\_16$, goto 4.3, or goto 4.2;
     - D. If *depth*=3, goto 4.3;
   - 4.2 If current depth<3, split the CU into four subCUs, *depth*++;
     - A. For $sub\text{-}CU_t$ ($1 \le t \le 4$), goto 4.1;
   - 4.3 With the comparison of current CU and its subCUs in lower depths, get the optimal RD cost $Cost$ for current CU.
     - A. If *depth*=0 && best size=64x64, goto 4.5; or if best size is not 64x64, goto 5;
     - B. If *depth*=1, set $Cost$ as $Cost32_j$,and update T_32 , *j*++, goto 4.4;
     - C. If *depth*=2, set $Cost$ as $Cost16_k$, and update T_16 , *k*++, goto 4.4;
     - D. If *depth*=3, save $Cost$, goto 4.4;
   - 4.4 If current CU is $subCU_4$, *depth*--, goto 4.3; or continue the following subCU,*t*++, goto 4.2A;
   - 4.5 *i*++, set $Cost$ as $Cost64_i$ and save it, goto 5;
5. If all CTUs are finished, goto 6; or goto 4;
6. End CompressPicture;

The $Cost\_T$ is different from $Cost$, where $Cost\_T$ is the RD cost of CU in one certain depth without further partition while $Cost$ is achieved by comparing the RD cost of the CU with its subCUs in the lower depths when returning from the leaf nodes. When $Cost\_T$ is less than the threshold, the value of $Cost\_T$ equals to $Cost$.

It is worth noting that the encoding of video sequences starts from intra coding, so that the T_64 of the first frame and the frames with reference picture using intra coding is 0. That is to say, there is no $Cost64_i$ recorded in intra frames. Therefore, the strategy for LCU takes effect in the frames using inter coding as well as their reference pictures.

It also should be pointed out that the partition of CU is determined after the motion estimation and motion compensation for all PU modes of current CU. Therefore, the final selected CUs in CTU are the results of traversing the corresponding CU and its father nodes.

4. Experiments and Results

4.1. Experimental Conditions

To validate the proposed fast inter prediction algorithm, we carried on the experiments under the following conditions.

Operating System: Windows 7;

Processor: Intel Core(TM) i3-2100 3.10GHz, 3.49GB RAM;
Development environments: Microsoft Visual Studio 2010;
Test Model: high efficiency video coding test model 14.0 (HM14.0);
Configuration: encoder_lowdelay_P_main;
Quantization parameter: QPs in one GOP with size being 4 are 35, 34, 35 and 33;
Tested videos: 30 different videos with different resolution;
Metrics: Peak Signal to Noise Ratio (PSNR), Encoding Time (ET) and Bitrate;
The PSNR is the reflection of the visual quality while Bitrate reflects the compression ratio. The computational complexity is measured by ET.

4.2. Experiments Results

We compared the proposed fast inter prediction algorithm with HM14.0 under the same parameters, and the results were shown in Table 1. The value Δ in the table is calculated through (4), where *proposed* is for the proposed algorithm while *HM* is for the HM14.0.

$$\Delta PSNR = \frac{PSNR_{proposed} - PSNR_{HM}}{PSNR_{HM}} \times 100\%$$

$$\Delta Bitrate = \frac{Bitrate_{proposed} - Bitrate_{HM}}{Bitrate_{HM}} \times 100\% \quad (4)$$

$$\Delta ET = \frac{ET_{proposed} - ET_{HM}}{ET_{HM}} \times 100\%$$

Table 1. The Comparison Results of Fast Inter Prediction Algorithm and HM14.0

Video sequences		Performance Evaluation (%)				
Resolution	Video name	ΔY_PSNR	ΔU_PSNR	ΔV_PSNR	ΔBitrate	ΔET
704x576	CITY	-0.22	0.13	0.15	0.25	-32.12
	CREW	-0.12	-0.19	-0.05	0.01	-22.03
	HARBOUR	-0.27	0.06	0.14	1.36	-26.17
	ICE	-0.11	0.09	-0.17	0.11	-30.65
	SOCCER	-0.01	0.14	0.04	0.02	-24.22
720x576	Parkrun	-0.20	0.08	0.09	0.48	-16.27
	Shields	-0.33	-0.03	0.01	-0.66	-32.26
	Stockholm	-0.17	-0.02	0.04	0.19	-29.14
	Mobcal	-0.34	-0.03	0.18	0.21	-30.32
832x480	Flowervase	-0.12	0.05	0.05	-0.05	-34.65
	BasketballDrillTex	-0.09	-0.07	-0.03	1.06	-25.48
	BasketballDrill	-0.07	-0.11	-0.18	0.71	-25.30
	BQMall	-0.07	-0.06	-0.00	0.18	-22.00
	Keiba	-0.10	-0.06	-0.22	0.10	-26.73
	Mobisode2	-0.08	-0.16	-0.22	0.68	-40.36
	PartyScene	-0.18	0.02	-0.00	0.90	-19.25
	RaceHorses	-0.18	0.05	-0.02	2.62	-24.28
1024x768	ChinaSpeed	-0.28	-0.07	-0.03	0.21	-31.31
1280x720	Johnny	-0.06	-0.04	-0.03	-0.31	-43.31

	KristenAndSara	-0.02	-0.02	-0.08	-0.18	-37.65
	vidyo1	-0.04	-0.03	-0.04	-0.04	-45.64
	FourPeople	-0.05	0.00	0.00	0.07	-37.86
1920x1080	ParkScene	-0.15	0.02	0.02	0.56	-30.05
	Kimono1	-0.10	-0.07	-0.01	0.52	-28.00
	BasketballDrive	-0.03	-0.05	-0.03	0.67	-25.27
	BQTerrace	-0.09	-0.01	0.02	-0.76	-32.64
	Tennis	-0.06	0.10	-0.02	0.29	-33.05
	Cactus	-0.02	0.02	-0.01	0.41	-25.20
2560x1600	PeopleOnStreet	-0.11	-0.06	-0.01	0.83	-14.43
	Traffic	-0.07	-0.02	-0.02	0.57	-33.20

From the data in the table we can see that: 1) Compared with HM14.0, the average loss of PSNR for the proposed fast inter prediction algorithm is about 0.12%. Such a small loss cannot damage the visual quality of the reconstructed videos. 2) The increase of bitrate of proposed algorithm is about 0.36%. Without evident increase of bitrate, high compression ratio can be ensured. 3) The improved algorithm brings significant time saving. An average 30% drop in encoding time means that the computational complexity is reduced evidently.

Figure 9 shows the comparison examples of proposed fast inter algorithm and HM14.0. For the same image, the yellow lines mark the consistent partition while in the inconsistent areas, the red lines are for HM14.0 and green lines are for the proposed algorithm. From the comparison of the CU sizes, we can see that the proposed algorithm can achieve very accurate CU sizes in most areas of the image. Even for the inconsistent areas, the difference in CU level between HM14.0 and the proposed algorithm is no more than 1.

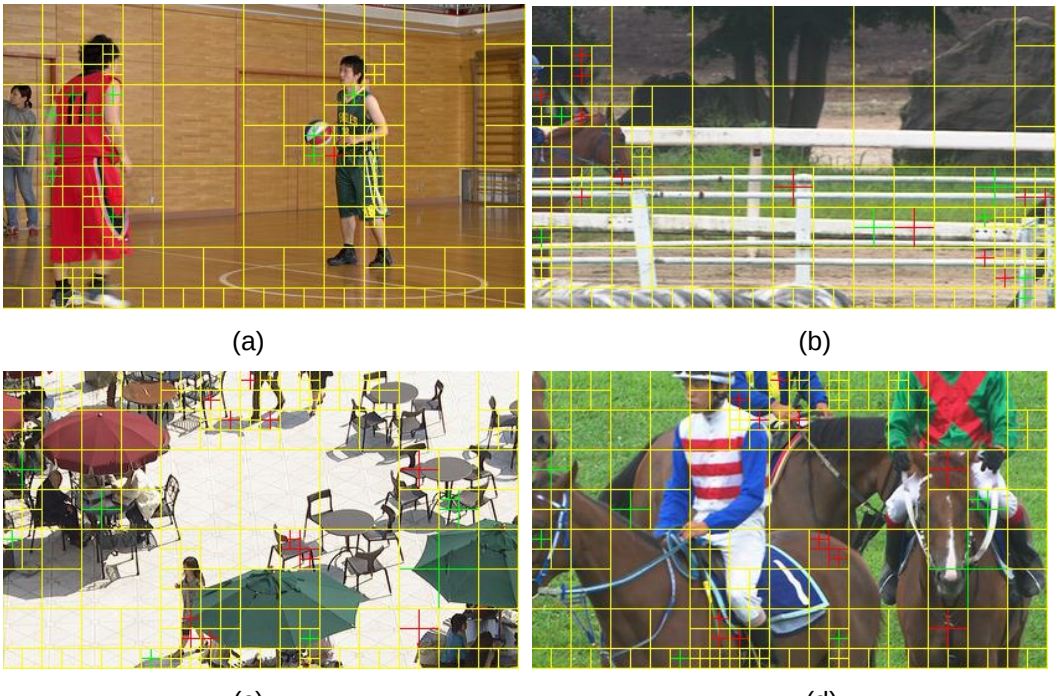
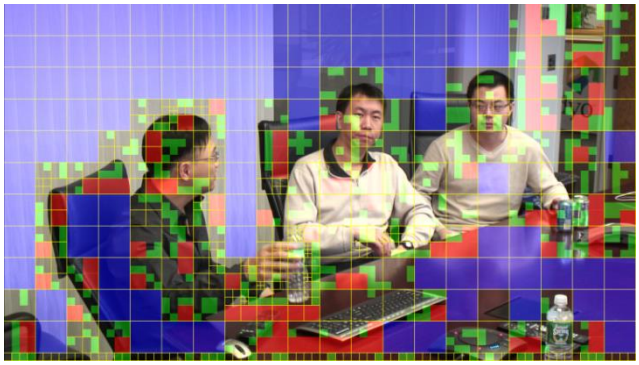


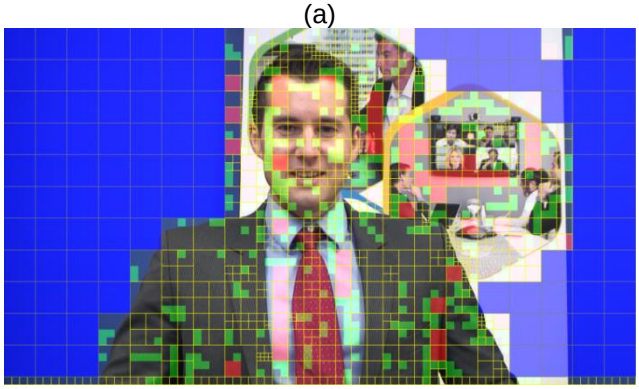
Figure 9. Comparison of CU Splitting (a) Basketball Pass (b) Keiba (c) BQ Square (d) Race Horses. Red Lines: HM14.0; Green Lines: Proposed Algorithm

4.3. Discussion

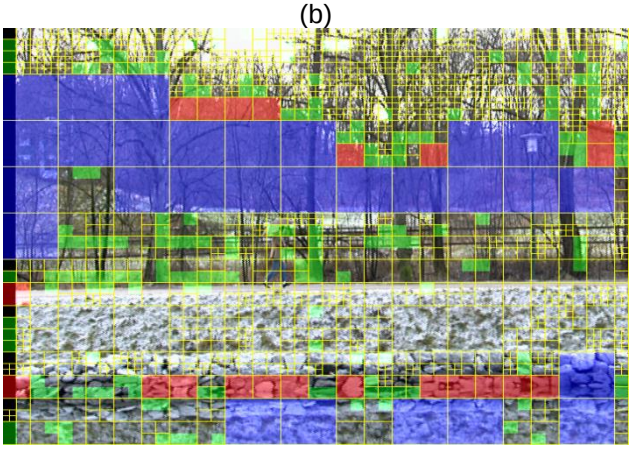
From the results in Table 1, we can see that there is a big difference between the tested videos in the reduction of encoding time. Generally, the improvement of encoding speed is obvious for the videos with simple scenes or small change, such as Flowervase and Mobisode2, and also for surveillance videos and conference videos with fixed backgrounds, such as Johnny and vidyo1. As Figure 10 (a) and (b) show, the CU sizes in most image areas of such videos can be determined in advance at LCU level or Non-LCU level by the proposed algorithm. However, the improvement of encoding time for videos with complex scenes and serious change is not so satisfactory, such as Parkrun, PartyScene and PeopleOnStreet. As Figure 10 (c) and (d) show, the optimal coding mode is determined by judging multiple CU levels for most areas in this kind of videos, so that it is not significant.



(a)



(b)



(c)

(d)

Figure 10. CU Sizes and the Early Determined Areas in (a) Vidyo1 (b) Johnny (c) Parkrun (d) PartyScene. Blue: LCUs; Red: 32x32 CUs; Green: 16x16 CUs

From the experimental results it can be seen that the proposed fast inter prediction algorithm based on RD cost provides good video quality at substantial improvement of encoding speed and negligible increase of bitrate. Compared with the previous algorithms, the max time saving can be up to 45% for some videos while the loss in bitrate is much less which is lower than 0.5% in average.

5. Conclusions

Complex inter prediction brings heavy computing cost and the huge increase of complexity leads to restriction on the promotion of HEVC, so that more and more scientific research institutes and enterprises begin to engage in studying the improvement of HEVC. To reduce the computational complexity, this paper proposed a fast inter prediction algorithm based on RD cost with using the temporal correlation to terminate the partition of LCU in early and using the local spatial information to terminate the partition of Non-LCU in early. With using the algorithm, the traversal of all CU partitions and PU modes for inter prediction can be avoided and the encoding speed can be improved.

The experiment results show that the method is computationally simple and feasible, and can achieve good performance on the whole. The loss in compression ratio and visual quality of reconstructed videos is negligible compared with HEVC test model. There is a significant reduction of computational complexity, especially for the videos with simple scenes or slowly changing images.

However, the performance for the videos with complex scenes and sharp changes should be enhanced. So the next goal of our research work is to further explore the correlation of CU sizes and video contents to improve the algorithm. Moreover, the proposed algorithm is implemented at CU level, which is independent with algorithms at PU level. Therefore, in the future, we will integrate the two kinds of approaches to HEVC to save more encoding time.

Acknowledgment

The authors would like to thank the students Heng Zhang and Lele Ren in vision computing and visualization laboratory of University of Science and Technology of China for providing assistance.

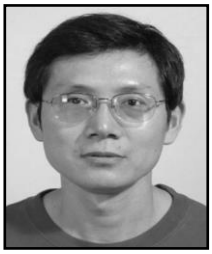
References

[1] J. R. Ohm and G. J. Sullivan, "High efficiency video coding: The next frontier in video compression [standards in a nutshell]", IEEE Signal Processing Magazine, vol. 30, no. 1, **(2013).**
[2] G. J. Sullivan, J. Ohm, W. J. Han and T. Wieg and, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE transactions on circuits and systems for video technology, vol. 22, no. 12, **(2012).**
[3] F. Sampaio, S. Bampi, M. Grellert, L. Agostini and J. Mattos, "Motion vectors merging: low complexity prediction unit decision heuristic for the inter-prediction of HEVC encoders", IEEE International Conference on Multimedia and Expo (ICME), Melbourne, Australia, **(2012)** July 09-13.
[4] L. Shen, Z. Liu, X. Zhang, W. Zhao and Z. Zhang, "An effective CU size decision method for HEVC encoders", IEEE Transactions on Multimedia, vol. 15, no. 2, **(2013)**.
[5] J. H. Lee, C. S. Park and B. G. Kim, "Fast coding algorithm based on adaptive coding depth range selection for HEVC", IEEE International Conference on Consumer Electronics-Berlin, Berlin, Germany, **(2012)** September 3-5.
[6] A. Glantz, M. Tok, A. Krutz and T. Sikora, "A block-adaptive skip mode for inter prediction based on parametric motion models", 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, **(2011)** September 11-14.
[7] J. L. Lin, Y. W. Chen, Y. P. Tsai, Y. W. Huang and S. Lei, "Motion vector coding techniques for HEVC", 13th International Workshop on Multimedia Signal Processing (MMSP), Hangzhou, China, **(2011)** October 17-19.
[8] M. Tok, A. Glantz, A. Krutz and T. Sikora, "Parametric motion vector prediction for hybrid video coding", 2012 IEEE Picture Coding Symposium (PCS), Krakow, Poland, **(2012)** May 7-9.
[9] S. Oudin, P. Helle, J. Stegemann, C. Bartnik, B. Bross, D. Marpe, H. Schwarz and T. Wiegand, "Block merging for quadtree-based video coding[C]", 2011 IEEE International Conference on Multimedia and Expo (ICME), Barcelona, Spain, **(2011)** July 11-15.
[10] P. Helle, S. Oudin, B. Bross, D. Marpe, M. O. Bici, K. Ugur, J. Jung, G. Clare and T. Wiegand, "Block merging for quadtree-based partitioning in HEVC", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, **(2012)**.
[11] P. Bordes, E. Francois and D. Thoreau, "Fast encoding algorithms for geometry-adaptive block partitioning", 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, **(2011)** September 11-14.
[12] X. Peng, J. Xu and F. Wu, "Exploiting inter-frame correlations in compound video coding", 2011 IEEE Visual Communications and Image Processing (VCIP), Tainan, Taiwan, **(2011)** November 6-9.
[13] M. Naccari, S. G. Blasi, M. Mrak and E. Izquierdo, "Improving inter prediction in HEVC with residual DPCM for lossless screen content coding", Picture Coding Symposium (PCS), San Josem, USA, **(2013)** December 8-11.
[14] J. Kim, S. Jeong, S. Cho and J. S. Choi, "An efficient bi-prediction algorithm for HEVC", 2012 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, **(2012)** January 13-16.
[15] S. Yang, H. J. Shim, K. Won, and B. Jeon, "Fast inter sub-partition prediction unit mode decision for HEVC", 2014 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, **(2014)** January 10-13.
[16] L. Shen, Z. Zhang and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatio-temporal correlations", IEEE Transactions on Circuits and Systems for Video Technology vol. 24, no. 10, **(2014)**.
[17] P. Nalluri, L. N. Alves and A. Navarro, "Fast motion estimation algorithm for HEVC", 2012 IEEE International Conference on Consumer Electronics, Las Vegas, USA, **(2012)** January 13-16.
[18] R. Garcia and H. Kalva, "HEVC inter-frame skip enhancement at low bit rate", 2014 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, **(2014)** January 10-13.
[19] J. Leng, L. Sun, T. Ikenaga and S. Sakaida, "Content based hierarchical fast coding unit decision algorithm for HEVC", 2011 International Conference on Multimedia and Signal Processing (CMSP), Guilin, China, **(2011)** May 14-15.
[20] J. Kim, S. Jeong, S. Cho and J. S. Choi, "Adaptive coding unit early termination algorithm for HEVC", 2012 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, **(2012)** January 13-16.
[21] X. Shen, L. Yu, and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule", 2012 IEEE Picture Coding Symposium (PCS), Krakow, Poland, **(2012)** May 7-9.
[22] J. Xiong, H. Li, Q. Wu and F. Meng, "A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence", IEEE transactions on multimedia, vol. 16, no. 2, **(2014).**

Authors

Jianfu Wang received the BE degree from the Department of computer science and technology, Ocean University of China, Qingdao, China, in 2010. He is currently working toward the Ph.D. degree at the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China. His research interests include image processing and video encoding and decoding techniques.

Lanfang Dong received her BE in Computer Science in 1991 from Lanzhou University in 1991, and obtained her ME in Computer Application in 1994 from USTC. Now she is an associate professor in the School of Computer Science and Technology at USTC. She was also a software engineer at Sun-USTC from 1994 to 2000.Her research interests include Computer Animation, Intelligent Image Analysis and Visualization in Scientific Computing.



Yinlong Xu received his B.S. in Mathematics from Peking University in 1983, and MS and Ph.D. in Computer Science from USTC of China in 1989 and 2004 respectively. He is currently a professor with the School of Computer Science and Technology at USTC. He is leading a research group there in networking and high performance computing. His research interests include network coding, wireless network, combinatorial optimization, design and analysis of parallel algorithm, parallel programming tools, etc. He received the Excellent Ph.D. Advisor Award of Chinese Academy of Sciences in 2006.