Frame-level Data Reuse for Motion-Compensated Temporal Filtering

Ching-Yeh Chen, Yi-Hau Chen, Chih-Chi Cheng, and Liang-Gee Chen
DSP/IC Design Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan
Email: {cychen, ttchen, ccc, lgchen}@video.ee.ntu.edu.tw

Abstract— **Motion-compensated temporal filtering (MCTF) is an open-loop prediction scheme, so the frame-level data reuse for MCTF is possible. In this paper, we propose two general frame-level data reuse schemes which can minimize the memory bandwidth of current and reference frames, respectively. And their relationships between the required memory bandwidth and the number of searching range buffers are also formulated under the constraint of the data dependency in Joint Scalable Video Model. Finally, we extend our analysis to pyramid MCTF and the impact of the inter-layer prediction scheme is also considered.**

I. Introduction

Motion-compensated temporal filtering (MCTF) is a new prediction scheme to remove the temporal redundancy of a video sequence. The concept of MCTF is to perform a wavelet transform in the temporal direction with motion compensation (MC) [1]. Based on MCTF, many open-loop video coding schemes are developed in order to provide spatial, temporal, or SNR scalability for many future applications. Currently, MPEG is standardizing a video coding standard, scalable video coding (SVC), for these applications, and the scalable extension of H.264/AVC with MCTF [2] is adopted as the reference software of SVC, Joint Scalable Video Model (JSVM). Compared to traditional close-loop video codings, the open-loop MCTF prediction scheme not only can avoid the catastrophic error propagation, which is due to the mismatch of the reconstructed frames between the encoder and decoder, but also can improve the coding performance of H.264/AVC [3].

In our previous works [4], [5], the multi-level MCTF with 5/3 or 1/3 filter is analyzed and several basic frame-level data reuse schemes are discussed. But the data dependency between the groups of pictures (GOPs) in JSVM is not considered. In this paper, we will consider this constraint and extend our previous works to two general frame-level data reuse schemes, in which the relationships between the required memory bandwidth and the number of searching range buffer are also derived. Moreover, the pyramid MCTF will be analyzed, including computational complexity, external memory bandwidth, and external memory size. Finally, this paper is organized as follows. In Section II, the pyramid MCTF in JSVM2.0 are introduced. The proposed frame-level data reuse schemes and the analysis of pyramid MCTF are presented in Sections III and IV, respectively. Section V will conclude this paper.

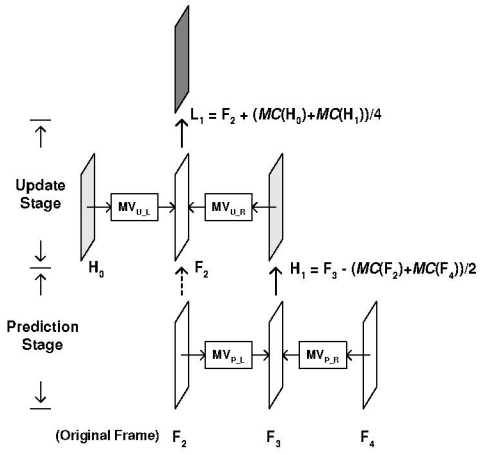

Fig. 1. The 5/3 MCTF scheme, where the light gray frames (H) are the highpass frames, and the heavy gray frames (L) are the lowpass frames.

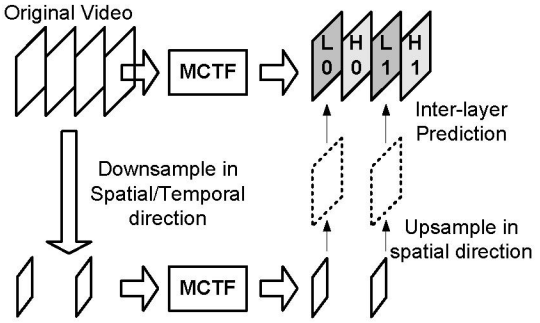

Fig. 2. The pyramid MCTF scheme with two spatial layers.

II. Motion-Compensated Temporal Filtering

MCTF is to perform a wavelet transform in the temporal direction with MC, and its coding performance depends on which wavelet filter is adopted. MCTF is usually implemented by use of the 5/3 filter with lifting scheme, because it can provide a good coding performance. For the sake of simplicity, MCTF represents the lifting-based MCTF using 5/3 or 1/3 filter in the following.

Figure 1 shows the operation of 5/3 MCTF, which consists of two lifting stages, prediction and update stages. The former

is using even frames to predict odd frames, and the generated residual frames are the highpass frames (H-frames). The latter is using H-frames to update the even frames, and the derived frames are the lowpass frames (L-frames). If the update stage of 5/3 MCTF is skipped, we call it as 1/3 MCTF and treat the even frames as L-frames. Note that MC is required in both stages for aligning the objects in different frames, but motion estimation (ME) is only performed in the prediction stage, in which the block-based motion model is usually adopted to find the best motion vectors, MV_{P_L} and MV_{P_R}. The motion vectors in the update stage, MV_{U_L} and MV_{U_R}, are derived from MV_{P_L} and MV_{P_R} for saving the motion vector cost. Figure 1 only shows the one-level MCTF scheme. The multi-level MCTF can be achieved by recursively performing one-level MCTF on the L-frames.

In JSVM, the spatial scalability is provided by pyramid MCTF, as shown in Fig. 2 which consists of two spatial layers. In order to provide spatial scalability, a new sequence with a smaller frame size is generated by downsampling the original sequence in the spatial direction, but the redundancy between different spatial layers is also induced. Hence *Inter-layer Prediction* is developed to remove this redundancy, in which we can use the sequence with a small frame size to predict the sequence with a large frame size. In pyramid MCTF, each spatial sequence will be predicted from 5/3 MCTF or *Inter-layer Prediction*. If more scalable spatial layers are wanted, more sequences with smaller frame sizes are generated and processed by the same procedure.

III. Frame-Level Data Reuse Schemes

In MCTF, ME takes the most part of computation and memory bandwidth. However, compared to traditional video coding, the main difference is that the reference frames in MCTF are the original or filtered frames, not the reconstructed frames. Hence in MCTF, the ME of different frames can be processed simultaneously, so the frame-level data reuse becomes possible. Although the frame-level data reuse is feasible, the data dependency between GOPs in JSVM limits the efficiency of the data reuse scheme. Therefore, in the following subsections, we not only propose two general frame-level data reuse schemes but also consider the data dependency between GOPs.

A. Previous Works

In [4], we proposed two basic frame-level data reuse schemes for prediction stages, double reference frames (DRF) and double current frames (DCF), as shown in Fig. 3 (a) and (b). DRF performs the bi-directional ME for every current block together. Therefore, for one current block, the data of one current block and two searching ranges are required to be accessed. DCF is proposed to reduce the memory bandwidth by sharing the searching range data for two current blocks in different current frames. Therefore, DCF not only saves half memory bandwidth but also reduces half searching range buffers (SR buffers) of DRF.

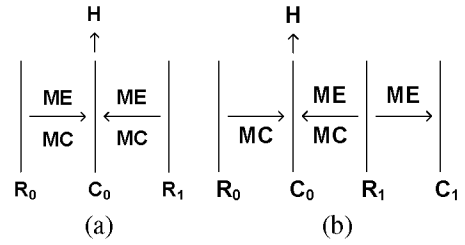

Fig. 3. The frame-level data reuse schemes (C: Current frame; R: Reference frame): (a) Double reference frames; (b) Double current frames.

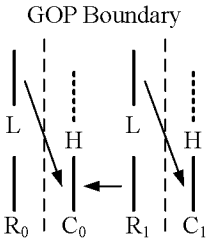

Fig. 4. The data dependency between GOPs in JSVM.

However, due to the data dependency between GOPs in JSVM, the performance of DCF scheme will be degraded. The data dependency is that for one GOP, the reference frame is the last L-frame not the original frame of the previous GOP, as shown in Fig. 4. Hence the frame-level data reuse is limited in one GOP.

B. Proposed Extended Double Reference Frames

Figure 3(a) shows the basic DRF. Because of the open-loop video coding scheme, we can further cascade several DRF to reduce more memory bandwidth, which is called the extended DRF scheme (EDRF), as shown in Fig. 5(a). In EDRF, the number of reference frames is always one more frame than that of current frames. Therefore, if there are N SR buffers, we can process $N-1$ current frames simultaneously, and then the relationship between the required memory bandwidth (BW_{EDRF}) and the number of SR buffers (N) is equal to

$$\begin{aligned} BW_{EDRF} &= \alpha\{(N-1)cur + Nref\} \\ &\quad + \gamma\{\beta cur + (\beta+1)ref\} + Kcur, \\ \alpha &= \left\lfloor \frac{K}{N-1} \right\rfloor,\ \beta = K - \alpha(N-1),\ \gamma = \begin{cases} 1, & if \beta > 0 \\ 0, & if \beta = 0 \end{cases}, \end{aligned} \quad (1)$$

where K is the number of total current frames which can be parallel processed, and cur and ref are the required memory bandwidth of one current frame and one reference frame, respectively. In (1), α is the number of EDRF, β is the remainder, γ is used to check if the remainder exists or not,

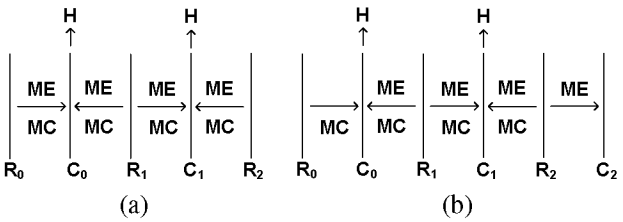

Fig. 5. The frame-level data reuse schemes (C: Current frame; R: Reference frame): (a) Extended double reference frames with $N = 3$; (b) Extended double current frames with $N = 2$.

TABLE I

THE COMPARISON OF THE REQUIRED MEMORY BANDWIDTH WITH EDCF AND EDRF IN ONE GOP OF JSVM.

Scheme	1 Level	2 Level	3 Level	4 Level	SR Buffer
EDRF (N=2)	$2cur + 2ref + 0MC$	$6cur + 6ref + 0MC$	$14cur + 14ref + 0MC$	$30cur + 30ref + 0MC$	2
EDRF (N=3)	$2cur + 2ref + 0MC$	$6cur + 5ref + 0MC$	$14cur + 11ref + 0MC$	$30cur + 23ref + 0MC$	3
EDCF (N=1)	$4cur + 2ref + 1MC$	$10cur + 5ref + 3MC$	$22cur + 10ref + 7MC$	$46cur + 19ref + 15MC$	1
EDCF (N=2)	$2cur + 2ref + 0MC$	$7cur + 5ref + 1MC$	$17cur + 10ref + 3MC$	$37cur + 19ref + 7MC$	2
EDCF (N=3)	$2cur + 2ref + 0MC$	$6cur + 5ref + 0MC$	$15cur + 10ref + 1MC$	$33cur + 19ref + 3MC$	3

and $Kcur$ is the memory bandwidth of outputted H-frames.

C. Proposed Extended Double Current Frames

Similarly, based on DCF, we can develop the extended DCF scheme (EDCF), as shown in Fig. 5(b), which cascades two DCF. For EDCF, the number of reference frames is one less frame than that of current frames. Hence the relationship between the required memory bandwidth (BW_{EDCF}) and the number of SR buffer (N) is

$$\begin{aligned} BW_{EDCF} &= \alpha\{(N+1)cur + Nref + MC\} \\ &\quad + \gamma\{(\beta+1)cur + \beta ref + MC\} + Kcur \\ &\quad - (2cur + MC), \\ \alpha &= \left\lfloor \frac{K+1}{N} \right\rfloor,\ \beta = (K+1) - \alpha N,\ \gamma = \begin{cases} 1, & if \beta > 0 \\ 0, & if \beta = 0 \end{cases}, \end{aligned} \quad (2)$$

where the notations are the same as (1), MC is the required memory bandwidth of MC, and the final part, $2cur+MC$, is resulted from that the prediction scheme of MCTF is $RCR..CR$, not $CRC..RC$.

D. Case Study

In this subsection, we further discuss the performances of EDRF and EDCF. Table I shows the required memory bandwidth of EDRF and EDCF with the various number of SR buffer (N) and different decomposition levels. From Table I, EDCF can minimize the memory bandwidth of reference frames but the memory bandwidth of current frames and MC are its overhead. Contrarily, EDRF minimizes the memory bandwidth of current frames but has a larger memory bandwidth of reference frames.

If there are the same SR buffers (N) in both schemes, EDCF has much less memory bandwidth of reference frames but with larger memory bandwidth of current frames and MC, compared to EDRF. In general, the memory bandwidth of one reference frame is much larger than that of one MC, and that of one MC is also larger than that of one current frame. Hence the tradeoff between the reference frames, current frames and MC in EDCF is worth in the average case.

Moreover, when the number of decomposition levels is increased, the performance of EDCF is become better because the impact of data dependency between GOPs is become less. If we increase the number of SR buffers (N), the overhead of EDCF can be reduced and it is possible for EDRF to share the searching range data between two current frames. Hence the required memory bandwidth of EDCF and EDRF can be further reduced, and because of the share of searching range, the reduction ratio of EDRF is larger than that of EDCF.

TABLE II

THE REQUIRED MEMORY BANDWIDTH WITH EDCF AND EDRF IN D1 FORMAT WITH SEARCHING RANGE [-64,64) AND 30FPS.

Scheme	1 Level MB/s	2 Level MB/s	3 Level MB/s	4 Level MB/s	SR Buffer KBytes
EDRF (N=2)	103.7	155.5	181.4	194.4	41.4
EDRF (N=3)	103.7	132.2	146.4	153.6	62.2
EDCF (N=1)	124.4	158.1	163.3	160.1	20.7
EDCF (N=2)	103.7	140.0	146.4	143.9	41.4
EDCF (N=3)	103.7	132.2	138.7	136.1	62.2

Table II shows a real case, in which the specification is D1 Format, 30 frames per second (fps), and the searching range is $[-64, 64)$. We assume that full search and Level C scheme [6] are adopted in ME. As for MC, we assume that the modified DCF scheme in [5] is used. By these assumptions, the memory bandwidth of one reference frame and one MC are nine and two times of that of one current frame, respectively.

In Table II, although the EDCF with N=1 can share the searching range between two current frames at two decomposition levels, the required memory bandwidth is still larger than that of EDRF with N=2. This is due to the overhead of EDCF and the constraint of data dependency between GOPs, so the performance of EDCF is degraded. Under the same hardware resources (N=2) with four decomposition levels, the memory bandwidth reduction of EDCF is 26.3%, compared to EDRF. Moreover, the required memory bandwidth of EDCF with N=2 is also less than that of EDRF with N=3. When the number of SR Buffers is increased, the memory bandwidth of EDRF and EDCF is reduced apparently but the required SR buffer size is increased largely.

Finally, compared to EDRF, the memory bandwidth reduction of EDCF is dependent on the searching range and what kind of macroblock-level searching range data reuse schemes is adopted. The larger the required memory bandwidth of one reference frame is, the better the performance of EDCF is.

IV. ANALYSIS OF PYRAMID MCTF IN JSVM

After we proposed two general frame-level data reuse schemes for multi-MCTF, we focus on the analysis to pyramid MCTF. Before starting the analysis of pyramid MCTF, *Inter-layer Prediction* should be introduced first. *Inter-layer Prediction* means that we can use the information of the sequence with a small frame size to predict the sequence with a large frame size. In JSVM, when the inter-layer prediction modes are considered, ME will be processed twice. One is the original and the other is with the information of previous spatial layer. In the following analysis, we assume the downsample ratio

between two spatial layers is δ, and there are J spatial layers, in which the largest frame size is 1, the smallest frame size is δ^{J-1}, and the frame rates in different spatial layers are the same.

A. Computational Complexity

For the computational complexity, motion estimation is still a major part in pyramid MCTF. In each spatial layer, the searching range and searching strategy of ME can be different. But for simplicity, we assume all parameters of ME are the same for each spatial layer. Hence the computational complexity of each current macroblock is the same, and then the computational complexity is direct proportional to the frame size. The computational complexity of pyramid MCTF with *Inter-layer Prediction* will be

$$\begin{aligned} CC_{J-level} =& (1+\delta+\delta^2+...+\delta^{J-1})CC_{1-level} \\ &+ (1+\delta+...+\delta^{J-2})CC_{1-level}, \end{aligned} \quad (3)$$

where $CC_{J-level}$ and $CC_{1-level}$ are the computational complexity of pyramid MCTF with J spatial layers and MCTF with the largest frame size, respectively. In (3), the first part is the sum of the computational complexity of MCTF in each spatial layer, and the second part is that of *Inter-layer Prediction*. Therefore, if *Inter-layer Prediction* is not adopted or the ME of *Inter-layer Prediction* can be skipped, the computational complexity will be only the first part.

B. Memory Bandwidth

Because all parameters of ME are the same, memory bandwidth is also direct proportional to the frame size. The required memory bandwidth consists of two parts, the memory bandwidth of each spatial layer and that of *Inter-layer Prediction*, and it can be written as

$$\begin{aligned} BW_{J-level} =& (1+\delta+\delta^2+...+\delta^{J-1})BW_{1-level} \\ &+ (\delta+...+\delta^{J-1}) \times FrameSize \times 30, \end{aligned} \quad (4)$$

where $BW_{J-level}$ and $BW_{1-level}$ are the memory bandwidth of pyramid MCTF with J spatial layers and MCTF with the largest frame size, respectively. The first part is the memory bandwidth of each spatial layer, and the second part is that of *Inter-layer Prediction*, in which the information of the sequence with the small frame size has to be loaded from external memory. Therefore, the extra memory bandwidth for *Inter-layer Prediction* is required. In the second part of (4), the series of δ is the amount of data per frame for *Inter-layer Prediction*, $FrameSize$ is the memory bandwidth of one frame, and 30 is 30 fps.

C. External Memory Storage

Due to the inter-layer prediction scheme, J spatial layers cannot be parallel processed. Therefore, the coding order is from the sequence with the smallest frame size to that with the largest frame size. Then, the external memory storage is

$$\begin{aligned} EMS_{J-level} =& EMS_{1-level} + \delta \times FrameSize \times GOPSize \\ &+ (\delta+...+\delta^{J-1}) \times FrameSize, \end{aligned} \quad (5)$$

where $EMS_{J-level}$ and $EMS_{1-level}$ are the memory storages of pyramid MCTF with J spatial layers and MCTF with the largest frame size respectively, and the $GOPSize$ is the number of frames in one GOP. The first part is the original for the sequences with the largest frame size, the second part is used to store a GOP in the previous spatial layer for the inter-layer prediction scheme, and the third part is used to store the L frames of the previous GOP in each spatial layer.

D. Summary of Pyramid MCTF

The computational complexity and memory bandwidth are direct proportional to the frame size of each spatial layer, when all parameters of ME are the same for each spatial layer of pyramid MCTF. Hence the required computational complexity and memory bandwidth are exponential decreased for these downsampled sequences, but the external memory storage can be reused, except the last L frame of the previous GOP in each spatial layer. As for *Inter-layer Prediction*, the computational complexity will be doubled, the increase of the external memory storage depends on the size of a GOP, and its required memory bandwidth is dependent on the frame size.

Finally, we give an example, in which the downsample ratio is 2 in both directions (δ=$\frac{1}{4}$), and the size of GOP is 16. The computational complexity and memory bandwidth of pyramid MCTF without *Inter-layer Prediction* is close to $\frac{4}{3}$ times of the originals, and the external memory storage increases a little. As for *Inter-layer Prediction*, the computational complexity will be doubled, and the extra increase of the memory bandwidth and external memory storage are close to $10Frames/s$ and $4Frames$, respectively.

V. Conclusion

In this paper, for open-loop MCTF, we proposed two general frame-level data reuse schemes, EDRF and EDCF, in which their relationships between the required memory bandwidth and SR buffers are derived. Under the same hardware resources, the performance of EDCF is better than that of EDRF, because EDCF can share the searching range data between two current frames. Finally, we provide the analysis of pyramid MCTF with or without the inter-layer prediction scheme. The inter-layer prediction scheme can remove the redundancy between two spatial layers. However, it requires double computation complexity and the increases of the required memory bandwidth and memory storage depend on the frame size and the size of a GOP, respectively.

References

[1] 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.

[2] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model (JSVM) 2.0 Reference Encoding Algorithm Description," ISO/IEC JTC1/SG29/WG11 Doc. N7084, Apr. 2005.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "MCTF and scalability extension of H.264/AVC," in *Proc. Picture Coding Symposium*, 2004.

[4] C.-T. Huang, C.-Y. Chen, Y.-H. Chen, and L.-G. Chen, "Memory analysis of VLSI architecture for 5/3 and 1/3 motion-compensated temporal filtering," in *Proc. ICASSP*, 2005.

[5] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, C.-J. Lian, and L.-G. Chen, "System analysis of VLSI architecture for motion-compensated temporal filtering," in *Proc. ICIP*, 2005.

[6] 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 Trans. CSVT*, vol. 12, no. 1, pp. 61–72, Jan. 2002.