MODE-DEPENDENT PIXEL-WISE MOTION REFINEMENT FOR HEVC

Lei Zhao, Jian Zhang, Siwei Ma and Wen Gao

Institute of Digital Media & Cooperative Medianet Innovation Center
Peking University, Beijing, P. R. China

ABSTRACT

High efficiency video coding (HEVC) standard is within the block-based hybrid coding framework, which essentially adopts prediction unit (PU) as the basic motion compensation unit. However, in the case of tiny motion, the actual motion vectors (MVs) for each sample may differ from the PU's MV, thus resulting in more residual energy. In this paper, a novel pixel-wise motion refinement method (PMR) is presented by extending the traditional bi-directional optical flow (BIO) to PUs with two uni-directional reference blocks. To get the robust MV for each sample, a median filtering process is introduced to MV shifting values of the neighboring pixels. Furthermore, a PU level mode-dependent pixel-wise motion refinement (MPMR) scheme is also presented to improve the coding performance. Simulation results demonstrate that the proposed method achieves 0.83% bitrate reduction on average for HEVC test sequences under HM12.0 low delay B (LDB) configuration, and achieves up to 4.92% bitrate reduction for sequences with complex motion, e.g. affine motion.

Index Terms— Video coding, HEVC, pixel-wise motion refinement, median filter, mode dependent

1. INTRODUCTION

The latest generation of video coding standard, High Efficiency Video Coding (HEVC) [1] [2], developed by the Joint Collaborative Team on Video Coding, jointly established by the International Telecommunication Union–Telecommunication (ITU-T) Video Coding Experts Group (VCEG) and International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) Moving Picture Experts Group (MPEG), was finalized in January 2013. As the successor of previous video coding standards, HEVC still adopts block-based coding framework, including block based motion estimation, block based motion compensation (ME/MC) and block based transform. As an essential process, ME/MC constitutes the foundation of video coding system, which drastically affects the coding performance. The basic ME/MC units utilized in HEVC are prediction units (PUs), of which the size range from 64x64 to 8x8 in inter prediction. HEVC employs 1/4 pel MV resolution in ME process, and some adaptive MV resolution schemes were proposed in [3][4]. Although block based MC is employed in HEVC for its simplicity, robustness and ease of implementation, the drawbacks are also obvious. For instance, in the case of tiny motion within the PU, the actual MV of each sample may differ from each other, and hence it fails to find an accurate match for blocks containing edges and complex textures. Consequently, the transform coefficients tend to contain more high-frequency energy, and the reconstructed frames usually suffer from block artifacts [5].

To deal with the problem mentioned above, a more efficient inter prediction approach is highly desired to accurately predict each pixels of the current block. In [6], a pixel wise sub-pixel ME method was proposed, combining block based ME and optical flow, the fractional pixel MV can be estimated without interpolation, which significantly reduces the coding complexity. In [5], a motion vector fields based video coding frame is presented, achieving comparable objective performance compared with HEVC. In [7], similar work was presented, A. Alshin et al. proposed a pixel wise motion refinement method called Bi-directional Optical flow (BIO), by combining the optical flow [8] concept and high accuracy gradients evaluation. As a combination of block wise and pixel wise MC algorithm, this method can handle the tiny movement within the block effectively, and hence, remarkable coding gain can be achieved. In [9], a strict mathematical explanation to BIO is provided by introducing Hermit interpolation, which regards the weighted average of two reference blocks as a pixel interpolation process.

However, in [7] [9], BIO was only applied for PUs that have "truly" bi-directional prediction blocks, i.e., forward direction and backward direction, which made BIO only compatible with Random Access (RA) configuration in KTA [10]. Moreover, as BIO is a MV refinement process by the mathematical estimation, the actual effects are not stable enough. To address the problems existing in the BIO algorithm, in this paper, we extend BIO to the PUs that have two forward reference blocks so that BIO is compatible with Low-delay-B (LDB) configuration. As an improvement, a median filtering process is conducted to MV shifting values of the neighboring pixels. Besides, we propose a PU level mode-dependent pixel-wise motion refinement (MPMR) scheme to further improve the coding performance. Experimental results show that our proposed scheme demonstrates substantially superior performance in terms of rate-distortion performance.

The rest of the paper is organized as follows. Section 2 reviews the BIO algorithm. Our proposed methods are presented in Section 3. Experimental results and analysis are given in Section 4 and the conclusion is drawn in Section 5.

2. REVIEW ON BIO

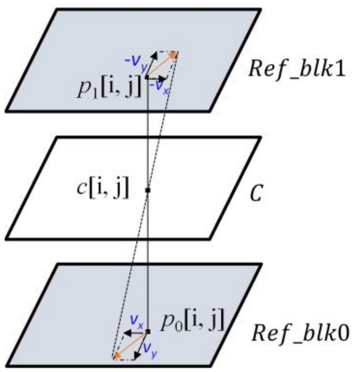


Fig. 1 Bi-directional optical flow.

Suppose a bi-prediction PU ***C*** in B slice, as depicted in Fig. 1, which has found two best bi-directional prediction blocks ***Ref_blk*0** and ***Ref_blk*1**. Note that ***Ref_blk*0** and ***Ref_blk*1** locate in the opposite time direction relative to PU ***C***. After that, the conventional bi-directional prediction in HEVC ***PRED*** for PU ***C*** is formulated as follows, which can be regarded as a linear interpolation process:

$$PRED_{HEVC}(c[\mathrm{i,j}]) = (p_0[\mathrm{i,j}] + p_1[\mathrm{i,j}])/2 \quad (1)$$

where p_0[i, j] and p_1[i, j] are the prediction samples from block ***Ref_blk*0** and ***Ref_blk*1**, respectively, for the sample *c*[i, j] in PU ***C***.

Since the values of reference pixels have derived through motion estimation, and the derivatives of reference pixels can also be estimated, the interpolation process can be optimized by Cubic Hermit interpolation as it matches both value of the nodes and the value of derivatives. Adopting the conception of optical flow and Hermit interpolation, the prediction for *C* is modified as [9]:

$$PRED_{BIO}(c[\mathrm{i,j}]) = (p_0[\mathrm{i,j}] + p_1[\mathrm{i,j}] + (v_x(I_x^{(0)}[\mathrm{i,j}] - I_x^{(1)}[\mathrm{i,j}]) + v_y(I_y^{(0)}[\mathrm{i,j}] - I_y^{(1)}[\mathrm{i,j}]))/2)/2 \quad (2)$$

where $\left(I_x^{(0)}[\mathrm{i,j}], I_y^{(0)}[\mathrm{i,j}]\right)$, $\left(I_x^{(1)}[\mathrm{i,j}], I_y^{(1)}[\mathrm{i,j}]\right)$ represent the gradients of luminance in position [i, j] on ***Ref_blk*0** and ***Ref_blk*1** respectively, and v_x, v_y denote the shifting values relative to the original position. However, as original BIO in [7] only applies to "true" bi-directional PUs, and hence it is only used with RA configuration in KTA.

3. PROPOSED METHOD

3.1 Pixel-wise Motion Refinement (PMR)

In this section, we extent BIO to the PUs with two forward reference blocks. Firstly, if the two forward predictions locates in different frames, a virtual reference frame is involved to illustrate the rationality. Suppose a virtual block ***Ref_virtual*** as shown in Fig. 2, which locates in the middle of ***Ref_blk*0** and ***Ref_blk*1**. Let's assume ***Ref_virtual*** is the best prediction for ***C*** among all the candidates, as ***Ref _virtual*** doesn't really exist due to the restriction of frame rate, it can only be derived by an interpolation process using ***Ref_blk*0** and ***Ref_blk*1**. Based on this assumption, the prediction of two forward blocks can also be treated as an interpolation process, which is consistent with the BIO model in [9]. Accordingly, it is reasonable to apply BIO to PUs with uni-directional predictions in different reference frames. Secondly, if the two reference blocks locate in the same frame, the above interpolation model is not valid, however, as BIO in [7] can also be treated as a post process that minimizes the difference between two reference pixels, which may also has positive effects on MC process, and hence the BIO is also reasonable to be applied in this case. Consequently, in this paper BIO is extended to PUs with two uni-directional reference blocks regardless of the location of the two predictions, and we called it PMR.

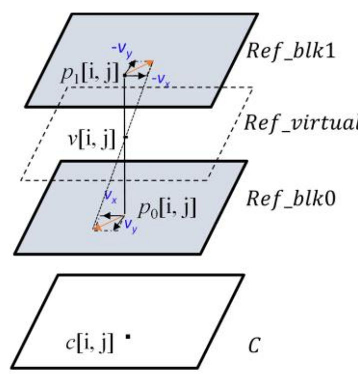


Fig. 2 Pixel-wise motion refinement (PMR) for PU with uni-directional predictions

3.2 Median filtering process

Essentially, PMR changes the final prediction value of each pixel so it can be regarded as a filtering process, the derivation of shifting value [v_x, v_y] acts as an intermediate process which significantly affects the performance of PMR.

However, as a pixel wise MV refinement, the derivation of [v_x, v_y] would inevitably bring some unpredictable errors, which may deteriorate the refine process. To restrain the influence of these noisy offsets, a filtering process for [v_x, v_y] is conducted. Fig. 3 depicts the filtering process, when the shifting values of four connected pixels have derived, i.e. $\Delta mv1(v_{x1}, v_{y1})$, $\Delta mv2(v_{x2}, v_{y2})$, $\Delta mv3(v_{x3}, v_{y3})$ and $\Delta mv4(v_{x4}, v_{y4})$, a filtering process is initiated, and the average of horizontal components and vertical components then calculated, as (3) shows:

$$\begin{aligned} v_{xave} &= (v_{x1} + v_{x2} + v_{x3} + v_{x4}) / 4 \\ v_{yave} &= (v_{y1} + v_{y2} + v_{y3} + v_{y4}) / 4 \end{aligned} \quad (3)$$

$\Delta mv\_ave(v_{xave}, v_{xave})$ denote the filtered offsets that will be assigned to the four pixels as the final offset values.

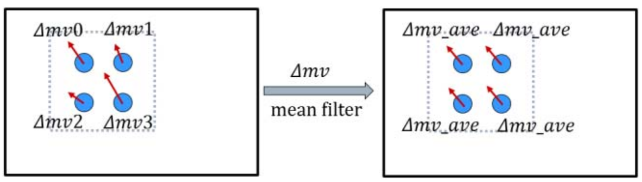
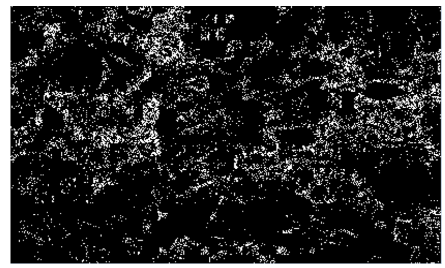


Fig. 3 Median filtering process.

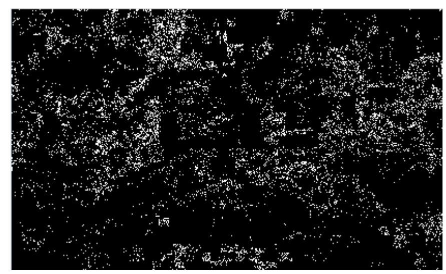
3.3 Mode-dependent Pixel-wise Motion Refinement (MPMR)

In this section PMR is further improved. Since PMR is a MV refinement process by mathematical estimation, the actual effects are not stable enough, which may deteriorate the prediction results. Fig. 4 shows the analysis results of 18th frame in *BQSquare_416x240*, where white area in Fig. 4(a) represents the pixels on which PMR has positive influence, and white area in Fig. 4(b) represents the pixels on which PMR has negative influence. This analysis inspires us to control PMR by using a switch. Considering that the basic unit of motion compensation (MC) in PMR is PU, we intuitively set a flag to indicate that whether each PU adopts the average of two reference blocks as final prediction. Let D_{PMR} denote the distortion produced by PMR, and D_{no_PMR} denote the distortion produced by the original MC method in HEVC. The switch is set to be on only when $D_{PMR} < D_{no\_PMR}$. However, as the added switch brings a burden to bit streams at the same time, an elaborate scheme is needed to put switches to PUs which really worth the overhead.

Table 1 shows the use ratio of PMR in different PU modes. For PUs with uni-directional reference blocks, the use ratio of PMR in skip mode and merge mode are 63% and 64% respectively, while only 44% PUs in AMVP mode choose PMR. Since AMVP PUs always correspond to the areas with complex texture and motion, resulting in relatively large residuals, optimizing the prediction for AMVP PUs will greatly affect the coding performance. Accordingly, the switches for AMVP PUs are necessary, and as the amount of AMVP PUs is least among the above three modes, the overhead bits will have little influence on the increase of bit stream. For skip and merge PUs, since the ratio of PMR is high, additional bits to indicate whether to conduct PMR or not might be useless. Based on the above analysis, we further propose a mode-dependent pixel-wise motion refinement flag adding scheme (MPMR) that the switch is only set to the PUs with AMVP mode, and for the PUs with skip and merge modes, PMR is used as default.



(a) PMR is better (9622 pixels)



(b) PMR is worse (7610 pixels)

Fig. 4 Analysis of PMR referring to all pixels in a frame

Table 1: The use ratio of PMR for PUs with uni-directional reference blocks.

HEVC sequences	Skip	Merge	AMVP
Traffic_2560x1600	60%	60%	44%
PeopleOnStreet_2560x1600	64%	61%	35%
Kimono_1920x1080	58%	49%	29%
ParkScene_1920x1080	57%	56%	39%
Cactus_1920x1080	63%	60%	41%
BasketballDrive_1920x1080	56%	52%	34%
BQTerrace_1920x1080	48%	42%	33%
BasketballDrill_832x480	75%	71%	50%
BQMall_832x480	70%	71%	44%
PartyScene_832x480	69%	73%	61%
RaceHorses_832x480	67%	63%	39%
BasketballPass_416x240	73%	71%	42%
BQSquare_416x240	73%	80%	70%
BlowingBubbles_416x240	64%	69%	53%
RaceHorses_416x240	68%	65%	46%
Average	**64%**	**63%**	**44%**

4. EXPERIMENTAL RESULTS

The coding performance of the proposed methods are presented in this section. HM12.0 is set to be anchor and the proposed methods are implemented based on it. All the experiments are conducted under common test conditions [11] except that the number of encoding frames is set to two times the frame rate for time saving concerns.

Original BIO only applies to PUs with bi-directional reference blocks under RA configuration, which does not work in other configurations [7]. In this paper, the proposed PMR and MPMR break this restriction and can be applied to low delay B (LDB) configuration. Table 2 shows the coding performance of proposed method in LDB case, wherein A represents the performance that PMR only applies to the predictions located in different frames, while B represents the performance that PMR only applies to the predictions located in the same frame. 0.69% and 0.20% luma bitrate reductions are achieved, respectively, which validates the effectiveness of PMR in these two cases. When applying PMR to PUs with two forward predictions regardless of their locations, i.e., the combination of A and B, 0.34% average coding gain is achieved with 0.88% loss in class B. However, this unstableness can be well handled by MPMR, which obtains on average 0.83% coding gain with 0.05% bitrate reduction in class B. It is also noticed that MPMR achieves better coding performance on sequences with complex motion, such as camera rotation and zoom in/out cases, i.e. MPMR achieves up to 4.92% coding gain on *Spincalendar_1280x720.* Table 3 further shows the coding performance on sequences with affine motion in LDB configuration with average 2.20% bitrate reduction achieved by MPMR.

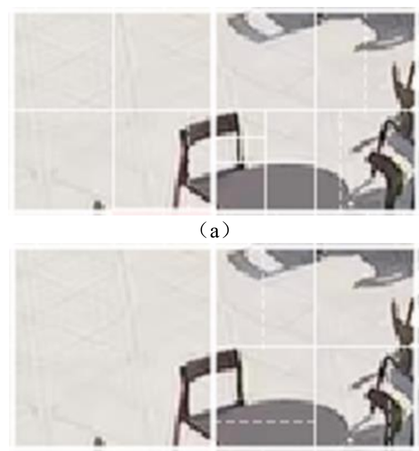
Table 2: Coding performance of proposed methods on HEVC sequences in low delay B (LDB) configuration

HEVC sequences	A	B	PMR	MPMR
Class A	-0.49%	0.04%	-0.09%	-0.61%
Class B	-0.27%	0.38%	0.88%	-0.05%
Class C	-0.93%	-0.17%	-0.74%	-1.25%
Class D	-1.07%	-0.48%	-1.10%	-1.52%
Class E	-1.12%	-0.98%	-0.98%	-1.53%
Average	**-0.69%**	**-0.20%**	**-0.34%**	**-0.83%**

Note that the efficiency of BIO benefits from the increased ratio of large size PU in frame partition [7]. Similar tendency is also observed when the proposed PMR is applied. Concretely, in our experiments under LDB configuration, Fig. 5(a) shows the CU and PU partitions without PMR, while Fig. 5(b) shows the CU and PU partitions with PMR. Thick solid lines indicate the boundary of CTU, and thin solid lines indicate the boundary of CU and dashed lines show the boundary of PU. One can notice that in Fig. 5(a), 2 CTUs are split into 14 CUs, whereas in Fig. 5(b), by adopting PMR, the same CTUs are only split into 5 CUs. Obviously, these results demonstrate that CU and PU tend to have larger size when PMR is applied, which means the fine motion inside PUs can be well handled by PMR.

Table 3: Coding performance of the proposed methods on sequences with affine motion in LDB configuration

Deformation sequences	A	B	PMR	MPMR
BlueSky_1920x1080	-0.34%	0.21%	0.52%	-0.64%
CactusPart_1920x1080	-0.27%	-0.13%	-0.03%	-0.75%
Station_1920x1080	-0.50%	-2.63%	-2.60%	-2.99%
Tractor_1920x1080	-0.17%	-0.29%	-0.01%	-1.17%
Spincalendar_1280x720	-3.63%	-2.98%	-4.95%	-4.92%
Jets_1280x720	-1.37%	-2.09%	-2.73%	-2.76%
Average	**-1.05%**	**-1.32%**	**-1.63%**	**-2.20%**



(a)

(b)

Fig. 5 Impact of PMR to CU and PU partitions: *BQSquare_416x240* sequence, POC=18, Qp=27.

5. CONCLUSION

This paper presents a pixel-wise motion refinement method (PMR) for PUs with uni-directional reference blocks, and a PU level mode-dependent scheme (MPMR) is presented. Experimental results demonstrate that the proposed method achieves 0.83% bitrate reduction on average for HEVC test sequences under HM12.0 low delay B (LDB) configuration, and achieves up to 4.92% bitrate reduction for sequences with complex motion, e.g. affine motion.

ACKNOWLEDGMENT

This work was supported in part by the National Basic Research Program of China (2015CB351800), National Natural Science Foundation of China (No. 61571017, No. 61421062), and Shenzhen Peacock Plan, which are gratefully acknowledged.

6. REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, pp. 1648-1667, 2012.

[2] High Efficiency Video Coding, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, 2013.

[3] J. Ma, J. An, K. Zhang, S. Ma, S. Lei, "Progressive Motion Vector Resolution for HEVC," in *VCIP*, 2013, pp.1-6.

[4] Z. Wang, J. Zhang, N. Zhang, S. Ma, "Adaptive Motion Vector Resolution Scheme for Enhanced Video Coding," in *DCC*, 2016.

[5] A. Zheng, Y. Yuan, H. Zhang, H. Yang, P. Wan, Au, O. C., "Motion vector fields based video coding," in *ICIP*, 2015, pp.2095-2099.

[6] Stanley H. Chan, Dung T. Võ and Truong Q. Nguyen, "Subpixel motion estimation without interpolation," in *ICASSP*, 2010, pp.722-725.

[7] A. Alshin, E. Alshina, T. Lee, "Bi-directional optical flow for improving motion compensation," in *PCS*, 2010, pp.422-425.

[8] B. D. Lucas, T. Kanade et al., "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, vol. 81, pp. 674- 679.

[9] A. Alshin, E. Alshina, M. Budagavi, K. Choi, J. Min, M. Mishourovsky, Y. Piao and Ankur Saxena, "Coding efficiency improvements beyond HEVC with known tools," in *Proc.SPIE* 9599, Applications of Digital Image Processing XXXVIII, 95991C, 2015.

[10] VCEG:'VCEG HM KTA-2.0', https://vceg.hhi.fraunhofer.de/svn/svn_HMKTASoftware/tags/HM-14.0-KTA-2.0/

[11] F. Bossen, "Common HM test conditions and software reference configurations," ISO/IEC JTC1/SC29/WG11, JCTVC-G1200, Geneva, CH, 2011.