MEMORY ANALYSIS OF VLSI ARCHITECTURE FOR 5/3 AND 1/3 MOTION-COMPENSATED TEMPORAL FILTERING

Chao-Tsung Huang, Ching-Yeh Chen, Yi-Hau Chen, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
Email:{cthuang, cychen, ttchen, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

To the best of authors' knowledge, this paper presents the first work on memory analysis of VLSI architectures for Motion-Compensated Temporal Filtering (MCTF). The open-loop MCTF prediction scheme has led the revolution for hybrid video coding methods that are mainly based on the close-loop MC Prediction (MCP) scheme, and it also becomes the core technology of the coming video coding standard, MPEG-21 part 13 - Scalable Video Coding (SVC). In this paper, the macroblock(MB)-level and frame-level data reuse schemes are analyzed for the MCTF. The MB-level data reuse is especially for the Motion Estimation (ME), and the Level C+ scheme is proposed, which can further reduce the memory bandwidth of the conventional Level C scheme. Frame-level data reuse schemes for MCTF are proposed according to the open-loop prediction nature.

1. INTRODUCTION

Existing hybrid video standards, such as MPEG-1/-2/-4 and the emerging H.264/AVC, mainly consist of a close-loop MCP scheme and a transform-based texture coder. The "close-loop" means it uses the reconstructed previous frames to predict the current frame, which forms a feed-back loop. The close-loop MCP scheme has been highly optimized for the compression efficiency in the last decade, and the H.264/AVC is a landmark of this development. However, for many video applications in the present and the future, the spatial, temporal, and Signal-to-Noise-Ratio (SNR) scalabilities become more and more demanded. Scalability means we can have multiple adaptations for one video bitstream, such as different frame sizes, frame rates, and visual qualities. The close-loop MCP scheme is hard to provide scalability while maintaining high compression efficiency because of the drift problem, like MPEG-4 FGS. The drift occurs when the encoder and decoder have different reconstructed frames, which exactly happens when the scalability should be provided. For overcoming the drift problem, the compression efficiency will be degraded very much and become unacceptable when there are many scalability layers.

The open-loop interframe wavelet coding scheme becomes a good alternative for scalable video coding, which concept is to perform wavelet transform in the temporal direction. But the coding performance is unacceptable without Motion Compensation (MC). In 1993, Ohm introduces a block-based displacement interframe scheme using the Haar filter [1]. However, the compression efficiency is still not comparable to existing MCP video standards until the lifting-based wavelet interframe scheme is proposed and the longer tap wavelet filters, like 5/3 filter, are used [2, 3]. For more details, please refer to [4].

MPEG has identified a set of applications that require scalable and reliable video coding technologies. After evaluating the response to Call for Proposals on Scalable Video Coding (SVC) [5], it has been shown that there is a new and innovative video technology that MPEG can bring to industry in a future video standard. In the two most significant proposals [6, 7] and many other proposals, the lifting-based MCTF is the core technology to provide scalable video coding. The MCTF not only can provide a variety of efficient scalabilities because the drift problem is prevented by the open-loop structure but also can increase the compression efficiency of H.264/AVC [7].

In this paper, we would like to present the first work on VLSI architecture of MCTF by analyzing the memory issues because MCTF is a breakthrough and the key component of the interframe wavelet video coding. This paper is organized as follows. In section 2, the MCTF schemes using the Haar, 1/3, and 5/3 filters, are introduced. In section 3, The MB-level data reuse schemes for ME are discussed, and a new Level C+ scheme is also proposed. The memory analysis of the 5/3 and 1/3 MCTF is presented in section 4, and some frame-level data reuse schemes are also proposed. This paper is concluded in section 5.

2. MOTION-COMPENSATED TEMPORAL FILTERING

In general, MCTF is a concept to perform wavelet transform in the temporal direction. The coding performance and coding delay depend on which wavelet filter is adopted. From recent experimental results [4], the MCTF is usually implemented by use of the 5/3, 1/3, or Haar filter with the lifting scheme. The lifting scheme is an efficient implementation method of wavelet filters, and it can guarantee the perfect reconstruction property. For simplicity, MCTF represents the lifting-based MCTF using the 5/3 or 1/3 filter in the following because the coding performance of the 5/3 filter is better than the Haar filter and the 1/3 filter is only a subset of the 5/3 filter.

The 5/3 MCTF can be simply illustrated by Fig. 1, in which only two lifting stages are involved. The prediction stage is using even frames to predict odd frames, and the residual frames are the highpass frames. The update stage is using the highpass frames to update the even frames, and the derived frames are the lowpass frames. The 1/3 MCTF is just to skip the update stage

This work was supported in part by National Science Council, Republic of China, under the grant number 91-2215-E-002-035 and in part by the MediaTek Fellowship.

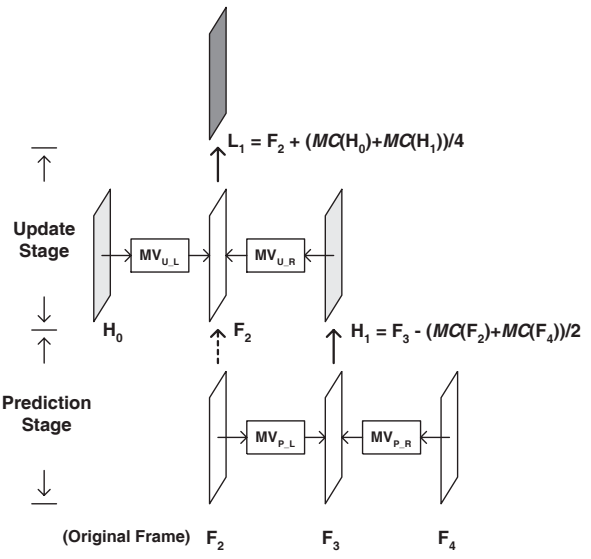

Fig. 1. The 5/3 MCTF scheme. MV_{P_L} and MV_{P_R} represent the motion vectors from the left and right neighbor frames for the prediction stage, respectively, and so represent MV_{U_L} and MV_{U_R} for the update stage. The light gray frames (H) are the highpass frames, and the heavy gray frames (L) are the lowpass frames.

of the 5/3 MCTF and treat the even frames as the lowpass frames. For aligning the objects in different frames, the two lifting stages both require motion vectors. The block-based motion model is usually adopted. For every block in the odd frames, ME should be performed to find the best motion vectors MV_{P_L} and MV_{P_R} in the prediction stage. As for the update motion vectors MV_{U_L} and MV_{U_R}, they are usually estimated and derived from MV_{P_L} and MV_{P_R} for saving the motion vector cost. Fig. 1 only shows the 1-level MCTF scheme, and it is the basic building block of the prediction schemes in SVC, including multi-level MCTF, In Band-MCTF, and multi-scale pyramid MCTF [8].

3. MACROBLOCK-LEVEL DATA REUSE FOR MOTION ESTIMATION

In the prediction stage, the ME is the most computation-costing part. Thus, the data reuse scheme of ME is still very important for MCTF, and the block-based motion model is assumed in the following. The block matching algorithm (BMA) is to find the best matched candidate block in the reference frame from the search region for every block in the current frame. The matching criteria is usually the Sum-of-Absolute-Difference (SAD). If the search range is $Horizontal : [-p_H, p_H)$ and $Vertical : [-p_V, p_V)$, the current block (CB) of size $B_H \times B_V$ and the corresponding search region are as shown in Fig. 2. The CB is usually of MB size in video coding systems. The full-search block matching algorithm (FSBMA) is to search every candidate block in the search region, and the fast BMA is trying to search much fewer candidate blocks than FSBMA.

In this paper, only the general data reuse scheme of the search region is discussed, which is the same for different detailed computation architectures. Obviously, the FSBMA needs to load the whole search region for every CB. The fast BMA can also load the whole search region or only load the selected candidate blocks.

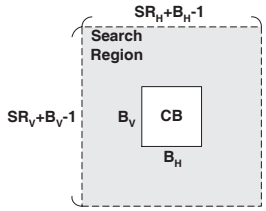

Fig. 2. The current block (CB) and search region for BMA. $SR_H = 2p_H$ and $SR_V = 2p_V$

However, the external memory bandwidth of the latter depends on the candidate block searching pattern of the adopted fast BMA algorithm and could be larger than the former that is the same as FSBMA. In the following, the data reuse schemes of FSBMA are reviewed, and a new scheme is also proposed.

3.1. Conventional schemes for FSBMA

Since the candidate blocks of one CB are overlapped a lot, and so are the search regions of neighboring CB's, there are four data reuse levels defined in terms of the degree of data reuse from Level A to Level D (weakest to strongest) [9, 10]. Two factors can be used to evaluate the performance of the data reuse schemes: local memory size for the reference frame and redundancy access factor (Ra). The local memory size represents the required memory size to buffer the data of candidate blocks. Ra is used to evaluate the memory bandwidth and defined as:

$$Ra = \frac{total\ memory\ bandwidth\ for\ reference\ frame}{minimum\ memory\ bandwidth\ (pixel\ count\ in\ total)}$$

The Level A scheme only reuses the overlapped pixels between two candidate blocks in the horizontal direction, and the Level B scheme reuses the overlapped pixels among candidate blocks in both horizontal and vertical directions. However, the Level C scheme is often used because of the current memory technology, and it reuses the horizontally overlapped region between two search regions of two neighboring CB's as shown in Fig. 3(a), in which only $B_H \times (SR_V + B_V - 1)$ pixels are required to loaded from external memory for every CB. Thus, the Ra of Level C scheme can be calculated as:

$$Ra_{Level\ C} \approx \frac{B_H \times (SR_V + B_V - 1)}{B_H \times B_V} \approx 1 + \frac{SR_V}{B_V} \qquad (1)$$

The required local memory size for the reference frame in the Level C scheme depends on the detailed implementation of ME architectures. In this paper, it is assumed to be one search region size, $(SR_H + B_H - 1)(SR_V + B_V - 1)$, such that MC can be performed immediately after ME without any other external memory access. As for the Level D scheme, it can minimize the memory access by fully reusing the horizontally and vertically overlapped search regions with a huge local memory size , $(W + SR_H - 1)(SR_V - 1)$, where W is the image width [10].

3.2. Proposed Level C+ scheme for FSBMA

The Level C scheme is always scanning the CB's in a raster scan fashion. Using a zig-zag scan fashion, we propose the Level C+ scheme that further reuses the vertically overlapped search regions but not fully reuses them as the Level D scheme. The proposed scheme is loading the search regions of the vertically neighboring CB's simultaneously. In another viewpoint, the current block is stretched in the vertical direction. For example, the new $B_{V,C+}$

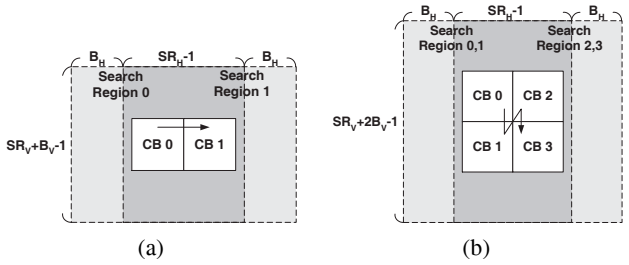

Fig. 3. Data reuse schemes for FSBMA. The heavy gray region is the overlapped and reused region. (a) Level C scheme. The CB's are scanned in a raster scan fashion; (b) Proposed Level C+ scheme. This is an example for $B_{V,C+} = 2B_V$ and $B_{H,C+} = B_H$, in which the CB's are scanned in a zig-zag scan fashion.

Table 1. Comparisons of MB-level data reuse schemes. The M in the general case is the number of reference frames for one current frame (as H.264/AVC). In the case of HDTV 1280×720 at 30 fps, five reference frames are used, and $B_H = B_V = 16$, $SR_H = SR_V = 64$, $B_{H,C+} = 2B_H$, and $B_{V,C+} = 2B_V$ are assumed. (MemBw: Memory Bandwidth; LMS: Local Memory Size.)

Reuse Scheme	General Case		HDTV720p 5 ref. frame	
	MemBw (pixels/pixel)	LMS for Ref. Frame (pixels)	MemBw (MB/sec)	LMS for Ref. Frame (KB)
Level C	$1+M*Ra_{Level\ C}$	$(SR_H+B_H-1) \times (SR_V+B_V-1)$	718.8	31.2
Level D	$1+M$	$(W+SR_H-1) \times (SR_V-1)$	165.9	423.0
Proposed Level C+	$1+M*Ra_{Level\ C+}$	$(SR_H+B_{H,C+}-1) \times (SR_V+B_{V,C+}-1)$	442.4	45.1

can be $2B_V$ as shown in Fig. 3(b) or even $3B_V$. Thus, the Ra of the proposed Level C+ scheme is:

$$Ra_{Level\ C+} \approx 1 + \frac{SR_V}{B_{V,C+}} \tag{2}$$

However, the zig-zag scan in the Level C+ scheme may violate the data dependency of MB's in some video standards. For example, before one MB is coded, its left, top, and top-right MB's should be coded first in H.264/AVC. Thus, the Level C+ scheme may need to stretch the CB in the horizontal direction and modify the zig-zag scan. The modification depends on the adopted encoder architecture. For H.264/AVC, $B_{H,C+}$ is required to be larger than or equal to $2B_H$ if $B_{V,C+} = 2B_V$. For comparison, the general case and real-life HDTV case are listed in Table 1 for the Level C, Level D, and proposed Level C+ schemes. The total memory bandwidth is the sum of the memory bandwidths for the current frame and reference frames. As shown in Table 1, the proposed Level C+ scheme can provide a good trade-off between the Level C and D schemes.

4. MEMORY ANALYSIS OF 1-LEVEL 5/3 AND 1/3 MCTF

The main difference between MCTF and MCP is that the reference frames in MCTF are the original frames and those in MCP are the reconstructed frames. Thus, the ME in MCP needs to be performed in a frame-by-frame fashion (the close-loop property). In MCTF, the ME of different frames can be performed simultaneously. In the following, we propose frame-level data reuse schemes for MCTF and compare them with direct implementation.

4.1. Prediction stage

In MCTF, the ME is only performed in the prediction stage, and the update stage only performs MC. The inputs of the prediction stage are the original frames, and the outputs are the highpass frames that are derived after ME and MC as shown in Fig. 1. The left and right branches of ME in the prediction stage are assumed to be separately performed as [8]. (Integer ME are performed separately, and, however, the locally refined fractional-pel ME can be optimized jointly.)

The direct implementation is to perform the left and right ME separately as shown in Fig. 4(a). The memory access (pixels/pixel) is as follows:

$$\underbrace{[(1+Ra)}_{Left\ ME} + \underbrace{1}_{Left\ MC} + \underbrace{1}_{H\ output} + \underbrace{(1+Ra)]}_{Right\ MC\&ME}/2 = Ra+2$$

where the divisor two exists because the prediction stage is performed for every two frames. The required total local memory is one current block buffer (CBB) and one search region buffer (SRB). SRB corresponds to the third column of Table 1 and depends on which MB-level data reuse scheme is used.

Two frame-level data reuse schemes for the ME are proposed, Double Reference Frames (DRF) and Double Current Frames (DCF), as shown in Fig. 4(b) and (c), respectively. The total local memory size of DRF is $1CBB + 2SRB$, and the memory access is:

$$\underbrace{[\quad 2Ra}_{Left\&Right\ Ref.} + \underbrace{1}_{Cur.\ input} + \underbrace{1]}_{H\ output}/2 = Ra+1$$

As for the DCF, the total local memory size is $2CBB + 1SRB$, and the memory access is:

$$\underbrace{[Ra}_{Ref.\ R_1\ input} + \underbrace{1}_{R_0\ MC} + \underbrace{1}_{H\ output} + \underbrace{2]}_{Left\&Right\ Cur.}/2 = Ra/2+2$$

The three mentioned reuse schemes are compared in Table 2. The DCF scheme can provide the smallest memory bandwidth if $Ra >= 2$, and the local memory size is also smaller than the DRF scheme and nearly equal to the separate scheme because the CBB is usually much smaller than SRB.

4.2. Combined prediction and update stages

In the update stage, only the MC is performed, and the motion vectors are derived from those in the prediction stage. Similarly, the MC of the update stage can also be performed by use of the DRF or DCF scheme. The DRF scheme of the update stage is as shown in the top of Fig. 5(a), and the memory access is:

$$\underbrace{(\quad 2}_{Left\&Right\ Ref.} + \underbrace{1}_{Cur.\ input} + \underbrace{1)}_{L\ output}/2 = 2$$

On the other hand, the DCF scheme is shown in the top of Fig. 5(c), and the memory access is:

$$\underbrace{(\quad 2}_{Left\&Right\ Cur.} + \underbrace{2}_{Ref.\ input} + \underbrace{2)}_{L\&U_L\ output}/2 = 3$$

Thus, combining the frame-level data reuse schemes of the prediction and update stages, there are four possible schemes as shown in Fig. 5 (The direct separate scheme is neglected here). In Fig. 5, the frames expressed by bold lines represent those need to be stored in the external memory for the 5/3 MCTF, and the frame delay numbers of the lowpass frames can also be found by counting the distance between the lowpass frame and the newest input frame. Besides, the required local memory is determined by the prediction

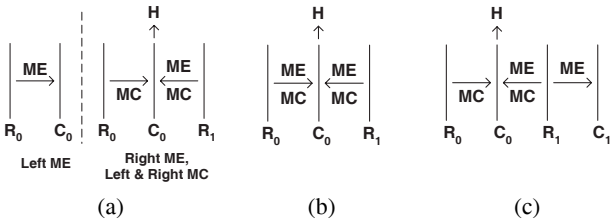

Fig. 4. Data reuse schemes for the prediction stage. (C: Current frame; R: Reference frame.) (a) Separate left and right ME; (b) Double reference frames ME; (c) Double current frames ME.

Table 2. Comparisons of frame-level data reuse schemes for the prediction stage of 5/3 MCTF.

Reuse Scheme	Separate Left & Right	Proposed DRF	Proposed DCF
MemBw (pixels/pixel)	Ra+2	Ra+1	Ra/2+2
LMS (pixels)	1CBB+1SRB	1CBB+2SRB	2CBB+1SRB

stage and is the same as Table 2. These four combined schemes of the 5/3 MCTF are summarized in Table 3. Furthermore, the data reuse schemes of the 1/3 MCTF are also listed. From this table, the update stage is always suggested to adopt the DRF scheme, not the DCF scheme. If $Ra >= 2$, using the DCF scheme for the prediction stage will require less memory bandwidth but more external storage and lowpass frame delay than using the DRF one.

5. CONCLUSION

In this paper, we analyze the memory issues of the core technology MCTF in the future video standard SVC. The MB-level data reuse scheme for ME is analyzed first, and a new Level C+ scheme is proposed, which is also useful for existing video standards. The memory issues for the 5/3 and 1/3 MCTF are discussed in terms of memory bandwidth, local memory size, minimum external storage size, and lowpass frame encoding delay. Two frame-level data reuse schemes, DRF and DCF, are proposed according to the open-loop MCTF nature. This analysis can be used as a reference for implementing the interframe wavelet video coding, and the detailed bandwidth and memory size depend on the adopted data reuse scheme, search range size, and frame size. Our future work will extend this work to Inverse MCTF, multi-level MCTF, IB-MCTF, and multi-scale pyramid MCTF schemes.

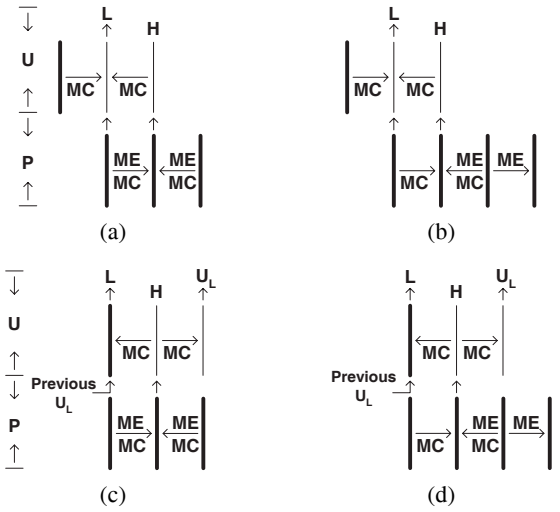

Fig. 5. Four frame-level data reuse schemes of 5/3 MCTF. (P: Prediction stage; U: Update stage; U_L: Partial result for L.) (a) P-DRF/U-DRF; (b) P-DCF/U-DRF; (c) P-DRF/U-DCF; (d) P-DCF/U-DCF.

Table 3. Comparisons of frame-level data reuse schemes for the 5/3 and 1/3 MCTF. (ES: External Storage; LFD: Lowpass Frame Delay.)

Prediction	DRF	DCF	DRF	DCF	DRF	DCF
Update	DRF	DRF	DCF	DCF	-	-
MemBw (pixels/pixel)	Ra+3	Ra/2+4	Ra+4	Ra/2+5	Ra+1	Ra/2+2
ES (frames)	4	5	4	5	3	4
LFD (frames)	2	3	2	3	0	0

6. REFERENCES

[1] J.-R. Ohm, "Advanced packet-video coding based on layered VQ and SBC techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 3, pp. 208–221, June 2002.

[2] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proc. IEEE International Conference on Image Processing*, 2001, pp. 1029–1032.

[3] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, pp. 1793–1796.

[4] D. Taubman, "Successive refinement of video: fundamental issues, past efforts and new directions," in *International Symposium on Visual Communications and Image Processing*, 2003, pp. 791–805.

[5] ISO/IEC JTC1, "Call for proposals on scalable video coding technology," ISO/IEC JTC1/WG11 Doc. N5958, Oct. 2003.

[6] J. Xu, et al., "3D subband video coding using barbell lifting," ISO/IEC JTC1/WG11 Doc. M10569/S05, Mar. 2004.

[7] H. Schwarz, D. Marpe, and T. Wiegand, "Scalable extension of H.264/AVC," ISO/IEC JTC1/WG11 Doc. M10569/S03, Mar. 2004.

[8] ISO/IEC JTC1, "Scalable Video Model 2.0," ISO/IEC JTC1/WG11 Doc. N6520, July 2004.

[9] Mei-Yun Hsu, "Scalable module-based architecture for MPEG-4 BMA motion estimation," M.S. thesis, National Taiwan Univ., June 2000.

[10] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 1, pp. 61–72, Jan. 2002.