Edge Direction-Based Fast Coding Unit Partition for HEVC Screen Content Coding

Mengmeng Zhang[1] and Yangxiao Ou[2]

[1] North China University of Technology, Beijing, P.R. China
zmm@ncut.edu.cn
[2] North China University of Technology, Beijing, P.R. China
nvniaobi@sina.com

Abstract. High efficiency Video Coding is a new video coding standard that presents numerous advantages over previous video coding standards. However, Rate distortion optimization (RDO) complexity is extremely high for screen content coding, which cannot adjust to the real-time performance. This paper proposed a fast and efficient algorithm based on edge direction to partition coding units (CUs) based on their relationship with edge direction. Sobel operator is used to determine the edge direction from the total image before intra prediction. The key point of this algorithm is to determine the relationship between the edge direction and CUs. Experimental results show that the proposed edge direction-based CU partition algorithm provides a decrease in the screen content coding processing time up to 39%, with a little increase in bit-rate(0.7% on average) and a negligible reduction in the PSNR value.

Keywords: HEVC, edge direction, Sobel, coding unit, screen content coding.

1 Introduction

High-efficiency Video Coding (HEVC) is a novel international video coding standard developed by the Joint Video Team from ISO/IEC MPEG and ITU-T VCEG. HEVC adopts numerous new tools, including a highly flexible data structure representation, which includes the coding unit (CU), prediction unit (PU), and transform unit (TU). RDO is applied, during the partitioning of largest CUs. HEVC supports four intra modes in intra coding, including intra 64x64, intra 32×32, intra 16×16, intra 8×8. Some new tools are also adopted in screen content coding, Intra Block Copy (IntraBC) reuses the notion of Sliding Window Lempel-Ziv coding, which represents a block of pixels by its length and a displacement pointing to a previous occurrence in spatially reconstructed region, to search frequently occurred patterns in the current frame. The complexity of mode decision and the intra predictor generation comprise 57% of intra coding, which serves as a processing bottleneck. In the Serial Real-time Communications System, for example the video conference, the high complexity of RDO is the biggest barrier.

Fast decision algorithms have been recently proposed to reduce computation complexity. A smooth CU usually selects a lager intra coding block, whereas a complex

X. He et al. (Eds.): MMM 2015, Part I, LNCS 8935, pp. 477–486, 2015.

CU usually chooses a smaller coding block because the block size of intra coding is dependent on CU smoothness. Designs [1] facilitated an improvement in the intra block size decision by using the number of edge points. This algorithm reduced computation complexity with a slight Peak Signal to Noise Ratio (PSNR) decrease in H.264 and a significant increase in the coding stream when applied to the screen content coding in HEVC. Designs [2] proposed a fast intra prediction mode based on edge direction, but efficiency is insufficient in terms of the mode decision. Design [3] proposed an edge mode algorithm in intra prediction and achieves sufficient efficiency, but this algorithm is done in HM7.0 which is not the best reference software for screen content coding. Design [4] proposed a content-based method with efficient inter prediction mode decision. In design [5], the author proposed an algorithm which can decide the prediction mode rapidly and the algorithm have a decrease of 50% in complexity of HEVC, but it has an big increase in bit-rate(about 2.5%). Paper [7] presented a design for a self-adaptive algorithm based on edge information and MAD which achieves sufficient efficiency and reduces intra prediction complexity. Design [8] provides a fast encoding techniques for skipping of IntraBC search which result in about 21%-24% encoding time reduction for intra coding.

This paper proposes an CU partition algorithm which is based on edge direction. This partition mode can provide structural support for some transform, for example, contourlet transform, bandelet transform and so on. Edge direction of the current CU is extracted by using the Sobel mask which has low complexity. In screen content coding, CUs usually have more complex but more regular edges than natural sequence and this is why edge information is more effective in intra prediction. Thus, the edge direction aids in selecting which CU requires partitioning. We introduce the concept of feature value to intuitively illustrate the relationship between edge direction and the partition mode. We achieve a range of 31% to 64% reduction in complexity when using the proposed algorithm compared with the use of HM12.1+RExt5.1 with a PSNR drops of only 0.03dB on average. Compared to the result in paper [9] which results in about 27.6% encoding time reduction and 10.9% BD-rate increasing in screen content coding, the proposed algorithm is more suitable for screen content coding.

The remainder of this paper is organized as follows. Section II presents the proposed algorithm. Section III discussed the experimental results and a simple conclusion is provided in Section IV.

2 Proposed Algorithm

2.1 Overview

Figure 1 illustrates that CU partition size selection problem can be reduced to a set of the most probable partitions, relative to the texture of the video frames. A large CU with high edge complexity must be subdivided to decrease individual complexity, whereas small partitions with low complexity can be combined to form large partitions. We note that highly detailed regions (blocks B and C having high edge complexity) are encoded using small CU sizes, whereas less detailed regions are encoded using large CU sizes (block A).

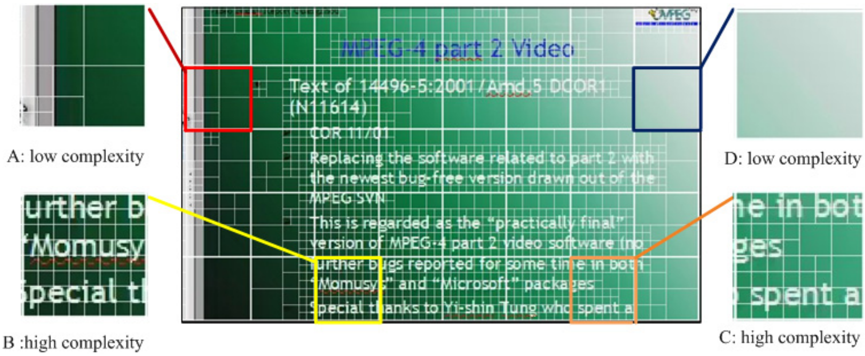

Fig. 1. CU borders in the first frame of sc_WordEditing

The flowchart of our proposed scheme is shown in Figure 2. As previously mentioned, the key point of the proposed algorithm is to determine the relationship between the selected CUs and their edge direction. Thus, a mathematical model must be built between the edge direction and the partition mode of CUs. In next section, we introduce a feature value to present the edge complexity of CUs.

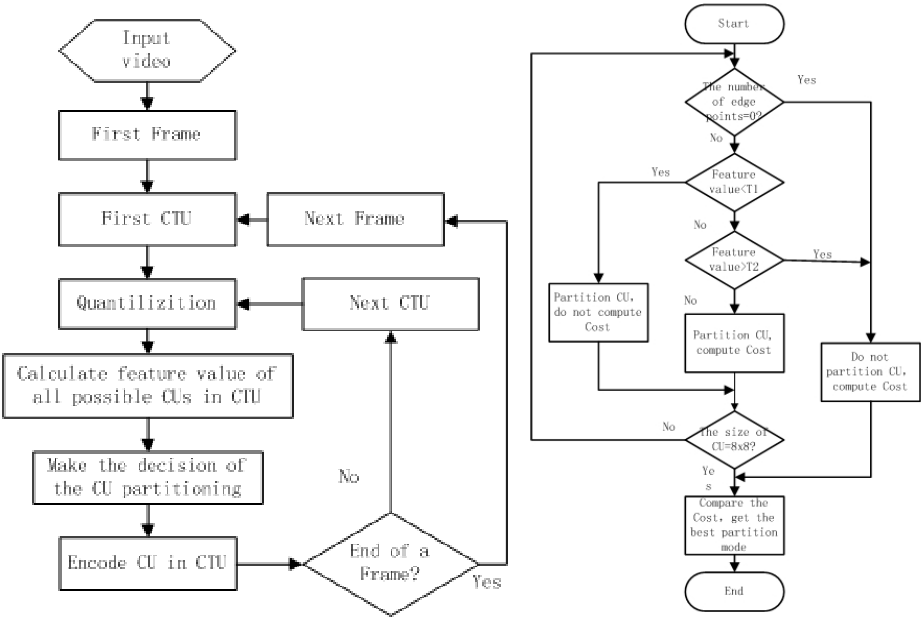

Fig. 2.(a) Flowchart of proposed algorithm **Fig. 2.**(b) Flowchart of mode decision

Fig. 2. Flowchart of the proposed algorithm

2.2 Feature Value Calculation

The key point of proposed algorithm is to determine out the relationship between the edge direction and the partition modes of CUs. We need to obtain a feature value that can directly show this correlation because we cannot establish a mathematical model between edge direction and the partition mode of CUs. The feature value presents the edge complexity of CU. The method to calculate the feature value is given follows.

Before calculating the edge information, we split the range $(0°, 180°)$into 12 intervals:. $(0°, 15°)$ $(15°, 30°)$ $(30°, 45°)$ $(45°, 60°)$ $(60°, 75°)$ $(75°, 90°)$ $(90°, 105°)$ $(105°, 120°)$ $(120°, 135°)$ $(0°, 150°)$ $(150°, 165°)$ $(165°, 180°)$.

To reduce computation complexity, we adopted 6 as the stepper to quantize the 256 pixel values, which will not result in too much details losing. We use Sobel operator to calculate the edge information of all the pixels for every CU.

−1	0	1
−2	0	2
−1	0	1

(a)Horizontal Mask

−1	−2	−1
0	0	0
1	2	1

(b)Vertical Mask

Fig. 3. Sobel operator masks

First, we calculate the gradient Rectangular Coordinate System using the classical Sobel operator. Two convolution masks of 3×3 pixels are presented in Figure 2. The gradient vector $D_{i,j}$ of pixel $p_{i,j}$ can be calculated as;

$$D_{i,j} = \{D_{x_{i,j}}, D_{y_{i,j}}\} \tag{1}$$

And;

$$D_{x_{i,j}} = p_{i+1,j-1} + 2 \times p_{i+1,j} + p_{i+1,j+1} - p_{i-1,j-1} - 2 \times p_{i-1,j} - p_{i-1,j+1} \tag{2}$$

$$D_{y_{i,j}} = p_{i-1,j-1} + 2 \times p_{i,j-1} + p_{i+1,j-1} - p_{i-1,j+1} - 2 \times p_{i,j+1} - p_{i+1,j+1} \tag{3}$$

Where $D_{x_{i,j}}$ and $D_{y_{i,j}}$ represent the differences in the vertical and horizontal directions respectively. Therefore, the amplitude of the gradient vector can be roughly estimated by $G_{i,j}$, which is defined as

$$G_{i,j} = \sqrt{D_{x_{i,j}}^2 + D_{y_{i,j}}^2} \tag{4}$$

We then choose an appropriate threshold T. If $G_{i,j} > T$, then pixel $p_{i,j}$ is considered as an edge point. Using the information shown above, we can get the gradient direction θ by using Formula (5).

$$\theta = arccos(\frac{D_{x_{i,j}}}{G_{i,j}}) \tag{5}$$

When we get the gradient direction θ of all the pixels in a CU, we do statistics on the number of edge points whose gradient direction fall within the 12 intervals shown above separately. The number of edge points in the 12 intervals are recorded as I_1 $I_2 I_3 \ldots . I_{12}$. Then we choose the largest from $I_i(i = 1,2,3 \ldots \ldots 12)$ and call it I. We then take ΔI as the feature value which represent the edge complexity of this CU. We calculate ΔI by using Formula (6).

$$\Delta I = I / \sum_{i=1}^{12} I_i \tag{6}$$

The sum of I_i (i =1,2,3…….12) is not equal to zero; otherwise, we consider that this CU has no edge points. At this point, we can obtain the feature value that presents the edge complexity of CUs. We then have to decide on which CU requires partitioning and which does not.

2.3 Threshold Decision and Judgment of Proposed Algorithm

We collected sufficient CUs which are partitioned by HEVC from different screen sequences under different Quantization Step (QP 22, 27, 32, 37), different frame rate (20, 30 ,60). These CUs are partitioned by RDO to obtain the threshold of the proposed algorithm. The CUs include 8×8, 16×16, 32×32 and 64×64. The sequences include six different screen content sequences which represent most of the sequences of screen content coding. With the work in the previous section, we can easily build a gradient line chart based on the correlation between the feature value and CUs shown in Figure 4.

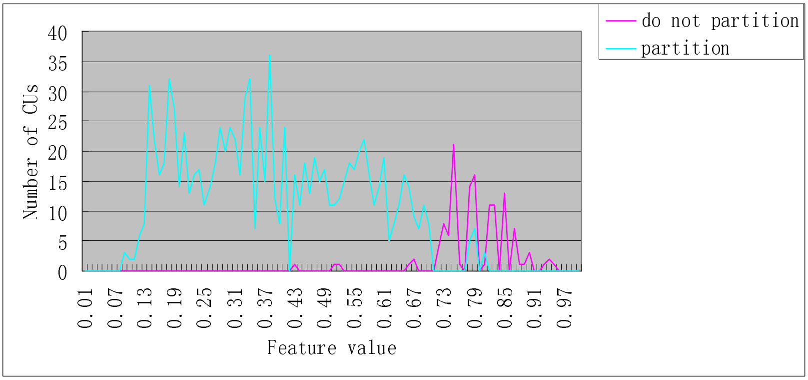

Fig. 4. (a) CUs of 64×64 size

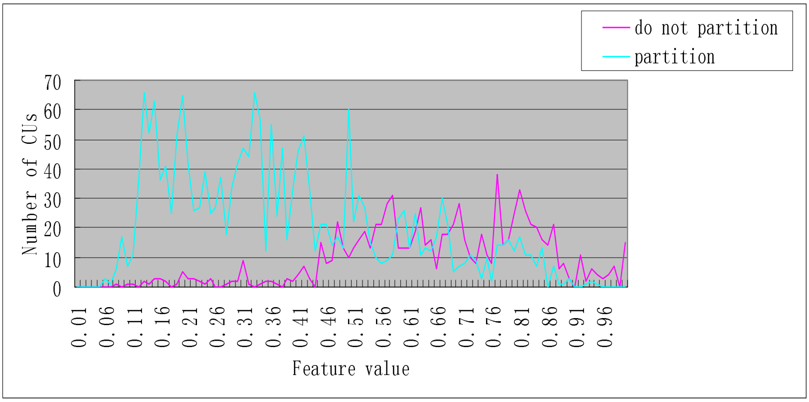

Fig. 4. (b) CUs of 32×32 size

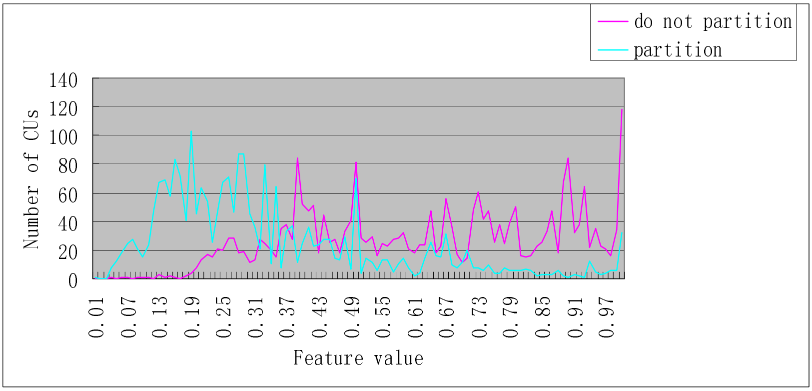

Fig. 4. (c) CUs of 16×16 size

Fig. 4. Relationship between CUs and feature value

Figure 4 (a), (b), (c) show a clear correlation between the feature value and CUs which are partitioned by HEVC. When the feature value is sufficiently small, the CU will be partitioned. Conversely, when feature value is large, the CU will not be partitioned. In Figure 4, we can see when the feature value is not large enough or sufficient small, we cannot decide this CU will be partitioned or not. In this case, we should adopt RDO as the principle to decide whether this CU will be partitioned or not. Based on this relationship, we can obtain the threshold of the CUs of 16×16, 32×32 and 64×64 separately, through which we can determine whether CUs require partitioning. We compute the integral for Figure 4 (a), (b) and (c) to obtain $F(\Delta I)$ and $D(\Delta I)$ which present the ratio of partitioned CUs and the CUs will not be partitioned.

$$F(\Delta I) = \sum_{i=0.00}^{\Delta I} q_i / (\sum_{i=0.00}^{\Delta I} n_i + \sum_{i=0.00}^{\Delta I} q_i) \tag{7}$$

$$D(\Delta I) = \sum_{i=\Delta I}^{1} n_i / (\sum_{i=\Delta I}^{1} n_i + \sum_{i=\Delta I}^{1} q_i) \tag{8}$$

As introduced in the previous section, ΔI is the feature value. n_i presents the number of CUs with the same feature value that will not be partitioned. q_i presents the number of CUs with the same feature value that will be partitioned (i=0, 0.01, 0.02…….1). Through our research, we derived these principles for selecting the threshold:

- When $F(\Delta I)$ is close to 1, then the threshold T_1 is equal to ΔI.
- When $D(\Delta I)$ is close to 1, then the threshold T_2 is equal to ΔI.

Based on these rules shown above, we can easily obtain thresholds T_1 and T_2 based on CU sizes. The best values of T_1 and T_2 are as follows.

- When the size of CU is 64×64, T_1 is equal to 0.7, and T_2 is equal to 0.97.
- When the size of CU is 32×32, T_1 is equal to 0.4, and T_2 is equal to 0.94.
- When the size of CU is 16×16, T_1 is equal to 0.17, and T_2 is equal to 1.

By now we get the feature value and threshold, so we can describe this algorithm in a CTU which is the basic partition unit in video frames. The proposed algorithm is as follow.

1. If no edge points are found in this CU, we will compute for the cost and decide that this CU will not be partitioned, then jump to step 6. Otherwise jump to step 2.
2. If the feature value (shown in section 2.2) of this CU is smaller than T_1 which is modified with the size of CU changing, the CU is partitioned without cost calculation, then jump to step 5. If not, jump to step 3.
3. If the feature value of this CU is greater than T_2 which is modified with the size of CU changing, we compute for the cost and do not partition the CU, then jump to step 6. If not, jump to step 4.
4. We partition this CU and compute for the cost and jump to step 5.
5. When this decision is completed and the decision is to partition this CU, we determine whether the size of this CU is 8×8. If not, we will skip to step 1. Otherwise we jump to step 6.
6. If the size of current CU is 8×8 or the decision is not to partition this CU, we choose the best partition mode based on the cost of the different partitioning structures.

3 Experimental Results

Up to all the frames of each screen content sequence are coded to test the performance of the proposed algorithm, and the test condition is "encoder_intra_main_rext". QP values are set to 22, 27, 32, and 37. A computer with a 2.8 GHz core was used in this experiment. We used HM12.1+RExt5.1 for the comparison to determine the performance of the proposed algorithm.

$$T = \frac{T_{proposed} - T_{HM12.1+RExt5.1}}{T_{HM12.1+RExt5.1}} \times 100\% \quad (9)$$

$$\Delta Y_{PSNR} = Y_{PSNR_{proposed}} - Y_{PSNR_{HM12.1+RExt5.1}} \quad (10)$$

$$\Delta Bitrate = \frac{Bitrate_{proposed} - Bitrate_{HM12.1+RExt5.1}}{Bitrate_{HM12.1+RExt5.1}} \quad (11)$$

In the equation (9), (10) and (11), $T_{proposed}$, $Y_{PSNR_{proposed}}$ and $Bitrate_{proposed}$ represent the encoding time, PSNR and the bitrate of the proposed algorithm. $T_{HM12.1+RExt5.1}$, $Y_{PSNR_{HM12.1+RExt5.1}}$ and $Bitrate_{HM12.1+RExt5.1}$ represent the encoding time, PSNR and the bitrate of HM12.1+RExt5.1. BD-PSNR/Rate measures are used to evaluate the changes in coding efficiency. The percentage difference in bit-rate ($\Delta Bitrate$) and the luminance PSNR difference (ΔPSNR) are also used to compare the proposed algorithm with HM 12.1+RExt5.1. Table 1 and Figure 5 shows the results comparison of the proposed algorithm with the design [1] which is optimized by us to adapt to the screen content coding, the proposed algorithm achieves an encoding time reduction of 39% compared with the anchor(the default algorithm of HM12.1+RExt5.1) with only a 1.1% loss of BD-rate on average. In Table 1, we can see the reduction time T

Table 1. Results of the proposed algorithm

Sequence	Proposed algorithm					Algorithm in paper [1]				
	QP	*ΔBitrate* (%)	T (%)	ΔY-PSNR (dB)	Y-BD-rate (%)	QP	*ΔBitrate* (%)	T (%)	ΔY-PSNR (dB)	Y-BD-rate (%)
sc_map_1280x720_60_8bit_444	22	0.6	-31	-0.03	0.6	22	0.9	22.1	-0.05	1.0
	27	0.5	-30.4	-0.01		27	0.7	22.0	-0.04	
	32	0.5	-30.1	0		32	0.6	23.7	-0.01	
	37	1.1	-28.7	0.02		37	1.0	23.9	0	
sc_web_browsing_1280x720_30_8bit_444	22	1.1	-37.1	0.02	1.2	22	4.7	29.1	-0.16	5.1
	27	1.2	-41.3	0.07		27	3.9	29.4	-0.14	
	32	1.1	-38.3	-0.01		32	2.8	29.5	-0.18	
	37	1.3	-37.4	-0.06		37	2.5	30.6	-0.25	
sc_wordEditing_1280x720_60_8bit_444	22	0.6	-32.2	-0.07	1.3	22	3.0	26.5	-0.26	4.9
	27	0.5	-28.5	-0.07		27	2.9	27.3	-0.22	
	32	0.8	-32	-0.08		32	3.0	27.6	-0.19	
	37	0.8	-31.5	-0.1		37	2.7	27.7	-0.29	
sc_SlideShow_1280x720_20_8bit_444	22	1.2	-60.6	-0.08	2	22	10.2	57.5	-0.08	11.1
	27	1	-62.8	-0.07		27	9.9	58.3	-0.19	
	32	1.5	-64.7	-0.02		32	8.4	59.1	-0.12	
	37	2.7	-63.1	0		37	7.8	61.0	-0.15	
sc_programming_1280x720_60_8bit_444	22	0.9	-34.3	-0.09	1.3	22	2.6	29.3	-0.18	3.2
	27	0.9	-31.9	0.01		27	2.3	28.5	-0.07	
	32	1	-33.7	-0.04		32	2.0	27.3	-0.13	
	37	1.1	-34	-0.07		37	1.7	27.5	-0.14	
sc_ppt_doc_xls_1920x1080_20_8bit_444	22	0.4	-31.6	-0.04	0.5	22	2.2	19.6	-0.20	2.7
	27	0.3	-27.3	0.02		27	1.9	18.7	-0.25	
	32	0.5	-28.1	-0.01		32	1.7	19.4	-0.10	
	37	0.5	-24.5	-0.08		37	1.4	22.2	-0.18	
Average		0.7	-39	-0.03	1.1		2.2	32	-0.15	4.7

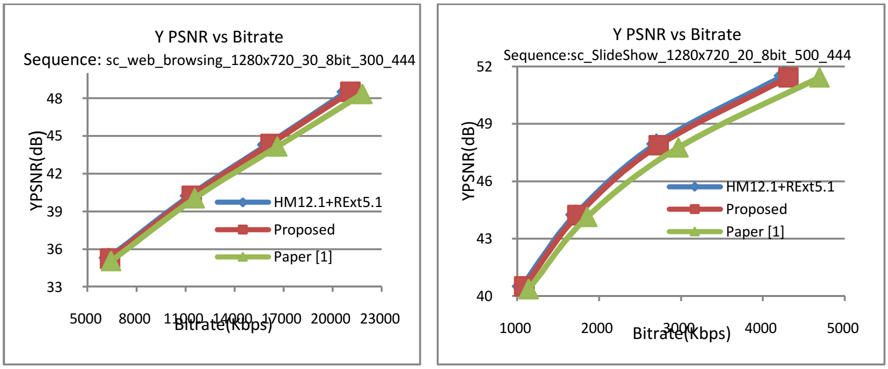

Fig. 5. Y PSNR vs Bitrate

of sequence "sc_ppt_doc_xls_1920x1080_20_8bit_444" is less than the others, because the edge direction of this sequence is more complex and it has more complex edge direction than the other sequences.

4 Conclusions

In this paper, we presented a new algorithm for partitioning CTU in the screen sequence by using the correlation between the edge direction and CUs. A strong correlation between edge direction and the partition mode of CUs was determined through analysis. Such analysis results in a simple decision that can be used for CU partitioning. The proposed algorithm aims to reduce computation complexity significantly with an acceptable loss of BD rate. Results show that the proposed algorithm significantly reduced coding time with an acceptable decrease in quality, which indicates that RDO computation complexity was significantly reduced. Thus, the proposed algorithm meets the requirements of screen content encoding. Next we will focus on transform and quantization, which can provides a significant bit-rate saving.

Acknowledgements. This work is supported by the Natural National Science Foundation of China (No.61370111, No. 61103113) and Beijing Nova Programme (Z141110001800000).

References

1. Chang, C.-Y., Chien, C.-A., Chang, H.-C., Chen, J.-W., Guo, J.-I.: A two level mode decision algorithm for H.264 high profile intra encoding. In: 2012 IEEE International Sysmposium on Circuits and Systems (ISCAS) (May 2012)
2. Ye, T., Zhang, D., Dai, F., Zhang, Y.: Fast mode decision algorithm for intra prediction in HEVC. ACM (August 2013)
3. Hu, S., Cohen, R.A., Vetro, A., Kuo, C.-C.J.: Screen Content Coding For HEVC Using Edge Modes. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (May 2013)
4. Lin, Y.K., Chang, T.S.: Fast block type decision algorithm for intra prediction in H.264 FRext. In: Proc. ICIP, vol. 1, pp. 585–588 (2005)
5. Na, S., Lee, W., Yoo, K.: Edge-Based Fast Mode Decision Algorithm for Intra Prediction in HEVC. In: 2014 International Conference on Cpnsumer Electronics (ICCE) (January 2014)
6. Yuan, Y., Sun, X.: Edge Information Based Effective Intra Mode Decision Algorithm. In: 2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC) (2012)
7. Bai, C., Yuan, C.: Fast coding tree unit decision for HEVC intra coding. In: 2013 IEEE ICCE-China Workshop, ICCE-China (April 2013)
8. Kwon, D.-K., Budagavi, M.: Fast intra block copy (IntraBC) search for HEVC screen content coding. In: 2014 IEEE International Symposium on Circuits and Systems (ISCAS (June 2014)
9. Jiangyong, S., Xiling, L., Jun, Z.: An Edge-based Approach for Video Text Extraction. In: International Conference on Computer Technology and Development, ICCTD 2009 (2009)

10. Wang, X., Zhai, C.: An Improved Fast Intra-prediction Algorithm in H.264/AVC Based on Edge Feature. In: 2008 International Conference on Computer Science and Software Engineering (2008)
11. Yan, S., Hong, L., He, W., Wang, Q.: Group-Based Fast Mode Decision Algorithm for Intra Prediction in HEVC. In: 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS) (2012)
12. Dung, T., Sole, J., Yin, P., Gomila, G., Nguyen, T.Q.: Selective Data Pruning-Based Compression Using High-Order Edge-Directed Interpolation, doi: 10.1109/Tip.2009.2035845 (2010)
13. Leng, J., Sun, L.: Content Based Hierarchical Fast Coding Unit Decision Algorithm For HEVC. IEEE Computer Society (May 2011)
14. Bai, H., Zhu, C., Zhao, Y.: Optimized Multiple Description Lattice Vector Quantization for Wavelet Image Coding. IEEE Transactions on Curcyuts and Systems for Video Technology (2007)