Applied Mechanics and Materials Vols. 556-562 (2014) pp 4365-4371
© (2014) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMM.556-562.4365

An Efficient Hardware-oriented Algorithm of Spatial Motion Vector Prediction for AVS HD Video Encoder

Minghui Yang[1, 2], Xiaodong Xie[2]

[1]Peking University Shenzhen Graduate School, 518055 Shenzhen, P. R. China

[2]National Engineering Laboratory for Video Technology, Peking University

100871 Beijing, P. R. China

{mhyang, xdxie}@jdl.ac.cn

Keywords: spatial motion vector, hardware pipeline, AVS.

Abstract. Motion Vector Prediction (MVP) plays an important role in improving coding efficiency in HEVC, H.264/AVC and AVS video coding standard. MVP is implemented by exploiting redundancy of adjacent-block optimal coding information under the constraint that MVP must be performed in a serial way. The constraint prevents parallel processing and MB pipeline based on LevelC+. In multi-stage pipeline, to some extent, adjacent-block best mode-decision information can hardly be obtained. In this paper, we propose a new hardware-oriented method to improve the coding performance at a cost of few hardware resources. When adjacent block is not available, spatial motion vector prediction (SMVP) for integer motion estimation (IME) and fraction motion estimation (FME) will take the IME best mode information and FME best mode information of left block as best information to derive PMV (Predicted Motion Vector) for current macro-block or block. Experimental results shows that the method we propose can achieve a better performance than the existing methods by 0.1db for the cases with intense movement and a non-degrading performance for flat cases.

I. Introduction

AVS video standard is developed by the audio video coding standard working group of China, which was approved by the Chinese science and technology department of Ministry of Information Industry in June 2002 and has been accepted as an option by ITU-TFGIPTV for IPTV applications [1]. The AVS-P2 is one part of AVS-video, which targets to high-definition (HD) digital video broadcasting and high-density storage media and achieves comparable performance with H.264/AVC with lower cost [1]. MVP is an important tool in AVS like in HEVC and H.264 /AVC. In a popular multi-stage MB-pipelined hardware structure (Fig.1), it is exploited in several stages such as IME, FME, MD, BG. SMVP which need best mode decision (MD) information (mainly including reference index and motion vector) of the adjacent MBs (including left, top-left, top and top-right MBs), which impedes the coding pipeline. In a popular coding order called n-stitched zigzag scan [2], sometimes it doesn’t satisfy those data dependencies.

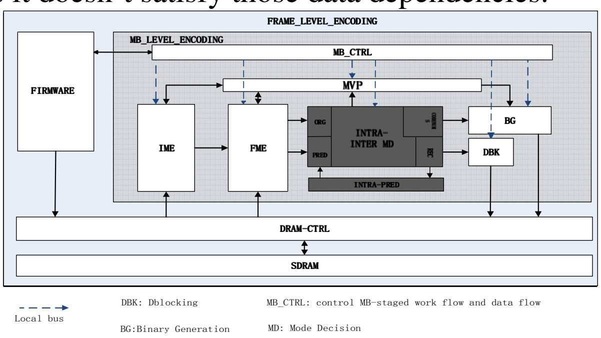

Fig. 1. AVS MB-staged pipeline engine.

To address this problem, works [2][3][4][5] proposed a similar method for H.264/AVC MVP in which when left MB's best information is not available, it will be replaced with top-left MB's information. To some extent, for flat sequences, this method indeed solves the data dependencies when PMV for integer motion estimation (IME) is implemented. However there still exits some flaws for which when top-left MB is INTRA, this method will be invalid because the real left MB may be not INTRA. Besides, for the sequences with intense local movement, this method will not work well.

For this, we propose efficient approximate SMVP algorithms based on friendly hardware, on condition that side-block information is not available. Two conditions here are studied. First, PMV for IME: when left MB is not available, it will be replaced with its best mode information transferred by ping pong buffers with the depth of three from IME. Second, PMV for FME: we use left-block best mode information from FME as left block optimal information.

The rest of the paper is organized as follows. In Section 2, first, we describe the standard AVS SMVP algorithm; second, we introduce zigzag HF5V3 coding order [4] and clarify the data dependencies of MVP. Section 3 proposes an efficient approximate SMVP algorithm. Afterward, in Section 4, we discuss the implementation results. Section 5 concludes this paper.

II. Standard AVS SMVP Algorithm and Data Dependencies of MVP-Based Zigzag HF5V3 Coding Order

A. SMVP Algorithm

SMVP exploits neighbor block information (such as MV and reference index) to predict current MB' s or block's (E's) motion vector (MV). The method of forming the motion vector predictor depends on the partition size and availability of adjacent MB's motion vectors. Fig.2 shows the location of neighbor MBs which are used to predict the MV of the current MB. Further details are described as follows.

According to AVS-p2 standard [6], MV of E can be calculated with A, B, C and D within four steps roughly described as follows.

First step:

Checking availability of the blocks (A,B,C and D) and whether they are INTRA mode , or whether their reference equals to E's.

Second step:

Checking whether the partition of E is 8x16 or 16x8.

Third step:

Scaling side-block MV to MVa, MVb and MVc.

Fourth step:

Using formula as follows, to derive the MV predictor of E that is a median of the MVs of partition A, B, and C.

$$PMVe = MEDIAN (MVa , MVb , MVc) \quad (1)$$

B. MVP pipeline scheduling

MVP plays a significant role in AVS-P2. Firstly, due to the high complexity of IME, MVP provides a start point for IME, which can reduce calculation works and can locate the desired search area as quickly as possible. Secondly, MVP also offers PMVs for FME and IME to calculate rate distortion (RD) cost by which the encoder pick out relatively better modes for next stage. Thirdly, Direct/Skip mode needs MVP to help to obtain reference pixels. Variable length coding (VLC) uses the result from MVP to get motion vector distortion (MVD).

Above all, we divide MVP into 4 parts by functions, namely MVP_IME ,MVP_FME , MVP_DIR and MVP_VLC. Because both MVP_IME and MVP_FME use SMVP, we will study mainly about SMVP in the rest of this paper. Fig.3 depicts a pipelined MVP Schedule as follows.

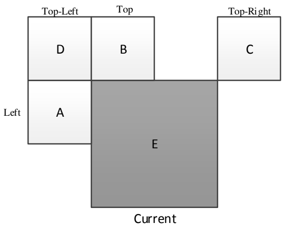

Fig. 2. Locations of current MB or block (E) and its neighboring blocks in motion vector prediction.

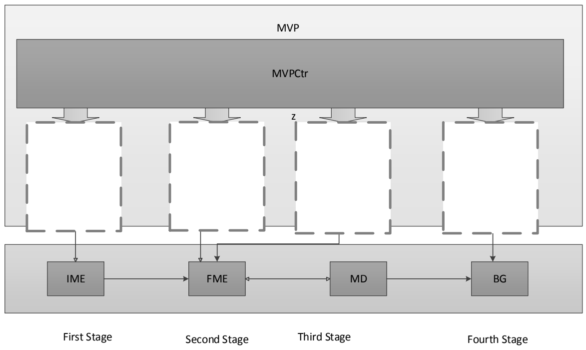

Fig. 3. Four stages MVP pipeline

MVP has a strong data dependencies with other functional blocks in a encoder, the hardware structure in Fig.3 cannot implement SMVP with exact adjacent-block information, which is processed in a zigzag HF5V3 [4] coding order.

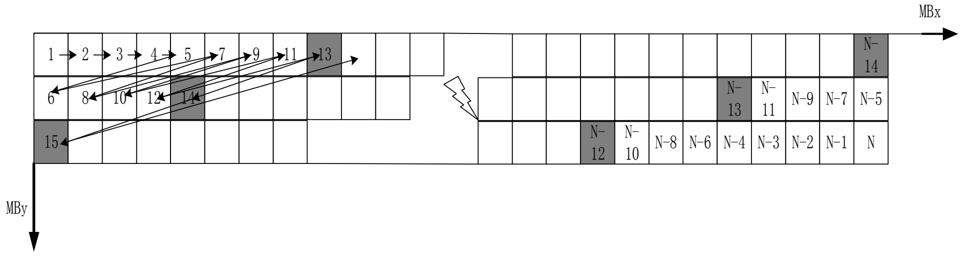

Fig. 4. Zigzag HF5V3 coding order

MBs are encoded in the order as demonstrated in Fig.4. Common zigzag scan begins setup from fifteenth MB and finishes by (N-12)th MB. From Fig.5, we can figure out that when MVP_FME module is implemented, top 15 MBs can't get left-MB's exact MVs. Additionally, the last 12 MBs in a zigzag unit (three rows of MBs as one zigzag unit) cannot get their left-block exact MVs either. However, MVP_IME module cannot get left block exact MV always.

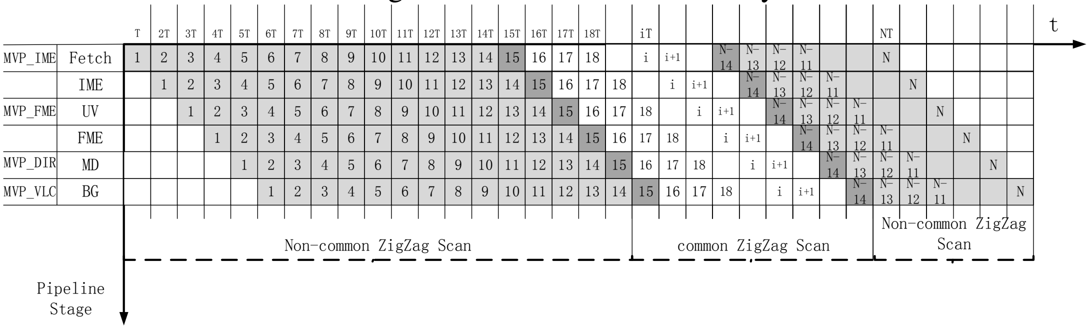

Fig.5. The implement of MB pipeline with zigzag HF5V3 coding order

III. The Proposed Algorithm

To solve the problems above more efficiently, we proposed an approximate SMVP algorithm that uses more ancillary information obtainable at a low extra cost.

A. Approximate Algorithm

For MVP_IME, we exploit left-block best mode information after IME, this method breaks the obstacles of data dependencies to make it possible that PMV for IME can be processed in pipeline at a comparably low price. The MV of A from IME is of integer pixel accuracy which is very close to the exact MV of the quarter-pixel accuracy with just an error of 1/4. In view of hardware-oriented structure, it is easy to fulfill this method with few resources required. In fact, we add two 3-depth FIFOs and a few condition decision. A more accurate start point is figured out by this method. As a result, IME can get more performance.

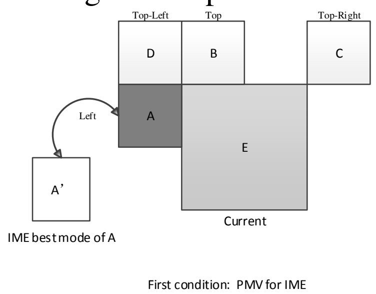

Fig. 5. A' is the best IME mode of A which is replaced with A'.

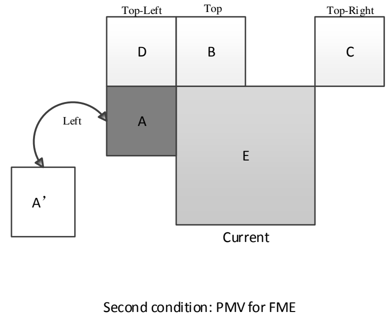

Fig. 6. Approximate SMVP for MVP_FME, A' is best mode information of A after FME

For MVP_FME, when left block of current MB is not available, it will be replaced with its best mode information after FME. Fig.6 shows an approximate SMVP for MVP_FME.

B. Algorithm Flow

Algorithm Flow is introduced as follows:

PMV for IME	PMV for FME
1) Check availability of A, B, C, D. a) If none of them is available. PMVe =0, Go to step 6) b) Else derive MV of A, B and C. Go to step 2). 2) Replace A with A' (best MV after IME). Go to step 4). 3) Scaling MV of A, B and C to MVa, MVb and MVc respectively. Go to step 5). 4) PMVe= MEDIAN(MVa, MVb and MVc). 5) Return PMVe.	1) Check availability of A, B, C, D. a) If none of them is available. PMVe =0, Go to step 6) b) Else derive MV of A, B and C. Go to step 2). 2) Common zigzag scan decision: a) YES: Jump to step 4) b) NO: Execute step 3) 3) Replace A with A' (best MV after FME). Go to step 4). 4) Scaling MV of A, B and C to MVa, MVb and MVc respectively. Go to step 5). 5) PMVe=MEDIAN(MVa, MVb and MVc). 6) Return PMVe.

IV. Implementation Results

In order to verify the method proposed, we map AVS-P2 reference software to a C-model with multi-stage pipelined structure. Two experiments are conducted on the same platform (C-model) with search range [-64,64]x[-64,-64] and two reference frames.

First, we converted the methods for multi-stage MVP [2][3][4][5] to our C-model for Experiment 1. Second, we embed our methods into C-model for Experiment 2. The same test sequences of three different sizes are applied to Experiment 1 and Experiment 2. On condition of four different QP (24,28,32,36), we obtained RD cures as Fig. 7 and Table 1.

Table 1 Comparison of the two experimental results

	experiment1				experiment2			
	IPPP		IPBB		IPPP		IPBB	
Squences	Bitrates	PSNR	Bitrates	PSNR	Bitrates	PSNR	Bitrates	PSNR
Basketball 720x576	3417.396	35.462	2777.607	35.155	3416.396	35.465	2773.3	35.155
	4882.764	37.55	4065.586	37.175	4884.258	37.553	4067.52	37.174
	6883.652	39.683	5923.184	39.314	6881.66	39.681	5933.965	39.316
	9638.802	41.891	8623.887	41.575	9640.986	41.892	8627.139	41.574
Flowergarden 720x576	2722.624	36.265	1813.477	35.885	2725.323	36.26	1815.431	35.886
	3996.563	38.446	2875.4	37.876	3999.316	38.444	2871.621	37.874
	5822.161	40.719	4651.553	40.238	5825.456	40.719	4650.762	40.24
	8312.448	43.05	7178.35	42.765	8313.41	43.05	7176.24	42.766
Mobilecalendar 720x576	4391.221	34.496	2813.701	34.296	4381.787	34.485	2806.586	34.291
	6434.238	36.664	4427.52	36.356	6437.373	36.668	4407.422	36.353
	9387.627	38.934	7006.377	38.63	9381.914	38.936	6985.518	38.63
	13484.502	41.309	10825.781	41.047	13473.721	41.311	10796.895	41.046
Night 1280x720p	5615.537	35.818	3996.65	35.639	5607.334	35.817	3940.078	35.631
	8550.586	37.779	6369.551	37.458	8534.317	37.781	6287.373	37.449
	13327.002	39.893	10556.602	39.46	13298.715	39.893	10448.498	39.45
	20578.66	42.213	17487.667	41.745	20540.156	42.214	17365.928	41.735
Sheriff 1280x720p	3384.727	36.411	2162.701	36.302	3377.402	36.413	2150.977	36.291
	5552.113	38.292	3703.33	37.944	5543.467	38.29	3697.777	37.941
	9099.977	40.326	6711.504	39.838	9083.115	40.324	6708.319	39.84
	14581.465	42.538	12122.109	42.075	14572.529	42.538	12117.539	42.075
Spincalendar 1280x720p	5740.781	34.532	3100.313	34.54	5727.305	34.536	3094.717	34.542
	9041.631	36.051	5264.616	35.905	9042.875	36.058	5219.502	35.903
	16281.383	37.919	11199.346	37.629	16276.816	37.92	11188.477	37.63
	28756.318	40.347	23448.604	40.084	28704.785	40.344	23431.729	40.085
BasketballDrive 1920x1080p	6034.629	37.552	5165.684	37.293	6049.072	37.555	5062.559	37.283
	9481.948	38.565	8011.069	38.295	9480.645	38.566	7857.92	38.284
	16911.299	39.58	14778.457	39.38	16916.25	39.579	14523.714	39.363
	37383.38	41.147	32829.111	40.893	37367.527	41.143	31934.257	40.851
CrowdRun 1920x1080p	24652.657	33.473	20613.962	33.098	24634.922	33.473	20603.262	33.099
	36901.586	35.38	32420.215	35.064	36891.914	35.379	32416.436	35.067
	56101.904	37.414	51412.91	37.193	56096.309	37.414	51387.422	37.193
	87931.992	39.87	83304.844	39.676	87946.941	39.871	83367.07	39.679
Sunflower 1920x1080p	4772.083	38.389	3197.9	38.185	4761.182	38.39	3070.189	38.181
	6860.947	40.104	4632.803	39.842	6856.143	40.102	4465.898	39.829
	9849.639	41.696	6883.477	41.498	9844.717	41.694	6666.504	41.492
	14325.645	43.179	10471.904	43.113	14336.531	43.177	10156.582	43.111

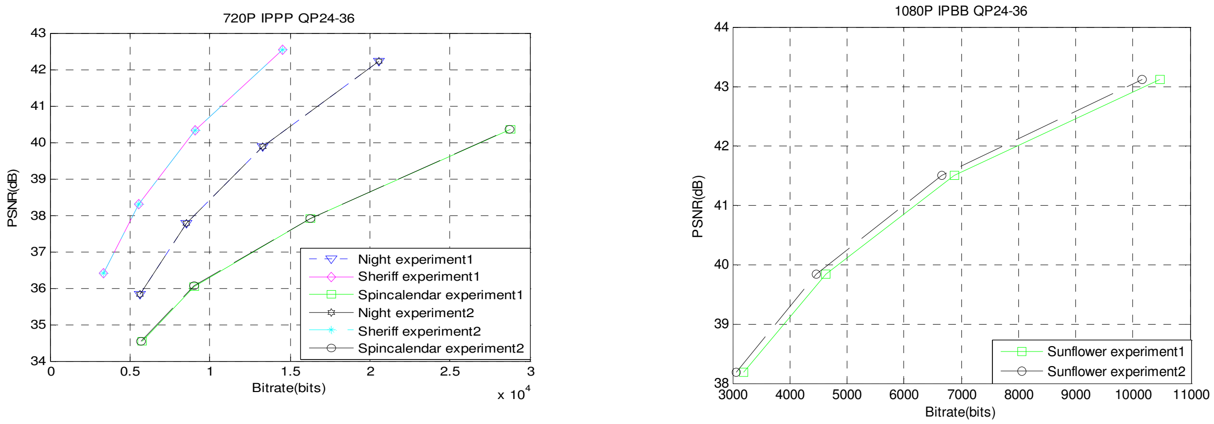

Fig. 7 The rate distortion curves of different standard sequences generated by Experiment 1 and Experiment 2, while Experiment 2 is for the method we proposed and Experiment 1 is for methods in [2][3][4][5].

From Fig.7, we can draw a conclusion that our proposed method achieves a little higher performance than the other method in experiment 1 by about 0.004, with a GOP structure of IPPP. However, this number increases to about 0.139 with picture size of 1080p and GOP of IBBP. Because Sunflower is a sequence of intense local movement, so it is not effective to replace left MB information with top-left MB information, it is likely that left MB and top-left MB moves in different direction with different MV in truth, this accumulative effects of error results in Experiment 2 gets higher PSNR than Experiment 1 with the same bitrates.

V. Conclusion

We proposed an efficient hardware-oriented SMVP algorithm. For most smooth sequences, our method gains a equal performance compared to the methods in literature. But for multi-direction motion sequences, the approach proposed in our paper can get a better result. In addition, the method proposed also can be well applied to alleviate the problem of MVP's data dependencies in H.264/AVC and HEVC hardware platform.

VI. Acknowledgement

This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No.61171139 and No. 61035001, and National High Technology Research and Development Program of China (863 Program) under contract No.2012AA011703.

References

[1] Wen Gao, et al. AVS Vide Coding Standard. Studies in Computational Intelligence, 2010, Volume 280, Intelligent Multimedia Communication: Techniques and Applications,Pages 125-166.

[2] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, and L.-G. Chen, "Level C+ data reuse scheme for motion estimation with corresponding coding orders," IEEE Trans. Circuits Syst. Video Technol., vol. 16, pp. 553–558, Apr.2006.

[3] Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture",Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on Volume 2, 23-26 May 2004 Page(s):II -273-6 Vol.2.

[4] Kaijin Wei, Rongwei Zhou,Shanghang Zhang,Huizhu Jia,Don Xie,Wen Gao,"An Optimized Hardware Video Encoder for AVS with LevelC+ Data Reuse Scheme for Motion Estimation", Multimedia and Expo (ICME) , 9-13 July 2012,pp. 1055 – 1060.

[5] Wei, Y., et al. Multi-stage motion vector prediction schedule strategy for AVS HD encoder. in Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on. 2010.

[6] GB/T20090.2 information technology—advanced audio video coding standard part2:Video ,2006