Sensors & Transducers, Vol. 183, Issue 12, December 2014, pp. 245-252

www.sensorsportal.com

Sensors & Transducers

http://www.sensorsportal.com

Fast Adaptive Coding Unit Depth Range Selection Algorithm for High Efficiency Video Coding

Fang Shuqing, Yu Mei, Chen Fen, Xu Shengyang, Jiang Gangyi
Faculty of Information Science and Engineering, Ningbo University, Ningbo, 315211, China
Tel.: 86-574-87600411
E-mail: jianggangyi@126.com

Received: 23 September 2014 /Accepted: 28 November 2014 /Published: 31 December 2014

Abstract: The emerging high-efficiency video coding standard employs a new coding structure characterized by coding unit, prediction unit and transform unit. It improves the coding efficiency significantly, but also introduces great computational complexity on the decision of optimal coding unit, prediction unit and transforms unit sizes. To reduce the encoding complexity, a fast adaptive coding unit depth range selection algorithm is proposed. In the proposed scheme, first of all, the average depth error between adjacent and their co-located largest coding unit are utilized to determine depth range of current largest coding unit. And then, depth scaling factor in the previous and back frame are obtained to shrink the depth range. Furthermore, we also propose a depth range correction algorithm for reducing misjudgment of changes in the larger sequences. Experimental results show that the former algorithm can save encoding time of about 10% more than Shen's algorithm with a BD-bitrates loss of 0.81 % and a BD-PSNR loss of 0.026 dB. Correction algorithm can save same encoding time of Shen's algorithm with a BD-bitrates lowering 0.76 % and a BD-PSNR improvement of 0.028 dB.

Keywords: HEVC, Depth range, Average depth error, Depth scaling factor, Temporal-spatial similarity.

1. Introduction

Currently, the content growth of high definition (HD) and the HD broadcasting service are offering uses to enjoy high quality and high resolution. In the future, the content growth of ultra high definition (UHD) and the UHD broadcasting service [1-4] also will be offered to users following the demands to needs for higher quality and higher resolution. However, MPEG-2, MPEG-4 and H.264 have a difficulty to meet above requirements. Therefore, in January, 2010, Video Coding Experts Group (VCEG) and ISO/IEC (MPEG) founded a Joint Collaborative Team on video coding to develop the next generation video coding standard, namely High Efficiency Video Coding (HEVC) [5] which aims to further reduce bit rate in half with the same reconstructed video quality compared with H.264 [6]. While HEVC took recursive quad-tree structured [7] coding unit (CU) which makes HEVC coding more efficient, but it also makes the HEVC have several times higher complexity.

There have been extensive researches on reducing HEVC computational complexity. Li [8] et al. employed spatial similarity to predict depth range of current largest coding unit (LCU), but it had a limitation on time saving because at least three depths were needed to traverse. Shen [9] et al. utilized temporal-spatial similarity to predict depth range of current LCU through assigning weights for adjacent LCU and co-located LCU in the previous frame. Although the method could reduce candidates of CU depth, it needed some improvement because of regardless of difference of video sequence. Fixed weights were not suitable for all sequences. In addition, Kim [10] et al. employed a threshold which was obtained by Bayesian theory training to terminate CU splitting process and Yu [11] et al. also terminated CU splitting process through mean squared error.

This paper proposes a fast adaptive CU depth range selection algorithm which improves Shen's method. We assign weights adaptively according to average depth error between adjacent and their co-located LCU for predicting depth range of current LCU accurately. Then, depth scaling factor in the previous and back frames are utilized to shrink depth range (DR) thus reducing computational complexity continue. Finally, we put forward a correction algorithm (CA) based on linear weighted for depth scaling factor. The CA can achieve higher performance than Shen's method.

The paper is organized as follows. In the next section, the complexity problem in CU splitting process is analyzed. In Section 3, the proposed method is described in detail. Section 4 shows the corresponding experimental results. Finally, we conclude this paper in Section 5.

2. Computational Complexity Problem

In HEVC, every CU will be split into four equal sub-CUs through recursive method. The size of sub-CU is 32×32, 16×16 or 8×8. The final segmentation result is determined by rate-distortion function [12]. Fig. 1 shows the segmentation process. The rate-distortion function is defined [13] as follows:

$$J = SSE_{luma} + w_{chroma} \times SSE_{chroma} + \lambda \times B, \tag{1}$$

where B represents necessary bits after predicting for current CU, SSE_{luma} and SSE_{chroma} is the sum of square error between original block and reconstructed block about luminance and chroma respectively, w_{chroma} is the weight of chroma and λ is the Lagrange multiplier.

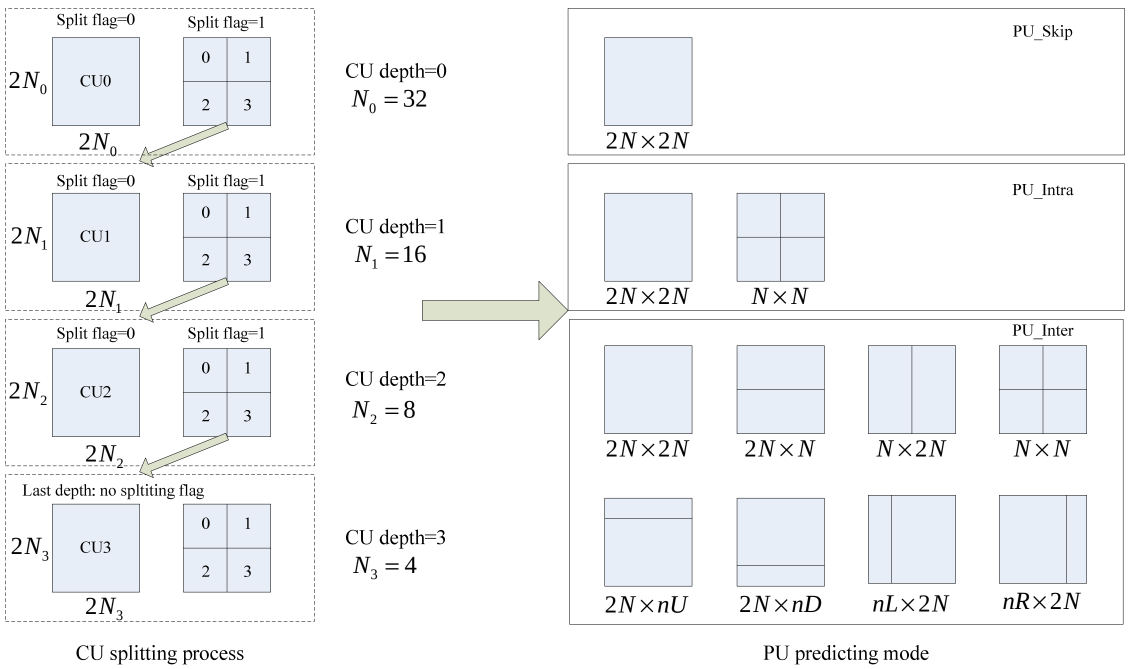


Fig. 1. CU splitting process in HM.

In HM, the CU is of four different possible sizes: 64×64, 32×32, 16×16 and 8×8. Each LCU can be split into four CUs recursively up to the maximum allowable hierarchical depth. CU splitting process in HM about a LCU is shown in Fig 1. Firstly, HM codes current 64×64 LCU and calculates rate-distortion cost. Then, it splits LCU into four sub-CU which depth is one and calculates rate-distortion cost of the first sub-CU. When depth is two or three, HM do the same process until the depth reaches three. Finally, HM starts recursion from depth 3 to depth 0. For instance, if four 8×8 sub-CU block's rate-distortion cost is less than rate-distortion of 16×16 CU, HM chooses 8×8 CU, otherwise choosing 16×16 CU. At the same time, the leaf node CUs can be further split into predicting unit (PU)s. PU is the basic unit for prediction and it allows multiple different shapes to encode irregular image patterns as shown in Fig 1. The transform unit (TU) is defined for residual transform and quantization. From the leaf node of CU, the TU can be split into four sub-TUs recursively until the minimal allowable TU depth, and signaled by a flag.

As analyzed above, HEVC performs full search on all possible CU size, mode and TU size by evaluating the rate distortion (RD) cost. Therefore, it results in a substantial computational complexity of HEVC about inter frame.

3. Proposed Method

In this section, a fast CU size decision algorithm based on temporal-spatial similarity and quad-tree coding structure is described, including adaptive CU depth range selection algorithm and correction algorithm. We start with statistical analysis for temporal-spatial similarity, which provide useful guidelines for assigning weights. We utilize average depth error (ADE) which is defined by previous frame and current frame to assign weights for co-located LCU and adjacent LCU in order to get predicting depth range. Then, we can continue shrink depth range according to depth scaling factor in the previous and back frames. Finally, for reducing misjudgment of changes in the larger sequences, a correction algorithm is proposed.

3.1. Statistical Analysis for Temporal-Spatial Similarity

HEVC utilizes quad-tree structure and CU depth to decide segmentation after traversing every depth in the [0,3]. However, we can reduce computational complexity through predicting DR in advance thus skipping unnecessary depth. This paper utilizes temporal-spatial similarity to predict DR type of current LCU. We define *ADE* according to Fig. 2 for utilizing temporal-spatial similarity accurately. The *ADE* is defined as follows

$$ADE = (|LADE| + |TADE|)/2\,, \qquad (2)$$

where *LADE* is average depth error between Left LCU and CLeft LCU, *TADE* is average depth error between Top LCU and CTop LCU.

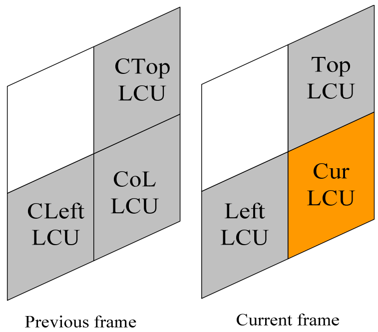


Fig. 2. Temporal-spatial similarity of LCU.

As we know, DR type of current LCU is similar to adjacent LCU and co-located LCU in the reference. We should define temporal similarity factor (TSF) and spatial similarity factor (SSF) in order to obtain relationship between ADE and temporal-spatial similarity. The *TSF* and *SSF* are defined as follows:

$$TSF_i = num_i / N_i\,, \qquad (3)$$

$$SSF_i = n_i / N_i\,, \qquad (4)$$

where *i* represents interval of *ADE*, with 0 standing for [0, 1], 1 for [1, 2] and 2 for [2, 3]. num_i is the 4×4 number of equal depth in the interval, n_i is the number of equal average depth in the interval, N_i is the total 4×4 number of interval. The temporal-spatial similarity results are demonstrated in Table 1.

Table 1. The temporal-spatial similarity results of LCU.

	CoL-LCU			Left-LCU			Top-LCU		
Sequence	**[0,1]**	**[1,2]**	**[2,3]**	**[0,1]**	**[1,2]**	**[2,3]**	**[0,1]**	**[1,2]**	**[2,3]**
BasketballDrill	0.576	0.307	0.039	0.334	0.345	0.791	0.376	0.384	0.801
BasketballDrillText	0.571	0.303	0.038	0.338	0.340	0.779	0.376	0.385	0.759
BasketballPass	0.775	0.453	0.344	0.201	0.325	0.333	0.598	0.599	0.667
BlowingBubbles	0.491	0.269	0.065	0.181	0.284	0.609	0.251	0.323	0.435
PartyScene	0.517	0.225	0.016	0.247	0.315	0.782	0.310	0.371	0.800
PeopleOnStreet	0.510	0.320	0.101	0.187	0.332	0.667	0.171	0.286	0.660
RaceHorsesC	0.413	0.279	0.158	0.102	0.217	0.250	0.132	0.250	0.278
Average	0.550	0.308	0.109	0.227	0.308	0.602	0.316	0.371	0.629

From Table 1, with increment of *ADE*, temporal similarity gradually reduces and spatial similarity improves. Accordingly, we can predict DR type of current LCU through obtaining adaptive weights according to *ADE*.

3.2. Adaptive Depth Range Selection Scheme

Video sequence has a strong temporal-spatial similarity especially in flat area. The DR type of current LCU is similar to adjacent LCU. At the same time, co-located LCU in the reference can be referential basis of current LCU because of high correlation for video sequence. So we can utilize adjacent LCU and co-located LCU to predict DR type of current LCU thus skipping unnecessary CU depth and PU predicting process. We can use average depth of co-located LCU, left LCU and top LCU to define $Depth_{pred}$ in order to determine DR type of current LCU. $Depth_{pred}$ is defined as follows:

$$Depth_{pred} = \sum_{i=0}^{N} w_i \times avedepth_i \ , \qquad (5)$$

where N=2, i is the index of reference LCU, with 0 standing for co-located LCU, 1 for left LCU and 2 for top LCU. $avedepth_i$ is average depth of corresponding LCU. w_i is weights of corresponding LCU and the sum is equal to 1. The weights of co-located LCU, left LCU and top LCU is $-0.1\times ADE+0.5$, $0.05\times ADE+0.25$ and $0.05\times ADE+0.25$ respectively according to analysis results as shown in Table 1.

We can predict DR type of current LCU according to $Depth_{pred}$ as mentioned above. The relationship between $Depth_{pred}$ and DR type is demonstrated in Table 2.

Table 2. Relationship between predicting depth and DR type.

$Depth_{pred}$	**Candidates Depth**	**DR**
0	0	[0,0]
(0, 0.5]	0,1	[0,1]
(0.5, 1.5]	0,1,2	[0,2]
(1.5, 2.5]	1,2,3	[1,3]
(2.5, 3]	2,3	[2,3]

From Table 2, we can skip unnecessary CU depth according to DR type which is obtained through $Depth_{pred}$ thus reducing computational complexity.

In order to prove accuracy of the DR type, we make a statistics about the accuracy of DR type including depth after HM algorithm. The accuracy is demonstrated in Table 3.

Table 3. The accuracy of DR type.

Sequence	**[0,0]**	**[0,1]**	**[0,2]**	**[1,3]**	**[2,3]**
BasketballDrill	94.32 %	96.41 %	96.34 %	91.95 %	95.83 %
Traffic	98.79 %	98.54 %	98.18 %	86.60 %	93.57 %
BasketballDrillText	94.80 %	97.11 %	96.10 %	92.54 %	89.58 %
BlowingBubbles	100 %	100 %	97.51 %	92.17 %	97.92 %
Cactus	98.55 %	98.63 %	98.44 %	92.04 %	94.57 %
PartyScene	98.10 %	97.59 %	96.80 %	92.21 %	93.64 %
PeopleOnStreet	97.02 %	98.05 %	96.49 %	96.58 %	92.13 %
ChinaSpeed	93.76 %	99.12 %	97.08 %	94.18 %	88.75 %
Average	96.92 %	98.18 %	97.12 %	92.28 %	93.25 %

From Table 3, misjudgment rate of five DR type are less than 8 %, especially in first three DR type only 1.82 %-3.08 %. So, proposed ACUDR algorithm can achieve high reliability.

3.3. Depth Scaling Factor Shrink DR

We can't reduce much computational complexity in interval [0, 2] and [1, 3] because we still have three depth need to traverse. As we know, neighbor frames have a similar CU depth because of high correlation for video sequence. Therefore, we define depth scaling factor (DSF) according to previous and back frames in order to reduce DR thus reducing computational complexity continue. The DSF is defined as follows

$$DSF_{ij} = count_{ij} / N \ , \qquad (6)$$

where i=0, 1 represents co-located LCU in the previous and back frame respectively, j represents CU depth. $count_{ij}$ represents CU depth 4×4 number in the co-located LCU, N=256.

For DR= [0, 2], if DSF_{00} and DSF_{10} are equal to 0, depth 0 is eliminated from the candidate depth; if DSF_{02} and DSF_{12} are less than 0.125, we traverse only interval [0, 1].

For DR= [1, 3], if DSF_{01} and DSF_{11} are equal to 0, depth 1 is eliminated from the candidate depth; if DSF_{03} and DSF_{13} are less than 0.125, we traverse only interval [1, 2].

For DR= [2, 3], if DSF_{02} and DSF_{12} are less than 0.125, depth 2 is eliminated from the candidate depth; if DSF_{03} and DSF_{13} are less than 0.125, we traverse only interval [2, 2].

3.4. Depth Range Correction Algorithm

The algorithm devised in section 3.2 would result in errors in predicting sharp scene shifting sequences. Therefore, a correction algorithm, CA, was invented to deal with this drawback. Shown in Fig. 3, co-located LCU in adjacent two frames, together with the left and upper LCU of current LCU were adopted to determine depth range to traverse through. The weighed formula is defined as follows:

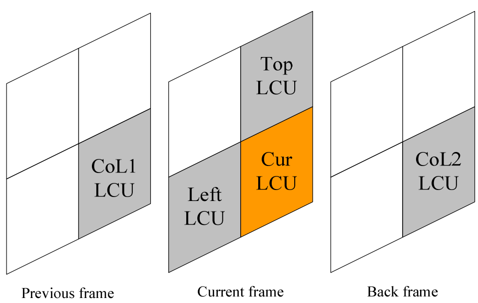


Fig. 3. Temporal-spatial similarity of LCU.

$$CU_i = \sum_{j=0}^{3} (w_j \times count_{ij} / N), \quad (7)$$

where i represents CU depth, CU stands for percentage of each CU depth; j is the index of reference LCU, with 0 standing for co-located LCU in the previous frame, 1 for LCU in the back frame, 2 and 3 for left and top LCU of current LCU respectively. $count_{ij}$ is the percentage of each CU depth i in LCU j. w_j is the weight of LCU j. w_0 and w_1 are set 0.3, while w_2 and w_3 are set 0.2, N symbolizes 4×4 number in one LCU, that is 256.

When CU_0, CU_1, CU_2 and CU_3 are known, those depth percent coefficients that are less than 0.1 will be removed from candidates. The rest CU depth will be traversed through to lower coding complexity.

4. Experimental Results and Discussions

To test the effectiveness of the proposed algorithm, HM9.0 [14] was selected as the test model. The experiments were conducted on a PC with configurations as follows: CPU Intel core i5-2500, clock speed 3.30 GHz, RAM 8 G, operating system Windows 7 and developer tool Microsoft Visual Studio 2008. 100 frames of each sequence were tested with random access mode [15] and with QP including 22, 27, 32, 37. Experimental results were presented in the form of BD-PSNR, (Bjøntegaard delta peak signal-to-noise rate), BDBR (Bjøntegaard delta bit rate) [16] and ΔT . BDPSNR indicates PSNR difference at the same bitrate, while BDBR shows bitrate variance at the same PSNR level. ΔT (%) is defined as follows

$$\Delta T = (T_p - T_{HM}) / T_{HM} \times 100\%, \quad (8)$$

where T_P and T_{HM} represents coding time of the proposed algorithm and original HM algorithm respectively.

Twelve sequences were tested on common test conditions. In the meantime, Shen's method was realized and compared with this paper's ACUDR, ACUDR+DSF and CA algorithms. Table 4 shows the comparison among ACUDR, CA and Shen's algorithms. Table 5 reveals the difference between ACUDR+DSF and Shen's method.

Table 4. Comparison results among Shen's method, ACUDR and CA algorithms.

Type	Sequence	Shen			Proposed method					
		ACUDR			ACUDR			CA		
		BDPSNR (dB)	BDBR (%)	ΔT (%)	BDPSNR (dB)	BDBR (%)	ΔT (%)	BDPSNR (dB)	BDBR (%)	ΔT (%)
Class A (2560×1600)	Traffic	-0.031	1.01	-37.57	-0.025	0.82	-36.42	-0.024	0.80	-32.42
	PeopleOnStreet	-0.025	0.63	-22.15	-0.019	0.47	-22.54	-0.028	0.70	-28.13
Class B (1920×1080)	BasketballDrive	-0.022	1.23	-33.85	-0.018	0.89	-32.73	-0.012	0.64	-30.73
	Cactus	-0.012	0.60	-32.22	-0.009	0.45	-31.69	-0.012	0.57	-29.53
Class C (832×480)	BasketballDrill	-0.061	1.68	-22.96	-0.046	1.27	-22.86	-0.025	0.69	-20.61
	BQMall	-0.078	2.01	-22.56	-0.047	1.22	-22.13	-0.021	0.55	-20.33
Class D (416×240)	BasketballPass	-0.050	1.15	-14.68	-0.003	0.094	-11.10	0.003	-0.076	-13.53
	BlowingBubbles	-0.009	0.25	-10.66	-0.033	0.74	-14.65	-0.004	0.11	-10.66
Class E (1280×720)	KristenAndSara	-0.029	1.07	-47.95	-0.014	0.53	-46.31	-0.014	0.55	-37.36
	Vidyo4	-0.038	1.48	-43.78	-0.029	1.16	-43.14	-0.004	0.15	-40.10
Class F	BasketballDrillText	-0.07	1.80	-22.73	-0.051	1.31	-22.28	-0.021	0.55	-20.75
	ChinaSpeed	-0.09	1.67	-26.67	-0.055	1.03	-26.46	-0.017	0.32	-28.19
Average		-0.043	1.22	-28.15	-0.029	0.83	-27.69	-0.015	0.46	-26.03

Table 5. Comparison results between ACUDR+DSF and Shen's algorithms.

Type	Sequence	Shen			ACUDR+DSF		
		BDPSNR (dB)	BDBR (%)	ΔT (%)	BDPSNR (dB)	BDBR (%)	ΔT (%)
Class A (2560×1600)	Traffic	-0.031	1.01	-37.57	-0.11	3.49	-41.96
	PeopleOnStreet	-0.025	0.63	-22.15	-0.047	1.18	-36.25
Class B (1920×1080)	Kimono	-0.010	0.33	-32.59	-0.037	1.27	-43.70
	Cactus	-0.012	0.60	-32.22	-0.034	1.70	-39.31
Class C (832×480)	PartyScene	-0.020	0.45	-20.79	-0.048	1.13	-31.96
	RaceHorsesC	-0.018	0.52	-15.86	-0.041	1.16	-27.34
Class D (416×240)	RaceHorses	-0.002	0.04	-11.63	-0.031	0.67	-21.07
	BlowingBubbles	-0.009	0.25	-10.66	-0.022	0.62	-19.53
Class E (1280×720)	Vidyo3	-0.017	0.56	-42.31	-0.049	1.71	-46.89
	FourPeople	-0.021	0.64	-42.23	-0.059	1.76	-45.89
Class F	BasketballDrillText	-0.07	1.80	-22.73	-0.08	2.06	-32.80
	ChinaSpeed	-0.09	1.67	-26.67	-0.078	1.46	-38.02
Average		-0.027	0.71	-26.45	-0.053	1.52	-35.39

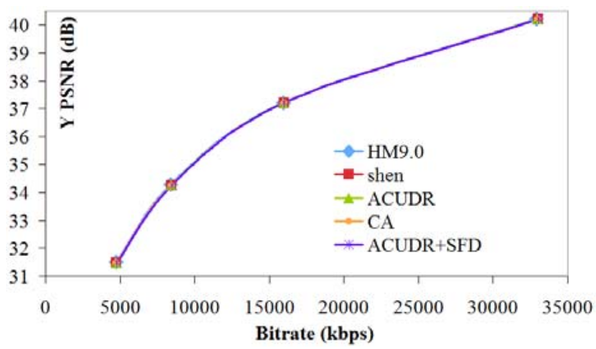
From Table 4, it can be seen that compared with HM9.0, Shen's methods saved 28.15 % coding time, reduced 0.043 dB PSNR and increased BDBR by 1.22 %. The proposed ACUDR algorithm, took up nearly the same coding time while BDPSNR increased by 0.014 dB and BDBR decreased by 0.39 % compared with Shen's method. CA algorithm was employed to deal with sharp scene shifting sequences. In comparison with Shen's, it cost comparable coding time with BDPSNR increasing by 0.028 dB and BDBR decreasing by 0.76 %. CA is especially applicable to BasketballDrill, BQMall and Class F sequences in that it predicted LCU depth range more accurately. According to Table 5, the combination of ACUDR+SFD saved 35.39 % time when compared with original HM9.0 algorithms. And it occupied 10 % less time than Shen's did without obvious increase in BDBR and decrease in BDPSNR. Fig. 4 shows rate distortion performance of sequences PeopleOnStreet, Kimono, PartyScene and BlowingBubbles tested with each algorithm (HM9.0, Shen's, ACUDR, CA and ACUDR+DSF).

From Fig. 4, it is clear that all the algorithms possessed comparable rate distortion performance, indicating some reliability of the proposed algorithms. Fig. 5 shows the LCU partitions of each scheme. The red part demonstrates the mismatches with the original partitions.

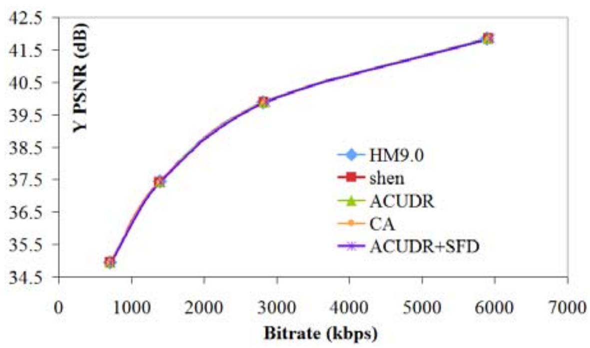
From Fig. 5, CA partition result is closer to the original one than Shen's. There only exist several mismatches, which have similar CU depths with original ones. Therefore, the global rate distortion performance stays stable.

5. Conclusions

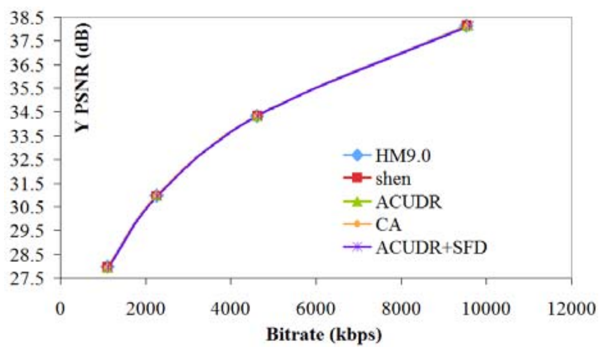
This paper proposes an adaptive CU depth selection algorithm, aiming to improve Shen's methods. Firstly, average depth error was adopted to decide the weight of adjacent LCU and co-located LCU, which was later used to determine LCU DR type. Secondly, depth scaling factors of the adjacent frames were used to shrink DR. Finally, with respect to sharp scene change sequences, a depth range correction algorithm was devised, which took advantage of neighboring frames depth scaling factors. Results showed that the proposed ACRDR+SFD took up 10 % time less than Shen's did while BDBR increased by about 0.81 % and BDPSNR decrease by 0.026 dB. Nevertheless, CA cost nearly equivalent time, with 0.76 % lower BDBR and 0.028 dB increase in BDPSNR. Further researches on inter prediction modes to avoid unnecessary mode traversal, thus lowering coding complexity.



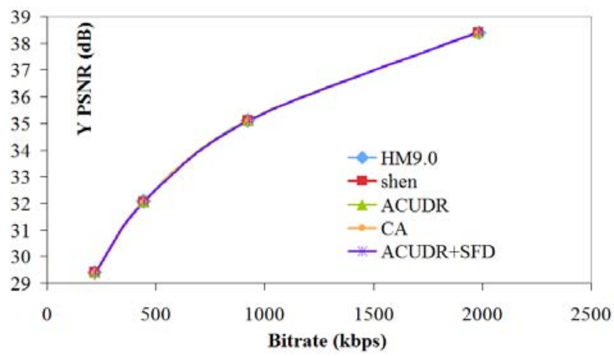
(a) PeopleOnStreet



(b) Kimono

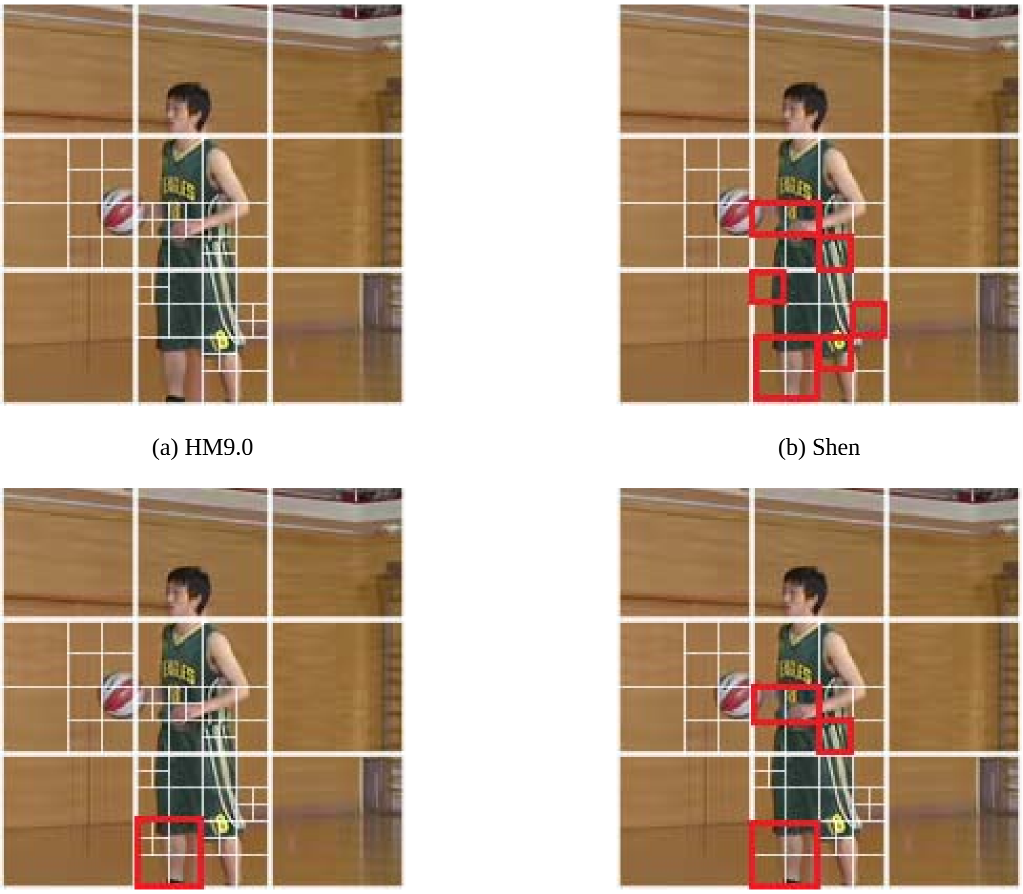


(c) PartyScene



BlowingBubbles

Fig. 4. Rate distortion performance of sequences.



(a) HM9.0 (b) Shen

(c) CA (d) ACUDR+DSF

Fig. 5. Quad-tree partitioning results of BasketballPass sequence when QP is equal to 32.

Acknowledgements

The research work was supported by National Natural Science Foundation of China (Grant No. U1301257, 61311140262, 61271270, 61171163), and Zhejiang Scientific and Technical Key Innovation Team (Grant No. 2010R50009).

References

[1]. Masoumi Saeid, Tabatabaei Raziyeh, Feizi-Derakhshi Mohammad-Reza, Tabatabaei Khatereh, A New Parallel Algorithm for Frequent Pattern Mining, *Journal of Computational Intelligence and Electronic Systems*, Vol. 2, Issue 1, 2013, pp. 55-59.

[2]. N. K. Mittal, Ahmed Mohd, Naaz Farha, Efficient palmprint recognition system implementation using 2D Gabor filters, *Journal of Computational Intelligence and Electronic Systems*, Vol. 1, Issue 2, 2012, pp. 154-160.

[3]. Y. X. Zhang, Y. Lu, Z. Luan, et al, Design and implementation of video image system, *Liquid Crystal and Display*, Vol. 28, Issue 3, 2013, pp. 424-428.

[4]. Y. H. Wu, L. X. Jin, K. Zhang, et al, Fast P frame mode selection algorithm based on monotonicity of Jmode value, *Liquid Crystal and Display*, Vol. 28, Issue 2, 2013, pp. 266-272.

[5]. T. Wiegand, J. R. Ohm, G. J. Sullivan, et al, Special section on the joint call for proposals on high efficiency video coding standardization, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 20, Issue 12, 2010, pp. 1661-1666.

[6]. T. Wiegand, G. J. Sullivan, G. Bjontegaard, et al, Overview of the H.264/AVC video coding standard, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, Issue 7, 2012, pp. 560-576.

[7]. G. J. Sullivan, J. R. Ohm, W. J. Han, et al, Overview of the high efficiency video coding (HEVC) standard, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, Issue 12, 2012, pp. 1649-1668.

[8]. X. Li, J. C. An, X. Guo, et al, Adaptive CU depth range, in *Proceedings of the 5th JCT-VC Meeting*, Geneva, Switzerland, 16-23 March 2011, Article ID JCTVC-E090.

[9]. L. Q. Shen, Z. Liu, X. P. Zhang, et al, An effective CU size decision method for HEVC encoders, *IEEE Transactions on Multimedia*, Vol. 15, Issue 2, 2013, pp. 465-470.

[10]. J. Kim, Y. Choe and Y. G. Kim, Fast coding unit size decision algorithm for intra coding in HEVC, in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE'2013)*, Las Vegas, USA, 11-14 January 2013, pp. 637-638.

[11]. Q. Yu, X. F. Zhang, S. Q. Wang, S. W. Ma, Early termination of coding unit splitting for HEVC, in *Proceedings of the Signal & Information Processing Association Annual Summit and Conference (APSIPA*

ASC), Hollywood, USA, 3-6 December 2012, pp. 1-4.
[12]. T. L. da Silva, L. V. Agostini, L. A. da Silva, Fast HEVC intra prediction mode decision based on edge direction information, in *Proceedings of the European Signal Processing Conference*, Bucharest, Romania, 27-31 August 2012, pp. 1214-1218.
[13]. B. Bross, W. J. Han, J. R. Ohm, et al, High efficiency video coding text specification draft 10 (for FDIS & Consent), in *Proceedings of the 5th JCT-VC Meeting*, Geneva, Switzerland, 14-23 January 2013, Article ID JCTVC-L1003.
[14]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. (https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-9.0).
[15]. F. Bossen, Common test conditions and software reference configurations, in *Proceedings of the 5th JCT-VC Meeting*, Geneva, Switzerland, 14-23 January 2013, Article ID JCTVC-L1003.
[16]. G. Bjontegaard, Calculation of average PSNR differences between RD-curves, in *Proceedings of the 13th Video Coding Experts Group (VCEG) Meeting*, Austin, American, 2-4 April 2001, pp. 1-4.