Proceedings of 2009 APSIPA Annual Summit and Conference, Sapporo, Japan, October 4-7, 2009

Low Memory Bandwidth Prediction Method for H.264/AVC Scalable Video Extension

Hsiao-Shan Huang, Gwo-Long Li, and Tian-Sheuan Chang
Institute of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan
E-mail: hshuang.ee96g@nctu.edu.tw, glli.ee95g@nctu.edu.tw, tschang@twins.ee.nctu.edu.tw

***Abstract*—Memory bandwidth issue becomes more and more critical in designing video coding system especially in scalable video coding due to its extra inter-layer prediction. This paper proposes a low memory bandwidth prediction method for inter and inter-layer residual prediction. The proposed method combines two predictions into one prediction process and reuses its data for lowering memory bandwidth requirements. The simulation results show that 67% of memory bandwidth in enhancement layer can be reduced with negligible rate distortion loss.**

Index Terms— scalable video coding (SVC), inter-layer prediction, motion estimation

I. INTRODUCTION

Recently, video coding has been developed rapidly to satisfy diverse applications range from mobile device display to high-definition TV. Traditional video coding standards optimize the video quality at a given bit-rate without considering the variety of transmission environment. An extension of H.264/AVC called scalable video coding (SVC) [1] addresses this heterogeneous environment problem.

SVC is a newest video coding standard which standardized by Joint Video Team (JVT). In SVC, it supports three scalabilities, including temporal, spatial and quality scalability. Temporal scalability supports video coding in different frame rate by using hierarchical B structure. Quality scalability is achieved by Fine-Grain Scalability (FGS), Coarse-Grain Scalability (CGS) or Medium-Grain Scalability (MGS). Spatial scalability is provided by varying frame resolutions.

Due to similarities between spatial layers, inter-layer prediction is adopted in SVC for reducing the redundancy existed between spatial layers. However, the inclusion of inter-layer prediction will additionally increase the memory bandwidth and computational requirements. Previous researches [2]-[4] have been proposed to reduce the computational complexity of SVC. However, they do not take the memory requirement issue into consideration. Since the memory access speed is far behind the data processing speed in modern video encoder design, the performance of designed video coding system is totally dominated by memory access. As a result, lowering the memory bandwidth requirements not only improves coding performance of video coding system but reduces the power consumption brought by memory read/write access. Therefore, a low-bandwidth prediction method which combines inter and inter-layer residual prediction into a single prediction module is proposed to deal with this problem.

This paper is organized as follows. We briefly describe the SVC prediction modes in Section II. In Section III, the proposed low bandwidth method is described in detail. Section IV shows the simulation results and the conclusion is given in Section V.

II. OVERVIEW OF PREDICTIONS MODES IN SVC

Except inherent coding modes in the H.264, the SVC additionally supports inter-layer prediction mode which uses reference layer information as the predictor to further reduce the redundancies existed between spatial layers. Fig.1 shows all supported prediction modes in SVC. In these prediction modes, the intra and inter prediction as shown in Fig.2 (a) are the same as the intra and inter prediction modes used in the H.264. For inter-layer prediction modes, the inter-layer intra prediction reuses the up-sampled texture from base layer (BL) as prediction reference for predicting the macroblock of enhancement layer (EL) when corresponding macroblock in BL is intra mode. For inter-layer residual prediction, the up-sampled BL residuals are subtracted from current data before performing motion estimation search. Fig.2 shows the concept of inter-layer residual prediction adopted in SVC. In contrast to inter prediction shown in Fig. 2(a), the operation of inter-layer residual prediction is the same as the inter prediction except the up-sampled BL residuals are subtracted from current pixels before feeding into integer motion estimation (IME) module. Inter-layer motion prediction adopts the up-sampled motion of BL as well as partition mode for prediction in enhancement layer. For selecting best mode, the final prediction mode is decided by *RDCost*. The function of *RDCost* is listed as follows:

$$J = D + \lambda \bullet R, \qquad (1)$$

where J represents *RDcost*, λ denotes Lagrangian parameter, D is distortion between current and reference data, and R refers to rate which is derived by computing the difference between selected MV and MVP.

The work is supported by National Science Council of Taiwan, under Grant NSC 97-2220-E-009-018.

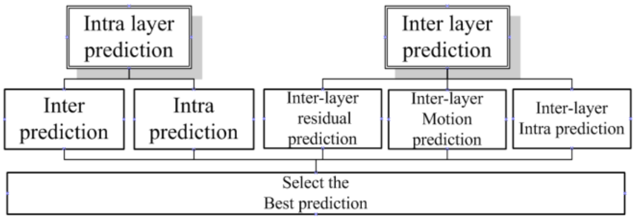
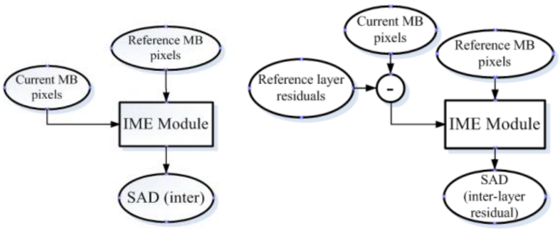


Fig.1 Prediction modes in SVC



(a) (b)

Fig.2 (a) Inter prediction process, (b) Inter-layer residual process in JSVM reference software

When computing the *RDCost* for inter mode prediction, the *D* term in equation (1) is derived by calculating the sum of absolute difference of each block (D_{Inter}) and it can be expressed as follows:

$$D_{Inter}(i,j) = \sum_{i=0}^{height} \sum_{j=0}^{width} |C(i,j) - F(i,j)|, \tag{2}$$

where ***C*** represents the pixels of current encoding block, ***F*** is the pixels of reference frame, *height* and *width* denote the current block size.

In contrast to inter prediction shown in Fig. 2(a), we can observe that the inter-layer residual prediction additionally substrates the up-sampled residual from current coding pixels before performing motion estimation search. Consequently, the *D* term in equation (1) can be calculated as follows when testing inter-layer residual prediction.

$$D_{ILres}(i,j) = \sum_{i=0}^{height} \sum_{j=0}^{width} |C(i,j) - B(i,j) - F(i,j)|, \tag{3}$$

where ***B*** represents the base layer up-sampled residuals.

From equations (2)-(3), it can be found that the main difference for calculating the *D* terms between inter and inter-layer prediction is that the inter-layer prediction additionally substrates the up-sampled residual from current coding pixels. Based on this observation, it is possible to reuse the current and reference data when testing both the inter- and inter-layer prediction modes.

III. PROPOSED LOW-BANDWIDTH PREDICTION METHOD

In this section, our proposed data reuse method for inter and inter-layer residual prediction is described in detail. Besides, a fast motion estimation algorithm is also adopted to further reduce memory requirement in motion estimation.

A. Proposed Data Reuse Method for Inter and Inter-layer Residual Prediction

From section 2, we observed some identical processes existed between inter and inter-layer residual prediction. Since the current macroblock data and the reference macroblock data of inter-layer residual prediction is the same as inter prediction, the difference between these two prediction modes is that inter-layer residual mode needs one more process which subtracts up-sampled base layer residual from current coding pixels before performing IME. Motivated by this observation, we can combine these two prediction modes into one motion estimation process and receive two prediction results at the same time. In other words, by changing the order of subtraction for D_{ILres} calculation, the equation (3) can be rewritten as follows.

$$D_{ILres}(i,j) = \sum_{i=0}^{height} \sum_{j=0}^{width} |C(i,j) - F(i,j) - B(i,j)| \tag{4}$$

Through the equation (4), the D_{Inter} can be derived by just extracting the results of *C-F* during the computation of equation (4). Finally, both the results of D_{Inter} and D_{ILres} can be obtained concurrently. The combined method is shown in Fig.3. When calculating the *RDCost* for inter prediction in IME module, we can perform inter-layer residual prediction at the same time by subtracting base layer residual. The advantage of our proposed method is that, in normal prediction process in the SVC, the current and reference pixels should be downloaded twice for both inter and inter-layer prediction. However, after the adoption of our proposal, the current and reference pixels only need to be loaded once for both prediction modes.

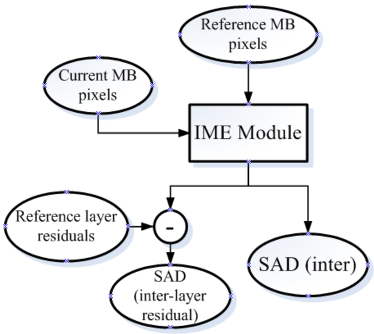


Fig.3 Proposed data reuse method which combines inter and inter-layer residual prediction

To demonstrate the efficiency of our proposed method, we show some numerical comparisons of bandwidth saving for our proposal and full search motion estimation with level C data reuse scheme [5]. The bandwidth requirements of full search motion estimation for loading reference data in EL per frame can be calculated as follows.

$$BW = [(SR \times 2 + 16)^2 \times MBl] + [(SR \times 2 + 1) \times 16 \times MBr] \quad (5)$$

where the *MBl* is the number of MBs located at left most column in a frame, the *MBr* is the number of MBs per frame except the *MBl*, and *SR* refers to the search range. For an example, the bandwidth requirement of full search motion estimation algorithm with level C data reuse scheme in encoding CIF image (search range = 16) can be calculated as follows.

$$\begin{aligned} BW &= [(16 \times 2 + 16)^2 \times 18] + [(16 \times 2 + 1) \times 16 \times 378] \\ &= 470.81\,Kbytes\ /\ Frame\,. \end{aligned}$$

However, the bandwidth requirement of our combined prediction method is reduced by a factor of two. As a result, only *235Kbytes/Frame* is required for our proposed method.

B. Adaptation of Fast ME Algorithm for Bandwidth Reduction

For reducing computational complexity of ME, many fast motion estimation algorithms have been proposed to decrease the ME complexity, such as TSS[6], 4SS[7] and DS[8]. But they are impractical to be realized by hardware, since data reuse is not effective and data access has no regular form. In this paper, a fast motion algorithm technique called small-cross search (SCS) motion estimation [9] is selected for our proposed low-bandwidth motion estimation method due to its regular memory access, data reuse and simplicity properties for hardware realization. Fig. 4 shows the concept of small-cross motion estimation algorithm. The operation of this algorithm is described as follows. First, the SADs of five positions labeled by 1 are calculated. If the minimum SAD is located at center position, the search operation is finished and the coordinate of center position is set as motion vector of current coding macroblock. Otherwise, the position with minimum SAD is set as the search center and another three additional positions labeled by 2 are checked. This operation is repeated until the position with minimum SAD is located at center or the search boundary is reached.

As described above that the small-cross search pattern can significantly reduce the computational complexity of motion estimation. However, this search pattern not only can achieve computational complexity saving, but it can gain the benefit of data reuse from the perspective of hardware realization. Fig.5 shows the phenomenon of data overlapping of small-cross search pattern. In this figure, the light circles indicate the overlapped area and the dark circles of top row, bottom row, left column, and right column are referred to the additional required pixels for SAD calculation in case of the position with minimum SAD is located at position 1, 2, 4, and 5, respectively. From this figure, we can notice that there is large part of data overlapping between any two adjacent search positions.

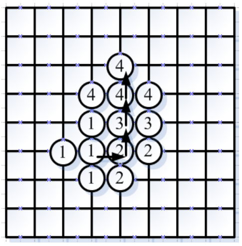


Fig.4 Example of small-cross search algorithm

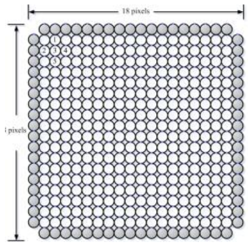


Fig.5 Phenomenon of data overlapping of small-cross pattern

Furthermore, the bandwidth requirements of small-cross pattern are analyzed as follows. In the first step, there are five positions with 18x18 pixels should be downloaded from external memory for evaluating the SADs. If the minimum SAD located at center position, there is no more pixels should be downloaded from external memory. Otherwise, if the position with minimum SAD located at any one of four corner positions, 18 addition pixels need to be downloaded from external memory for evaluation. From the above analysis, we can observe that the bandwidth requirements of small-cross pattern are proportional to the search steps. Therefore, the bandwidth requirements per frame of small-cross search pattern can be calculated as follows.

$$BW = [(18 \times 18) + (18 \times N)] \times MBs\,, \quad (6)$$

where *18x18* indicates the required pixels for computing the SADs for first five positions, *MBs* denotes the number of MBs in a frame and the *N* is the repeated steps for searching the best result.

However, the steps in SCS are not fixed since it depends on sequence content. This property results in the difficulty for hardware implementation. From the memory requirement analysis mentioned above, we observed that the SCS might result in higher memory requirements than level C data reuse scheme when the search steps reach a certain number. In order to find out the average steps of SCS, some simulations are performed. Fig.6 shows the simulation results of SCS method. The vertical and horizontal axes are the accumulated probability and search steps in SCS, respectively. Simulation shows that SCS method can find the best matching within 10 steps. Hence, we restricted SCS method to search 10 steps when realizing the SCS motion estimation algorithm. From equation (6), the bandwidth requirement can be fixed to,

$$BW = [(18 \times 18) + (18 \times 10)] \times 396 = 194.9\,Kbyte\ /\ Frame$$

in CIF size. Therefore, 17.0% memory bandwidth requirement can be reduced by SCS method when compared to level C data reuse method if the ±16 search range is adopted.

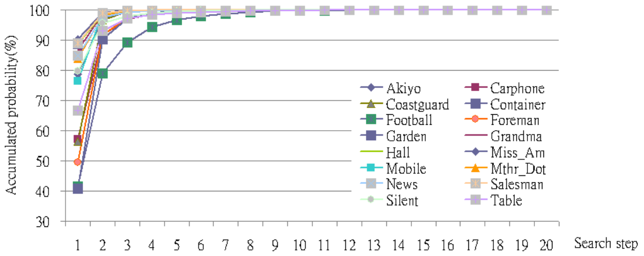


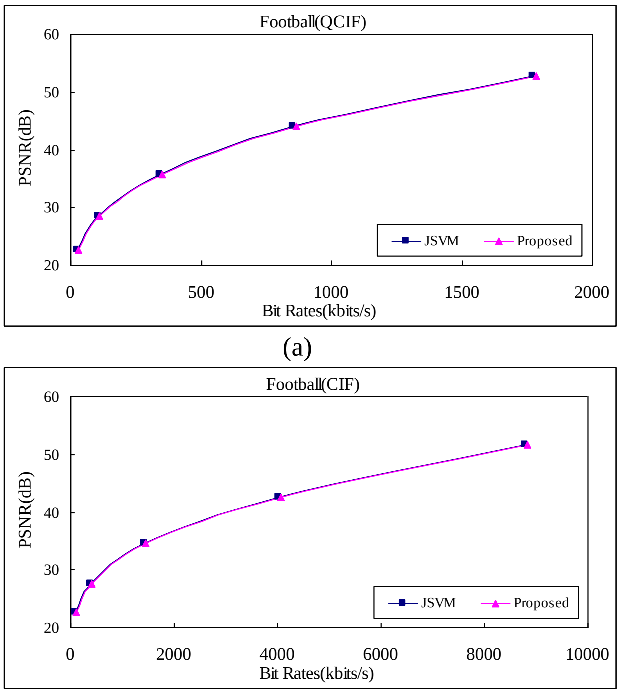
Fig.6 The steps probability of SCS method

IV. SIMULATION RESULTS

The proposed algorithm is implemented on a JSVM8.9 [10]. The test condition is shown in Table I.

Table I Simulation conditions

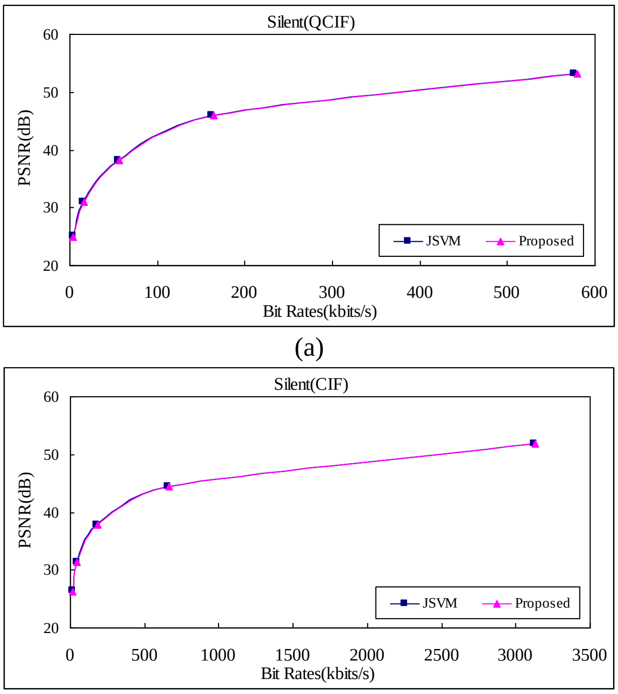
Codec	JSVM 8.9 encoder
Testing sequences	Football, Silent, Mobile, Akiyo, Container, Weather
QP	8, 18, 28, 38, 48
Resolution	QCIF and CIF
Frame Rate	15Hz
Encoder configuration	MV search range : ±16 pels. GOP size : 4. Reference frame number : 1 Adaptive selecting inter-layer prediction in enhancement layer



(a)

(b)

Fig.7 Rate-Distortion curve of Football. (a) QCIF (b) CIF



(a)

(b)

Fig.8 Rate-Distortion curve of Silent. (a) QCIF (b) CIF

The simulation results of our proposed method are compared with fast full search motion estimation algorithm in JSVM8.9. Fig.7 and Fig.8 show the rate-distortion performance comparisons for BL (in QCIF size) and EL (in CIF size). Two different motion behavior sequences, Silent and Football are shown in figures. From these figures, we can observe that the proposed method can achieve near the same rate distortion performance when compared to JSVM8.9. The detail comparisons for PSNR degradation and bitrate increase are shown in Table II. From this table, we observed that only 1.16% and 0.001dB bitrate increase and PSNR degradation in average for our proposed method, respectively. The rate distortion performance degradation mainly is caused by the limited steps of SCS algorithm. However, compared to the bandwidth saving which is critical issue in designing motion estimation, this rate distortion decrease is ignorable. Furthermore, the memory bandwidth requirements can be saved up to 67% for all sequences in average.

V. CONCLUSION

In this paper a low memory bandwidth prediction method is proposed to solve the problem of extreme memory bandwidth demands brought by inter-layer prediction of SVC. By combining both inter and inter-layer residual prediction, the reference data for prediction can be reused efficiently. Simulation results show that the proposed method can save 67% memory bandwidth requirement with slight rate distortion performance degradation when compared to JSVM8.9.

Table II PSNR degradation and bitrate increase comparisons

Sequence	**Resolution**	**QP:8**		**QP:18**		**QP:28**		**QP:38**		**QP:48**	
		B(%)	**PSNR**	**B(%)**	**PSNR**	**B(%)**	**PSNR**	**B(%)**	**PSNR**	**B(%)**	**PSNR**
Football	**QCIF**	0.485	0.033	1.118	0.022	2.208	-0.001	3.880	0.009	5.364	-0.123
	CIF	0.372	0.032	0.868	0.027	1.691	0.015	2.616	-0.003	3.108	-0.119
Silent	**QCIF**	0.240	0.003	1.052	-0.002	1.735	-0.018	1.878	-0.008	0.476	-0.084
	CIF	0.205	-0.001	0.753	-0.001	1.168	-0.004	1.377	-0.010	-0.113	-0.069
Mother	**QCIF**	0.351	-0.030	0.749	-0.010	2.013	-0.014	3.944	-0.048	1.313	-0.064
	CIF	0.105	0.003	0.293	-0.018	0.914	-0.005	2.837	-0.039	-0.549	-0.142
Mobile	**QCIF**	-0.058	0.020	-0.111	0.011	0.210	0.015	1.463	-0.015	0.492	-0.007
	CIF	0.046	0.003	0.103	-0.001	0.340	0.001	1.483	-0.010	1.025	-0.047
Akiyo	**QCIF**	-0.214	0.005	0.058	-0.015	1.258	0.044	1.090	0.007	-3.479	-0.041
	CIF	-0.096	-0.001	-0.053	-0.002	0.301	0.009	0.347	-0.046	-1.705	-0.011
Container	**QCIF**	0.061	-0.007	-0.122	-0.016	0.791	-0.034	0.276	-0.064	4.477	-0.128
	CIF	0.042	-0.002	-0.096	-0.006	0.575	-0.019	0.735	-0.034	0.276	-0.076
Weather	**QCIF**	0.221	-0.011	0.624	-0.004	1.857	0.002	3.654	0.011	4.187	-0.015
	CIF	0.271	0.002	0.665	-0.002	1.638	-0.024	3.149	-0.014	1.940	-0.022
Average	**QCIF**	0.198	0.006	0.607	0.001	1.865	-0.001	2.946	-0.021	2.596	-0.073
	CIF	0.168	0.006	0.472	0.000	1.259	-0.006	2.269	-0.026	1.376	-0.078

REFERENCES

[1] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz and M. Wien, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6: JVT-X201 'Joint Draft ITU-T Rec. H.264 | ISO/IEC 14496-10/Amd.3 Scalable video coding,' 24th Meeting, Geneva, Switzerland, Jun.29-Jul.5, 2007.

[2] H. Li, Z. Li and C. Wen, "Fast mode decision algorithm for inter-frame coding in fully scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp.889-895, July 2006.

[3] B. Lee, M. Kim, S. Hahm, C. Park, and K. Park, "A fast selection scheme in inter-layer prediction of H.264 scalable video coding," *in Proceeding of IEEE Symposium on Broadband Multimedia Systems and Broadcasting*, April 2008.

[4] H. Li, Z. G. Li, C. Wen, and L. P. Chau, "Fast mode decision for spatial scalable video coding," *in Proceeding of IEEE International Symposium on Circuits and Systems*, pp.3005-3008, May 2006.

[5] 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 Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp.553-558, April 2006.

[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *in Proceeding of IEEE National Telecommunication Conference*, pp. G5.3.1-G5.3.5, December 1981.

[7] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits Systems for Video Technology*, vol. 6, pp. 313-317, March 1996.

[8] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Transactions on Image Processing*, vol.9, no.2, pp.287-290, February 2000.

[9] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Transactions on Communications*, vol. 38, no.7, pp. 950-953, July 1990.

[10] J. Vieron, M. Wien and H. Schwarz, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6: JVT-U203 'JSVM 8 software' 21th Meeting, Hangzhou, CN, Oct. 2006.