# Adaptive search range by neighbouring depth intensity weighted sum for HEVC texture coding

Tsz-Kwan Lee, Yui-Lam Chan and Wan-Chi Siu

High efficiency video coding (HEVC) outperforms H.264/AVC by providing a bitrate reduction of about 50% while having almost the same perceptual quality. It adopts more flexible partitioning in motion estimation (ME) which gains higher coding efficiency at a cost of increased coding complexity. This letter exploits depth maps in the emerging multi-view plus depth (MVD) videos to adjust the search range in ME for HEVC complexity reduction. With the aid of depth intensity variation among neighbouring blocks, the proposed algorithm establishes an adaptive search range according to a weighted sum of the motion vectors from the neighbouring blocks. The weights are then derived from their depth variations. Compared to the fast test zone search (TZS) in HEVC, the proposed algorithm saves 65% coding time in ME on average with insignificant rate-distortion degradation.

*Introduction:* The latest High Efficiency Video Coding (HEVC) standard achieves about 50% bitrate reduction with similar perceptual quality compared to its predecessor H.264/AVC [1]. The coding gain is mainly from its more flexible block partitioning in motion estimation (ME), which is especially crucial for coding high resolution 3D videos in the multi-view plus depth (MVD) format [2]. However, the flexible block partitioning mechanism in HEVC induces more ME computations. In hybrid video coding, ME performs block-based search for every location within a pre-defined search range [2]. With the motion vector predictor *MVP* from a neighbouring block as the search centre, the optimal motion vector *MV* is selected by minimizing the rate-distortion cost within the pre-defined search range. The true motion vector *TMV* of the current block is then formed by

$$TMV = MVP + MV \tag{1}$$

HEVC utilizes an advanced motion vector predictor (AMVP) for the determination of *MVP* to a block as an initial search centre. With a fixed search range of 64 pixels for both full-search (FS) and fast test zone search (TZS) integer-pixel ME, *MV* is obtained from a range of [-64, +64]. TZS is one of the fast ME algorithms adopted in the HEVC test model [3] by restricting the number of search locations. In TZS, a diamond or square search pattern with various sizes is used for its centre search point initialization. However, the multiple initial search point selection is still a major burden on TZS. Other works focus on applying specific search patterns or directional search to reduce search points within a fixed search range [4]. Nevertheless, various search patterns bring irregular data flow which is not preferable for hardware implementation [5]. Besides, spatial neighbouring blocks contain highly homogenous contents to the current block, AMVP is therefore selected among their motion vectors (MV). It implies if *MVP* is very similar to *TMV*, *MV* becomes very small as stated in (1). In this circumstance, the search range can be reduced adaptively. Unnecessary search point computations can therefore be avoided for saving coding time. An adaptive search range (ASR) algorithm can then deliver both search point reduction and regular data flow.

*Adaptive search range:* Some existing ASR algorithms correlate the search range of the current block with the motion characteristics of its neighboring blocks. In [5], Cauchy distribution is used to model the search range for one frame and MV differences in the neighboring blocks are used to adjust the search range for the block being encoded. In [6], the maximum difference of the estimated TMV and the optimal AMVP is used to give the ASR in HEVC. Such ASR, however, can only be determined from the results of AMVP selection. In [7], MV in the co-located block is used to define the ASR without considering whether the co-located block is within the same object. The most recent ASR algorithm in [8] adopts a linear adaptive search range model (LAM) with an overdetermined equation system. The parameters in the system can be solved if the size of prediction units (PU), MVs, and predictors are given. The ASR is then adjusted by a fixed scale factor. To the best of our knowledge, no work has noted so far to adopt the new features provided in MVD videos for defining an ASR. In this letter, we propose to make use of depth maps in MVD videos and MVs from neighbouring blocks to yield the ASR algorithm for HEVC.

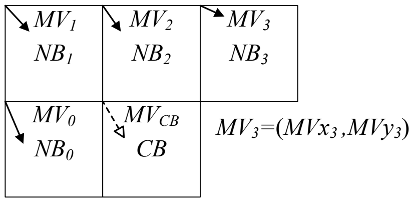


**Fig. 1** *Illustration of spatial neighboring blocks with high motion homogeneity to current block and their associated motion vectors*

*Proposed algorithm:* An object in a video frame always occupies a region covered by several blocks. It is obvious that spatial neighbouring blocks contain highly homogenous contents to the current block. Consequently, MVs of spatial neighbouring blocks can be utilized to estimate the motion range of the current block. In the proposed ASR algorithm, an adaptive search range is adjusted by determining whether the current block and its spatially neighbouring blocks belong to the same object. The correlation among MVs in the same object can then be employed to specify the new search range adaptively. Depth maps associated with every pixel of the color textures in MVD videos capture the distances of the objects from the camera. It implies that depth maps can be used to distinguish blocks within the same object. Therefore, in this letter, depth information is suggested to be a good feature to exploit the correlation between MVs of spatially neighbouring blocks for generating ASR.

The ASR algorithm proposed in this letter takes a weighted sum of the neighbouring blocks' MVs to predict the search range of the current block *CB* in order to reduce unnecessary computations in ME. In Fig. 1, $MV_i$ is the motion vector with two components ($MVx_i$, $MVy_i$) in the horizontal and vertical directions respectively from a neighbouring block $NB_i$ where $i$ = 0, 1, 2, and 3. The new ASR of the current block $asr_x(CB)$ and $asr_y(CB)$ for the horizontal and vertical directions, respectively, can then be estimated from $MV_i$ in Fig. 1, and can be written as

$$asr_x(CB) = \left\lceil \sum_{i=0}^{3} |MVx_i| \cdot \frac{\beta_i}{\sum_{j=0}^{3} \beta_j} \right\rceil \tag{2}$$

$$asr_y(CB) = \left\lceil \sum_{i=0}^{3} |MVy_i| \cdot \frac{\beta_i}{\sum_{j=0}^{3} \beta_j} \right\rceil \tag{3}$$

In (2) and (3), a weighted factor $\beta_i$ of each $NB_i$ depends on how relevant its $MVx_i$, and $MVy_i$ to $asr_x(CB)$ and $asr_y(CB)$ of *CB* respectively, and it makes use of depth information in MVD videos. Here, $\beta_i$ can be formulated by

$$\beta_i = e^{-|D_i|} \tag{4}$$

where $\beta_i$ is the output of the exponential decay function of which $e$ is Euler's number with the decay rate $\frac{1}{e}$. The exponent $|D_i|$ is the absolute difference in average depth intensity values between the neighbouring block $NB_i$ and the current block *CB*. Values with small $D_i$ output an exponentially high $\beta_i$ and vice versa. Smaller depth intensity difference hints that the probability of $NB_i$ and *CB* comprising the same object is high, which reflects this $NB_i$ is more correlated to *CB*. The rationale is

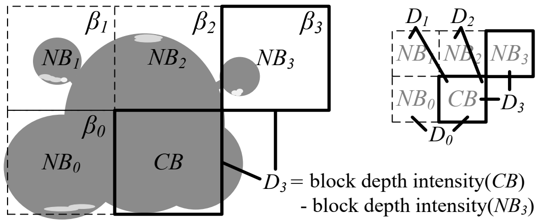


**Fig. 2** *Neighboring blocks with higher similarity in block depth intensity are very likely representing the same object*

that a depth map can presumably reveal the object distance within a 3D space. Therefore, it is a piece of indicative information to decide which blocks belong to the same object as illustrated in Fig. 2. In the proposed algorithm, a higher weight $\beta_i$ will be issued to *MV* in which its associated block $NB_i$ represents higher similarity of the average depth value to *CB* (i.e. a smaller value of $|D_i|$). From the example in Fig. 2, $|D_0| < |D_2| < |D_1| < |D_3|$ is observed. It can be concluded that $\beta_0 > \beta_2 > \beta_1 > \beta_3$. Therefore $NB_0$ is closest to *CB* in terms of depth distance and contents. The ASR is then determined based on the amplitude of the weighted MVs of the neighbouring blocks (i.e. $MV_0$ has a stronger influence than other MVs in this example). Finally, MV of *CB*, $MV_{CB}$, can be obtained by ME with the horizontal and vertical search ranges as $[-asr_x(CB), +asr_x(CB)]$ and $[-asr_y(CB), +asr_y(CB)]$, respectively.

**Table 1:** Performance evaluation of proposed NDIWS to conventional fixed search range FS and TZS, and existing fast algorithm LAM [8] in HM14.0. Resolution 720p (S1: Balloons, S2: Kendo, S3: Lovebird1, S4: Newspaper) and resolution 1080p (S5: Poznan_Hall2, S6: Undo_Dancer, S7: GT_Fly)

| Compared to | | Sequences | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | S1 | S2 | S3 | S4 | S5 | S6 | S7 | Avg. |
| **NDIWS** | | | | | | | | | |
| FS using Fixed SR | BD-PSNR (dB) | -0.01 | -0.03 | -0.01 | -0.01 | -0.01 | -0.05 | -0.02 | -0.02 |
| | BD-rate (%) | +0.38 | +0.85 | +0.24 | +0.43 | +0.56 | +1.43 | +0.60 | +0.64 |
| | Δtime (%) | -96.04 | -98.50 | -99.26 | -98.47 | -92.22 | -91.06 | -89.65 | -95.03 |
| FS using LAM [8] | BD-PSNR (dB) | -0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | BD-rate (%) | +0.25 | +0.06 | -0.07 | +0.13 | +0.02 | -0.03 | +0.05 | +0.06 |
| | Δtime (%) | -68.18 | -39.71 | +5.13 | -40.47 | -21.56 | -38.19 | -20.17 | -31.88 |
| **TZS+NDIWS** | | | | | | | | | |
| TZS using Fixed SR | BD-PSNR (dB) | -0.01 | -0.03 | -0.02 | -0.02 | 0.00 | -0.03 | -0.01 | -0.02 |
| | BD-rate (%) | +0.36 | +1.09 | +0.49 | +0.54 | +0.13 | +0.90 | +0.19 | +0.53 |
| | Δtime (%) | -73.07 | -67.89 | -84.71 | -82.83 | -67.81 | -43.30 | -40.87 | -65.78 |
| TZS using LAM [8] | BD-PSNR (dB) | 0.00 | -0.01 | -0.01 | -0.01 | 0.00 | +0.01 | +0.01 | 0.00 |
| | BD-rate (%) | +0.07 | +0.42 | +0.37 | +0.40 | +0.12 | -0.32 | -0.19 | +0.12 |
| | Δtime (%) | -37.86 | -29.63 | -54.29 | -55.63 | -33.45 | -18.44 | +40.69 | -26.94 |

*Simulation Results:* The proposed ASR algorithm using neighbouring depth intensity weighted sum has been integrated into the HM 14.0 reference software, and is referred to as NDIWS. Its asymmetric ASR for FS was compared with the conventional FS using a fixed search range of [-64, +64] and the most recent LAM algorithm for ASR [8]. It is noted that TSZ is designed for squared search windows. Therefore, the search range using NDIWS was computed as $\max\left(asr_x(CB), asr_y(CB)\right)$ when TSZ with NDIWS (TSZ+NDIWS) was tested, where *max( )* is the maximum function aiming at bounding all probable movement among the *x* and *y* directions. TSZ+NDIWS was further compared to the conventional TZS with the fixed search range and the adaptive search range determined by LAM [8]. All tested algorithms were evaluated with four QPs of 22, 27, 32, and 37 under the low-delay P configuration specified in the common test condition of HEVC [3]. Full quad-tree structure for all CU, PU, and TU was utilized. Bjontegaard (BD) measurement in terms of BD-rate (%) and BD-PSNR (dB) were used to measure the average coding efficiency, and Δtime (%) represents coding time change in percentage as compared with the benchmarking algorithms. Positive and negative values denote increments and decrements, respectively. The test platform used for simulations was a 64-bit MS Windows 8.1 OS running on an Intel Core i7-4770 CPU of 3.4 GHz and 16.0 GB RAM.

The upper part of Table 1 lists the performance of NDIWS compared to FS with the fixed search range. It averagely saves 95% coding time over FS while its BD-PSNR drops 0.02dB and its BD-rate increases by 0.64%. In comparison to LAM [8], NDIWS saves encoding time by 31% while only introducing an insignificant BD-rate increase of 0.06%. The proposed NDIWS saves more time as its asymmetric search range considers movement in the horizontal and vertical directions separately. It is due to the fact that most of the objects do not move diagonally. By integrating the proposed ASR into TZS, TSZ+NDIWS reduces 65% time on average compared to the conventional fixed search range TZS. The corresponding BD-PSNR decreases by 0.02dB while the BD-rate increases by 0.53%. As compared with TZS using LAM [8], TSZ+NDIWS reduces averagely 26% of coding time while only introducing an insignificant BD-rate increase of 0.12%. The reason is that LAM [8] only formulates a fixed relationship of the motion information by offline trainings. Instead, TSZ+NDIWS utilizes the depth intensity correlation for the current block from its neighbouring blocks adaptively in order to reduce unnecessary computations with insignificant BD loss.

*Conclusion:* In this letter, an adaptive search range selection algorithm was proposed by considering depth information in MVD videos. The ASR is determined by a weighted sum of the neighbouring blocks' MVs in which their weights depend on the absolute difference in average depth intensity values between the neighbouring blocks and the current block. It results in a complexity reduction in ME. The proposed ASR is compatible with FS and other fast search algorithms such as TZS in HEVC. Simulation results demonstrated that it is able to reduce 65% of coding time on average among various sequences in the fast TZS algorithm with negligible loss on BD performance.

*Acknowledgments:* The work was supported by Centre for Signal Processing, HKPolyU, and a grant from the Research Grants Council of the HKSAR, China (PolyU 5119/13E). Tsz-Kwan Lee acknowledges the research studentships provided by the University.

Tsz-Kwan Lee, Yui-Lam Chan and Wan-Chi Siu (*Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong*)
E-mail: enylchan@polyu.edu.hk

**References**

1 Vanne, J., Viitanen, M., Hämäläinen, T.D., and Hallapuro, A.: 'Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs', *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**, (12), pp. 1885–1898

2 Sullivan, G.J., Boyce, J.M., Chen, Y., Ohm, J.R., Segall, C.A., and Vetro A.: 'Standardized extensions of high efficiency video coding (HEVC)', *IEEE J. Sel. Top. Sign. Proces.*, 2013, **7**, (6), pp. 1001–1016

3 Kim, I.K., McCann, K., Sugimoto, K., Bross, B., Han, W.J., and Sullivan, G. J.: 'High Efficiency Video Coding (HEVC) Test Model 14 (HM14) Encoder Description', Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-P1002, San Jose, US, March 2014

4 Yang, S.H., Jiang, J.Z., and Yang, H.J.: 'Fast motion estimation for HEVC with directional search', *Electronics Letters*, 2014, **50**, (9), pp. 673-675

5 Dai, W., Au, O.C., Li, S., Sun, L., and Zou, R.: 'Adaptive search range algorithm based on cauchy distribution', Proc. IEEE Visual Commun. and Image Process. (VCIP), November 2012, pp. 1–5

6 Chien, W.D., Liao, K.Y., and Yang J.F.: 'Enhanced AMVP mechanism based adaptive motion search range decision algorithm for fast HEVC coding', Proc. IEEE Int. Conf. on Image Process. (ICIP 2014), October 2014, pp. 3696–3699

7 Kim, S., Lee, D.K., Sohn, C.B., and Oh, S.J.: 'Fast motion estimation for HEVC with adaptive search range decision on CPU and GPU', Proc. IEEE China Summit & Int. Conf. on Signal and Information Process. (ChinaSIP 2014), July 2014, pp. 349–353

8 Du, L., Liu, Z., Ikenaga, T., and Wang, D.: 'Linear adaptive search range model for uni-prediction and motion analysis for bi-prediction in HEVC', Proc. IEEE Int. Conf. on Image Process. (ICIP 2014), October 2014, pp. 3671–3675