MOTION VECTOR DERIVATION OF DEFORMABLE BLOCK

Na Zhang, Xiaopeng Fan, Debin Zhao, Wen Gao, Fellow IEEE

Dept. of Computer Science and technology, Harbin Institute of Technology, Harbin, China
hitnzhang@hit.edu.cn, fxp@hit.edu.cn, dbzhao@hit.edu.cn, wgao@jdl.ac.cn

ABSTRACT

Motion estimation (ME) plays an important role in most video encoding systems since it could significantly affect coding performance. However, both the next generation video coding standard High Efficiency Video Coding (HEVC) and the current video coding standard H.264/MPEG-4 AVC employ block matching motion estimation (BMME) which is based on translation motion model. This makes it difficult to represent the complex motion accurately such as rotation, zoom, and etc. In this paper, we propose an adjacent-block-based prediction model to improve the prediction performance of a deformable block. Based on this model, the motion information of each 4x4 block, i.e. the minimum partition (MP) in a coding unit (CU) of HEVC, is derived from the motion information of the nearest neighbors to the four corners of current prediction unit (PU). We integrate our method into HEVC as an additional choice of its merge mode. Simulation results show that our proposed method has better performance compared to HM4.0, the BD bit rate saving is up to 15.4%, while the encoding and decoding complexities are almost the same.

Index Term — video coding, adjacent-block-based prediction model, motion estimation, motion prediction, deformation motion

1. INTRODUCTION

BMME is crucial to most video coding standards, such as MPEG-1/2/4, ITU-T H.261/263/264 and HEVC. By using motion estimation and compensation, we are able to reduce temporal redundancy that exists between frames, which contribute to high compression efficiency. However, BMME assumes that all the pixels in a block undergo the same motion, resulting in poor prediction for the blocks with deformation motion.

To overcome the defect of the block matching algorithm (BMA), Lee and Wang [1] suggest using nodal-based deformation model which assumes that a selected number of control nodes in a block can move freely, and the displacement of any interior point can be interpolated from nodal displacements. This method is sensitive to the initial value, and requires a great amount of computation. Wang and Ostermann [2] have pointed out that the mesh-based model can give better prediction than the block-based model when existing non-translation motion, such as head rotation and turning. However, when a block includes multiple objects moving in different directions, these two motion models will have a bad prediction performance. An affine motion field prediction based on translational motion vectors (MVs) is proposed in [3] for better modeling complex motion, which is used as a post-processing step after mode decision and ME. A parametric skip mode based on higher-order parametric motion model is presented in [4] for better prediction of complex motion.

The emerging HEVC standard provides significantly better coding efficiency compared to H.264/AVC especially for high-definition (HD) resolution video sequences [5]. However, it still employs BMME and disables 4x4 inter partitions in HM4.0 [6]. Therefore, in order to further improve the coding efficiency of HEVC, it is desirable to handle the block containing complex motion.

In this paper, to model complex motion more effectively, we propose an adjacent-block-based prediction model. Based on this model, the motion information of each MP in the current PU can be interpolated from the motion information of the adjacent MPs to the four corners of current PU. In this way, we can give a more accurate motion prediction for the deformable block.

In the proposed model, we approximate the motion of each MP in a deformable block as translational motion, but we use the translational motion of MPs in a deformable block to approximate the deformation motion of the deformable block. Our method does not code motion parameters as the previous algorithms did for processing deformable block. We acquire motion information of a deformable block by the motion information of its neighbors.

The rest of the paper is organized as follows. Section 2 gives a brief overview of merge mode in HEVC. Section 3 describes the details of our proposed method. In Section 4, our experimental results are presented. Finally, Section 5 concludes this paper.

2. MERGE MODE IN HEVC

2.1. Overview of Coding Structure in HEVC

In HEVC, there are three basic units: coding unit (CU), prediction unit (PU), transformation unit (TU). Coding unit (CU) is a basic unit with a square shape. It has a similar role to the macroblock in H.264/AVC. CU size is limited to values which are a power of 2 and are greater than or equal to 8. The CU structure within the largest coding unit (LCU) can be expressed in quadtree adapted to the picture. PU as a basic unit for prediction is defined only for CU which is not further splitted. TU is another basic unit for transformation and quantization.

2.2. Background of Merge Mode

As we know, we need to transmit motion parameters for each inter predicted block separately. To improve coding efficiency, the block merging process allows to merge several blocks with similar movement into a single region. In this case, we only need to transmit motion parameters for the entire region once, rather than transmitting motion parameters for each block of the area respectively [7]. Owing to the feature, merge mode contributed significantly to the effectiveness of HEVC. In [8], the skip mode uses the merge method instead of the advanced motion vector prediction (AMVP) method to derive motion information, which further increases the utilization of merge mode.

2.3. Merge Mode in HM4.0

Merge mode was gradually improved with the update of the version of reference software of HEVC. In this section, we will briefly describe the merge mode in HM 4.0.

Merge mode first constructs the merging candidate list and then selects motion information (such as MV, reference index) of the best candidate according to rate-distortion optimization (RDO) criterion as that of the current block. The merging candidate list is constructed of the elements as shown in Fig. 1(a) which is given as specified order: A_1, B_1, B_0, A_0, B_2, Col.

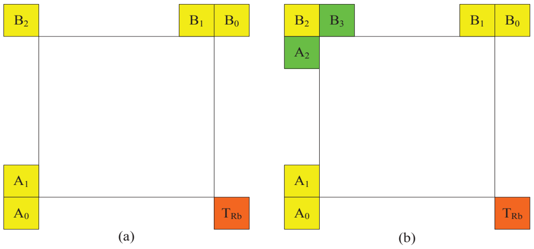

Fig. 1. Candidate positions: (a) for merge mode in HM 4.0, (b) for control MPs of our proposed method

3. PROPOSED METHOD

Our method interpolates motion information of each MP from the motion information of the adjacent MPs to the four corners of current PU by bilinear interpolation. In the following, we will introduce our method in detail.

3.1. Derivation of Control MP

From Fig. 1(b) we can see that there is at least one candidate position (such as B_2, B_3, A_2 for above left corner) around each of the four corners of current PU. We employ one control MP for each corner. We obtain the control MP separately for each reference list according to the following priority order:

(1) For above right corner, we first check position B_1, if it is unavailable or its prediction mode is intra, we then check B_0; Otherwise, B_1 is used as control MP.
(2) For left bottom corner, the checking priority is A_1, A_0.
(3) For above left corner, the checking priority is A_2, B_3, B_2.

Only when the motion information of control MP around each corner can be derived and they are not identical in at least one reference list, can we interpolate the motion information of each MP in the current PU.

3.2. Derivation of Interpolation Kernel

Let the number of control MPs be fixed as 4, corresponding to the four corners. The MV of the *k*th control MP of B_m (block with index of m) is specified by $d_m(MP_k)$ and the MV of the target MP is $d_m(MP_x)$. Then the motion function of the block is described by

$$d_m(MP_x)=\sum_{k=1}^{4}\phi_{m,k}(x)d_m(MP_k), MP_x \subseteq B_m \quad (1)$$

The interpolation kernel $\varphi_{m,k}(x)$ depends on the contribution of the *k*th control MP of B_m to MP_x. The interpolation function here we use is bilinear function. We set different interpolation kernels according to the selected control MPs. Similarly, the interpolation kernels for blocks of different sizes can also be derived. Fig. 2 depicts four different cases of the selected control MPs when B_2 is selected as the control MP for above left corner. Formula (2) gives the interpolation kernels corresponding to Fig. 2 (d).

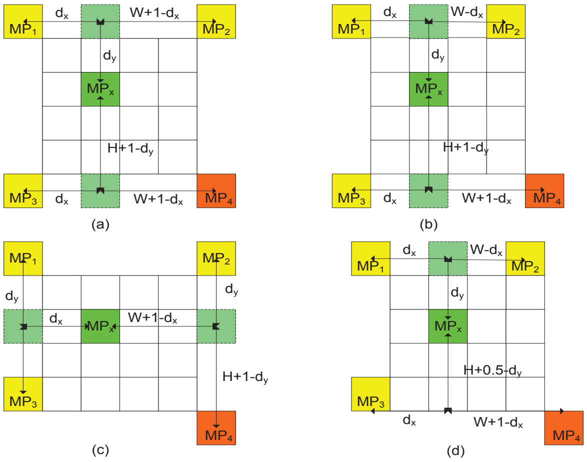

Fig. 2. Four different cases of the selected control MPs when B_2 is selected as the control MP for above left corner

$$\begin{aligned}
\phi_{m,1}(x) &= (W - dx)\cdot(H + 0.5 - dy)/(W\cdot(H+0.5)) \\
\phi_{m,2}(x) &= dx\cdot(H + 0.5 - dy)/(W\cdot(H+0.5)) \\
\phi_{m,3}(x) &= (W + 1 - dx)\cdot dy/((W+1)\cdot(H+0.5)) \\
\phi_{m,4}(x) &= dx\cdot dy/((W+1)\cdot(H+0.5))
\end{aligned} \tag{2}$$

In formula (2) and Fig. 2, W denotes the width of the block. H denotes the height of the block. And dx represents the distance between the center of target MP and the center of left control MP (MP_1 or MP_3) in the horizontal direction, dy represents the distance between the center of target MP and the center of above control MP (MP_1 or MP_2) in the vertical direction. All the distances denote the actual values divided by 4. As shown in Fig. 2, a 16x16 block, 2Nx2N PU splitting mode, W is 4.

3.3. Derivation of Reference Index for Target MP

After determining the weighting factor of each control MP, we can derive reference index $refIdxLXMP_x$ (with X being replaced by 0 or 1) of the target MP. Reference index of the control MP_k (k=1,2,3,4) in list X (with X being replaced by 0 or 1) is $refIdxLXMP_k$.

- If all of the four control MPs have the same reference indices, $refIdxLXMP_1$ is assigned to $refIdxLXMP_x$.
- Otherwise, the weights of control MPs whose reference indices are the same are added together.
  - If the number of reference indices with maximum weight is only one, $refIdxLXMP_x$ is equal to the corresponding reference index .
  - Otherwise, the minimum reference index with maximum weight is assigned to $refIdxLXMP_x$.

3.4. Derivation of Motion Vector for Target MP

Before using the MVs of control MPs to interpolate the displacement of target MP, if the MV of a control MP (such as MP_i) pointing to a different reference picture, the MV is scaled to the target reference picture $refIdxLXMP_x$ as the final MV. The MV of MP_i scaling process similar to derivation process for temporal MV prediction candidate [6] is derived as follows:

$$\begin{aligned}
&td = Clip3(-128, 127, PicOrderCnt(currPic) - \\
&\qquad refPicOderCnt(currPic, refIdxLXMP_i, LX)) \\
&tb = Clip3(-128, 127, PicOrderCnt(currPic) - \\
&\qquad refPicOderCnt(currPic, refIdxLXMP_x, LX)) \\
&tx = (16384 + Abs(td/2))/td \\
&ScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6) \\
&mvLXMP_i = ClipMv(Sign(ScaleFactor * mvLXMP_i) * \\
&\qquad ((Abs(ScaleFactor * mvLXMP_i) + 127) >> 8))
\end{aligned}$$

The above functions are standard functions in HEVC.

After the MV scaling process, we can utilize the MVs of control MPs to interpolate the MV of any MP in the current block using formula (1) according to the interpolation kernels derived from Section 3.2. We provide the interpolated MV of every MP in a rotary 16x16 block and the scaled MV of each control MP as an example in Fig. 3. All MVs are given in quarter-luma-sample unit.

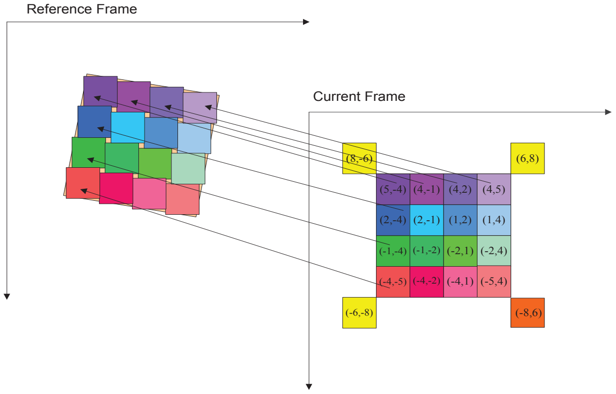

Fig. 3. MV of each MP in a rotary 16x16 block and the scaled MV of each control MP

We can observe that our proposed method provides a more accurate description for the movement of a rotary block compared to conventional ME for which there is only one MV for a block. Similarly, our method can predict more accurately for scaling and other deformation movements.

To ensure the robustness of our scheme, RDO is used to decide whether to adopt the proposed method or not. If our method is adopted, we don't need to transmit merge index to decoder compared to original merge mode, but we need to send an additional flag to inform decoder that encoder uses our proposed model. Otherwise, we need to send merge index and transfer an additional flag to decoder.

4.EXPERIMENTAL RESULTS

Our proposed method is implemented in HM4.0 and simulated under the common test conditions [9]. Taking into account the effectiveness of implementation, we only apply the adjacent-block-based prediction model to the block using merge mode with CU size larger than or equal to 16x16.

Table I provides the simulation results of the following sequences: Cactus (300 frames, involve local rotation), BlueSky (200 frames, camera rotation, many details), Station (100 frames, long zoom out, many details), Tractor (200 frames, from the local details to the whole scene), Shields(100 frames, significant zoom, many details), Vidyo3 (200 frames, involve head rotation and hand movement), Flowervase (416x240, 200 frames, from the whole scene to the local details); BasketballDrive(300 frames), BQMall (300 frames), RaceHorses (416x240, 300 rames) are video sequences with common motion. Entropy coding mode in Table I is CABAC, due to that CAVLC has been removed [10].

Table I. Coding performance comparison between the proposed scheme and HM 4.0

Sequence	BD Bit Rate(%)								
	RA HE			LB HE			LP HE		
	Y	U	V	Y	U	V	Y	U	V
Cactus_1920x1080	-1.8	-1.6	-1.7	-1.4	-1.2	-1.1	-1.2	-1.5	-1.3
BlueSky_1920x1080	-1.7	-1.1	-1.1	-2.9	-2.2	-2.1	-2.1	-1.8	-1.8
Station_1920x1080	**-7.2**	**-5.8**	**-5.9**	**-15.4**	**-10.8**	**-10.9**	**-15.0**	**-12.6**	**-11.9**
Tractor_1920x1080	-3.7	-2.2	-2.2	-8.5	-6.1	-5.8	-8.1	-6.4	-6.2
Vidyo3_1280x720	-1.0	-0.6	-0.7	-2.5	-2.4	-1.4	-2.0	-2.2	-1.9
Shields_1280x720	-2.7	-1.8	-2.3	-6.7	-5.3	-4.4	-5.0	-3.9	-3.8
Flowervase_416x240	-1.2	-1.0	-1.2	-1.7	-1.9	-1.7	-0.3	1.1	0.0
BasketballDrive 1920x1080	-0.2	-0.3	-0.2	-0.1	-0.2	0.0	-0.2	0.0	-0.4
BQMall_832x480	-0.1	-0.3	-0.4	-0.1	0.1	0.3	-0.1	-0.9	-0.1
RaceHorses_416x240	-0.1	-0.2	-0.4	-0.1	0.1	0.0	-0.3	-1.0	-0.7
Enc Time[%]	102%			102%			102%		
Dec Time[%]	104%			105%			104%		

From Table I we can see that our proposed method has particularly good performance for the first seven sequences which involve deformation motion. Especially for *Station*, the proposed scheme provides 7.2%, 15.4%, 15.0% BD bit rate saving corresponding to RA HE, LB HE, LP HE. The proposed method also has good performance for the remaining six sequences. Even for sequences with common motion such as the last three sequences in Table I, our method has no performance loss. We can also observe that our method has almost the same time cost as HM4.0.

Table II taking *Station* as an example presents the utilization rate of merge mode and our proposed model in HM 4.0 and the proposed method. In Table II, PM denotes the proportion of the MPs using merge mode; PO represents the percentage of the MPs using the proposed model; ∆PM denotes the variation of PM. We can see that our proposed method increases the utilization of merge mode, which makes our solution has the potential to reduce the bits sent to the decoder. The average utilization of our model for Station is 41.2%. PO can achieve 53.5% for Station in RA HE when the value of QP is 27.

Table II. Utilization rate of merge mode and our proposed model for Station in HM 4.0 and the proposed method

			HM_4.0rc1	Proposed		Compare
Sequence	cfg	QP	PM(%)	PO(%)	PM(%)	ΔPM(%)
Station_1920x1080	LB HE	22	80.3	33.1	86.5	6.2
		27	79.5	40.9	87.7	8.2
		32	82.8	42.5	88.0	5.2
		37	87.1	43.5	90.4	3.3
	LP HE	22	67.8	31.3	81.0	13.2
		27	64.3	42.4	83.4	**19.1**
		32	70.0	45.5	85.9	15.9
		37	82.3	45.1	88.8	6.5
	RA HE	22	64.0	42.9	72.7	8.7
		27	70.4	**53.5**	79.7	9.3
		32	80.6	45.4	85.6	5.0
		37	86.7	28.4	89.5	2.8
	Overall		76.3	41.2	84.9	8.6

5. CONCLUSION

In this paper, a new adjacent-block-based prediction model for more accurately predicting the deformable block is proposed. We integrate our method into HM4.0 as an additional choice of its merge mode. Different from HM4.0, we simultaneously utilize the motion information of multiple control MPs when deriving the motion information in merge mode. The experimental results obtained by the proposed algorithm show that our proposed method has significantly better coding performance than HM4.0 for sequences with complex motion and has no performance loss for the rest of sequences with common motion, which verifies that our proposed model is able to deal with deformable block effectively.

ACKNOWLEDGMENT

This work was supported in part by the Major State Basic Research Development Program of China (973 Program 2009CB320905), the National Science Foundation of China (NSFC) under grants 61100095 and the Program for New Century Excellent Talents in University (NCET) of China under grants NCET-11-0797.

REFERENCES

[1] Ouseb Lee, Yao Wang, “Motion-Compensated Prediction Using Nodal-Based Deformable Block Matching,” Journal of Visual Communication and Image Representation, vol. 6, no.1, pp. 26-34, Mar. 1995.

[2] Yao Wang, J. Ostermann, “Comparison of Block-Based and Mesh-Based Motion Estimation Algorithms,” IEEE International Symposium on Circuits and Systems, vol. 2, pp. 1157-1160, Jun. 1997.

[3] Roman C. Kordasiewicz, Michael D. Gallant, and Shahram Shirani, “Affine Motion Prediction Based on Translational Motion Vectors,” IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 10, pp. 1388-1394, Oct. 2007.

[4] A. Glantz, M. Tok, A. Krutz, and T. Sikora, “A block-adaptive skip mode for inter prediction based on parametric motion models,” IEEE International Conference on Image Processing, pp.1201-1204 , Sept. 2011.

[5] K. Ugur, K. Andersson, A. Fuldseth et al., “High performance, low complexity video coding and the emerging HEVC standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no.12, pp. 1688-1697, Dec. 2010.

[6] JCT-VC, “WD4: Working Draft 4 of High-Efficiency Video Coding,” JCTVC-F803, Torino, Jul. 2011.

[7] M. Winken, S. Boße, B. Bross et al., “Description of video coding technology proposal by Fraunhofer HHI,” JCTVC-A116, Dresden, Apr. 2010.

[8] B. Bross, J. Jung, Y.-W. Huang et al., “BoG report of CE9: MV Coding and Skip/Merge operations,” JCTVC-E481, Geneva, Mar. 2011.

[9] Frank Bossen, “Common test conditions and software reference configurations”, JCTVC-F900, Torino, July, 2011.

[10] JCT-VC, “WD5: Working Draft 5 of High-Efficiency Video Coding,” JCTVC-G1103, Geneva, Nov. 2011.