

# Analysis and VLSI Architecture of Update Step in Motion-Compensated Temporal Filtering

Chih-Chi Cheng, 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: {ccc,cychen,ttchen,lgchen}@video.ee.ntu.edu.tw

**Abstract**—Motion-compensated temporal filtering (MCTF) is the core technology of the next generation scalable video coding schemes. There are two key steps in MCTF process, prediction step and update step. In this paper, an efficient scheme along with the hardware architecture of update step for VLSI implementation is proposed. The proposed update step scheme can reduce 55% off-chip memory access and turn the irregular access into regular access. 25% on-chip memory access is also reduced in deriving inverse motion vectors. 80% hardware area is saved by reusing the hardware resources of prediction step.

## I. INTRODUCTION

Motion-compensated temporal filtering (MCTF) is the core technology of the next generation scalable video coding schemes such as scalable extension of H.264/AVC [1]. The basic idea of MCTF is to perform the discrete-wavelet transform (DWT) in temporal domain on motion-threaded pixels [2]. As in spatial-domain DWT, the operations in MCTF are slightly different between different adopted DWT filters.

Figure 1 shows the basic scheme of single-level (5,3) MCTF. All MCTF operations can be classified into two steps, the first one is the prediction step and the second one is the update step. As shown in the lower half of Fig. 1, in prediction step, pixels in different frames are motion threaded by block-based motion estimation (ME) [2]. After ME, motion compensation (MC) is performed on pixels of the same motion thread with weighting according to the adopted DWT filter.

In update step, each  $4 \times 4$  block in the reference frame should have an inverse motion vector (MV) to maintain the motion thread. An example of deriving inverse MV can be seen in the upper part of Fig. 1. The dotted arrows are the MVs determined at prediction step, and the solid arrow is the derived inverse MV. After deriving the inverse MVs, MC is applied again to produce the low-band frame with weighting according to the adopted DWT filter.

Update step is a new video coding technique that appears firstly in MCTF. In this paper, the scheduling and hardware architecture of update step are proposed for VLSI implementation. As shown in [3], off-chip memory access can occupy more than 60% total power of a handheld system. The target of the proposed scheme is to reduce the off-chip memory access without increase of the on-chip buffer size of MCTF system. With the proposed scheme, the off-chip memory access can be reduced by 53% and turned from irregular access into regular access. 25% on-chip memory access is also reduced in deriving

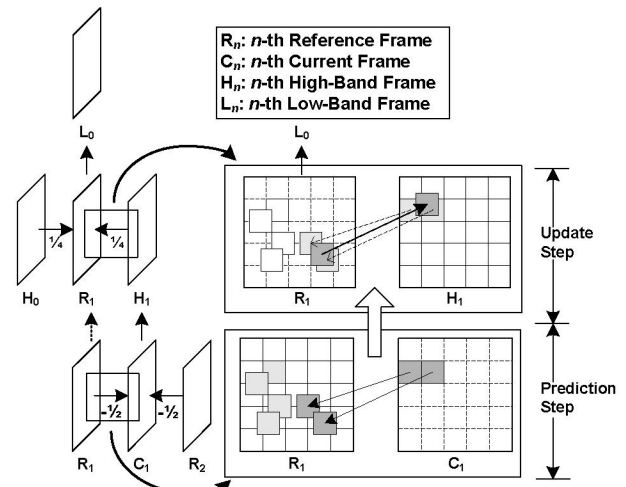


Fig. 1. The scheme of (5,3) MCTF and the basic operations of prediction and update step.

inverse MV. About 80% area is saved by reusing the MC engine in fractional ME of prediction step.

This paper is structured as follows. Section II presents the proposed scheme of deriving inverse MVs. Section III presents the proposed ME-based level C+ MC. Hardware architecture of update step is proposed in Section IV. Experimental results are shown in Section V, and Section VI concludes this work.

## II. THE PROPOSED SCHEME OF DERIVING INVERSE MVs

Figure 2 shows the operations for deriving inverse MVs of one direction. Each  $4 \times 4$  block in high-band frame (ex. block 1) will overlap 4 blocks in the reference frame (block  $a$ ,  $b$ ,  $c$ , and  $d$ ).

For a block in reference frame, searching all blocks overlapping to it in search range is required if deriving inverse MVs is processed MB-by-MB (also block-by-block) of reference frame. To avoid the induced high computation complexity, the preferred scheme is to deriving inverse MVs MB-by-MB of high-band frame.

In this scheme, motion information that contains currently selected inverse MV and the corresponding overlapped area is buffered for each  $4 \times 4$  block of reference frame. For example, for block 1 in Fig. 2, motion information of block  $a$ ,  $b$ ,  $c$ , and  $d$  is read from an inverse MV buffer. If the overlapped area induced by currently processed MV is larger than the stored one, the corresponding motion information is updated and written back to inverse MV buffer.

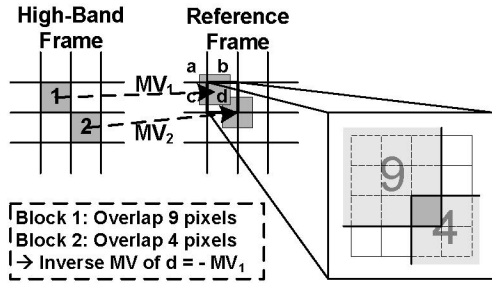


Fig. 2. The operations of deriving inverse MV. Two  $4 \times 4$  blocks, block 1 and block 2, overlap block  $d$  with 9 and 4 pixels, respectively. The inverse MV of block  $d$  is therefore  $-MV_1$ .

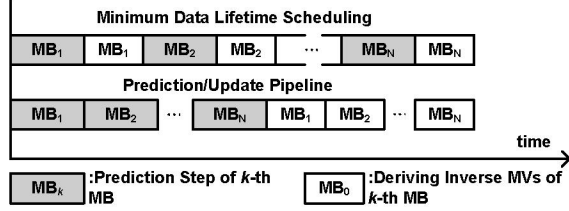


Fig. 3. Two different task scheduling between prediction step and deriving inverse MV in a frame (assume there are  $N$  MBs in a frame).

Since both the prediction step and the deriving inverse MVs are processed MB-by-MB of high-band frame, there is a chance to interleave the processing of prediction and deriving inverse MVs. Figure 3 shows two possible types of scheduling. The first type is the minimum data lifetime scheduling, and the second type is the proposed prediction/update pipeline.

#### A. Minimum Data Lifetime Scheduling

When a  $4 \times 4$  block in high-band frame is generated, its motion vector has to be used for deriving inverse MVs. If this MV is not used out immediately, it has to be stored in off-chip memory for update step. This will induce additional off-chip memory access. The minimum data lifetime scheduling is to be used out the MV right after it is generated to reduce the off-chip memory access.

Once a  $4 \times 4$  block finishes its prediction step, motion information of the overlapped four  $4 \times 4$  blocks in reference frame has to be read from off-chip memory. The motion information of the four overlapped blocks is then updated and written back to off-chip memory.

The required off-chip memory access of processing one  $4 \times 4$  block in high-band frame is therefore  $2(\text{forward and backward}) \times [4(4 \text{ overlapped blocks}) \times 2(\text{read and write}) \times (\text{the number of bits for motion information})]$ . The overlapped area ranges from 0 pixels to 16 pixels, and it requires 5 bits to be represented. Assume a MV occupies 16 bits, then the off-chip memory access for one  $4 \times 4$  block of high-band frame is 336 bits. Please note that it is irregular memory access.

#### B. The Proposed Prediction/Update Pipeline

The proposed prediction/update pipeline is to separate the prediction step and the update step of one frame. In this case, the MVs of high-band frame have to be stored in the off-chip memory. This will induce off-chip memory access of one read and one write of each MV in high-band frame.

However, once the prediction step is finished, the search range buffer for ME in prediction step is available to buffer

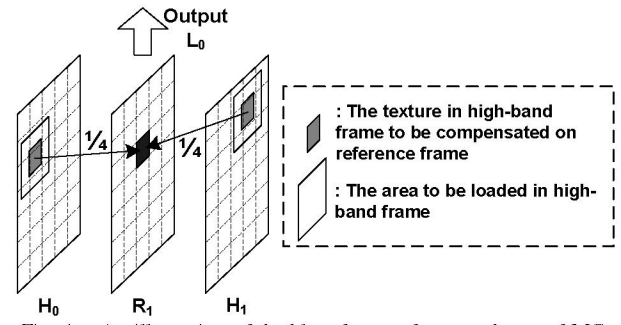


Fig. 4. An illustration of double reference frames scheme of MC.

the motion information. This search range buffer is usually sufficient for using level D data reuse scheme [4] to buffer the motion information. The level D data reuse scheme can achieve minimum off-chip memory access of reference frame. After finishing processing one MB row, the motion information of the top MB row in the buffer will contain the final inverse MVs.

For one  $4 \times 4$  block in high-band frame, the off-chip memory access is  $2(\text{forward and backward}) \times [(2(\text{read and write MV of current block}) + 1(\text{write inverse MV})) \times (\text{the number of bits for one MV})]$ . If a MV occupies 16 bits, this number will be 96 bits. Compared with Section II-A, it is a 71% reduction.

### III. THE PROPOSED ME-BASED LEVEL C+ MC

#### A. The Double Reference Frames Scheme

During MC, a  $4 \times 4$  block in the reference frame loads texture of left and right high-band frames according to the derived inverse MV. A low-band frame is produced after a reference frame accomplishes the bi-directional MC in update step. If double reference frames scheme [5] in Fig. 4 is utilized, both sides of MC are accomplished at the same time, and there is no off-chip memory access caused by partial results of MC.

Please note that the loaded pixels is larger than  $4 \times 4$  because of the need of interpolation for fractional-pixel accuracy motion. In JSVM [1], the interpolation filter with 6 taps is utilized, and  $10 \times 10$  pixels have to be loaded for a  $4 \times 4$  block. Assume a pixel is represented as 8 bits, the off-chip memory access for MC of one  $4 \times 4$  block is therefore  $[2(\text{read } R_1 \text{ write } L_0) \times (4 \times 4) + 2(\text{forward and backward}) \times (10 \times 10)] \times 8 = 1856$  bits. The memory access is irregular since the inverse MV can point to anywhere within search range.

#### B. The Proposed ME-based Level C+ MC

In MCTF system, bi-directional ME is required as discussed in Section I. Therefore, assume the search range is  $[-16a, 16a]$ , there are two search range buffers with size at least  $(16 \times (2a + 1))^2$  pixels available when processing MC. The basic idea of the proposed ME-based MC is to preload the search range of current MB into the on-chip search range buffer. The irregular off-chip memory access thus can be avoided.

The double current frame scheme is proposed to reuse the search range in ME-based MC as shown in Fig. 5. The loaded search range in high-band frame is shared by two reference frames. The off-chip memory access of this part thus can be

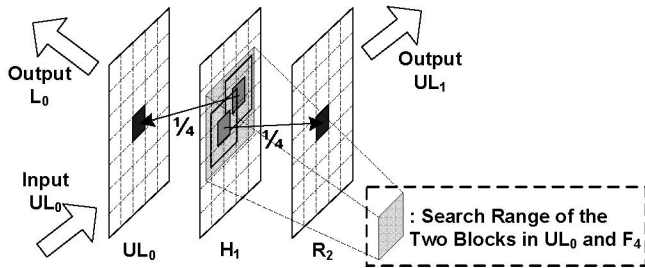


Fig. 5. An illustration of the proposed ME-based Level C+ MC scheme.

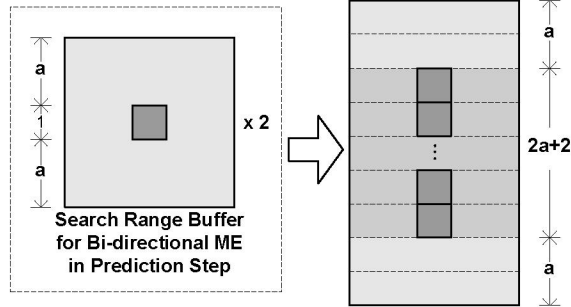


Fig. 6. The reuse scheme for the proposed ME-based level C+ MC. If the search range of one MB is  $[-16a, 16a]$ , the combined two search range buffer is sufficient for vertical data reuse of  $(2a + 1)$  MBs.

halved. The price paid is the reading and writing of the partial results of reference frame after left-side MC ( $UL_0$ ).

Since there are two search range buffers, the level C+ scheme reuse [5] of MC is possible. Figure 6 shows the vertical data reuse scheme for ME-based level C+ MC. Vertical data reuse of  $2a + 2$  MBs is achieved in this scheme. With the proposed scheme, each pixel has to be averagely loaded only  $1 + \frac{2a}{2a+2}$  times [5]. If search range is  $[-32, 32]$  ( $a = 2$ ), for a  $4 \times 4$  block, the off-chip memory access is  $[2(\text{load } UL_0, R_2) + 2(\text{output } L_0, UL_1) + (1 + \frac{4}{6})(\text{load } H_1)] \times (4 \times 4) \times 8 = 725.33\text{bits}$ . The off-chip memory access is reduced by 53%, and the irregular access is turned into regular access.

#### IV. THE PROPOSED VLSI HARDWARE ARCHITECTURE OF UPDATE STEP

Figure 7 shows the proposed hardware architecture of update step. The target frame size is CIF format ( $352 \times 288$ ), and the search range size is  $[-32, 32]$  in both vertical and horizontal direction.

When deriving inverse MVs, the MVs of high-band frame are inputted MB-by-MB. The motion information is stored in *Left Search Range Buffer* with level D data reuse scheme. After processing one MB row, the derived inverse MVs of the one MB row are outputted to off-chip memory for the use of MC.

The MC engine follows the ME-based level C+ MC scheme in Fig. 5. With this search range size, vertical data reuse of 6 MBs is utilized to reduce the off-chip memory access.

##### A. Hardware Reuse Scheme

Not all hardware resources are necessarily dedicated for update step. As discussed in Section II and III, the search range buffers needed in bi-directional ME can be reused for *Left Search Range Buffer* and *Right Search Range Buffer*.

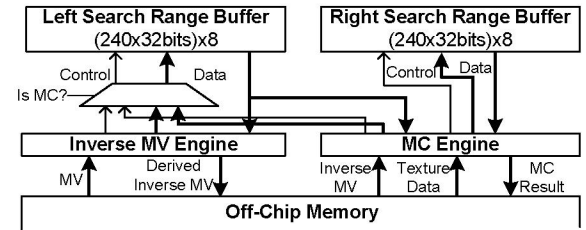


Fig. 7. The proposed hardware architecture of update step.

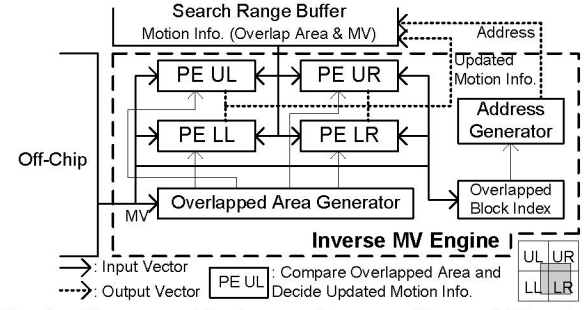


Fig. 8. The proposed hardware architecture of inverse MV engine.

The organization of search range buffer for ME is assumed to be as the work in [6], and the resulted search range is 8 banks of 240 ( $= 1920$ ) words in left or right direction with each word of 32 bits. If one 32-bit word is used to buffer motion information of one  $4 \times 4$  block. For level D data reuse of deriving inverse MVs,  $\lceil \frac{352}{4} \rceil \times \lceil \frac{31+16-(-32)}{4} \rceil = 1760$  words are required. Therefore, the size of search range buffer in one frame is sufficient for the level D data reuse.

The MC engine can reuse the MC module needed in fractional ME of prediction step. Therefore, the hardware resource dedicated for update step is only the *Inverse MV Engine*.

##### B. Hardware Architecture of Inverse MV Engine

Figure 8 shows the proposed hardware architecture of *Inverse MV Engine*. At each cycle, one  $4 \times 4$  block in high-band frame (current block) is processed. The MV of current block is loaded from off-chip memory. In *Overlapped Block Index*, the logical address of the overlapped four  $4 \times 4$  blocks in reference frame will be generated. The *Address Generator* maps the logical address into physical memory address and reads the motion information of the four overlapped blocks from the search range buffer.

The overlapped area of four overlapped blocks caused by current block is calculated in *Overlapped Area Generator*. Four PEs are utilized to compare the overlapped area caused by current block and that caused by previously processed blocks. The updated results of four overlapped blocks are then written back into the search range buffer.

##### C. On-Chip Memory Access Reduction

In Section II and III, two schemes are proposed to transfer the off-chip memory access into the on-chip memory access. This can greatly reduce the total system power and system bus traffic. A scheme shown in Fig. 9 is proposed to further reduce the on-chip memory access.

TABLE I

THE COMPARISONS OF OFF-CHIP MEMORY BANDWIDTH IN DIFFERENT LEVEL MCTF BETWEEN THE CONVENTIONAL AND THE PROPOSED UPDATE STEP SCHEME. BANDWIDTH VALUE IS SHOWN IN MBPS, AND THE CIF-SIZED FRAME IN 30FPS WITH SEARCH RANGE SIZE [32, 32] IS ASSUMED.

Access Type	Conventional			Proposed			Reduction		
	Inverse MV	MC	Total	Inverse MV	MC	Total	Inverse MV	MC	Total
	Irregular	Irregular		Regular	Regular				
1-Level MCTF	15.97	85.92	101.88	4.56	44.61	49.17	71.4%	48.1%	51.7%
2-Level MCTF	31.93	174.11	206.05	9.12	79.07	88.20	71.4%	54.6%	57.2%
3-Level MCTF	43.91	240.83	284.74	12.55	102.39	114.93	71.4%	57.5%	59.6%
4-Level MCTF	51.89	285.50	337.39	14.83	117.09	131.91	71.4%	59.0%	60.9%
5-Level MCTF	56.88	313.49	370.37	16.25	125.96	142.21	71.4%	59.8%	61.6%

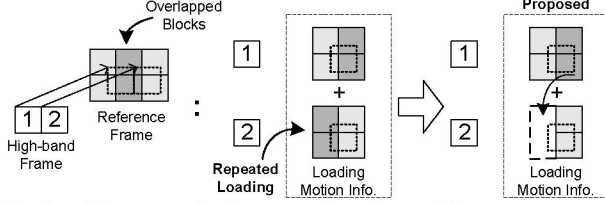


Fig. 9. The proposed scheme to reduce the on-chip memory access.

TABLE II

THE ON-CHIP MEMORY ACCESS OF DERIVING INVERSE MV WITH (B) AND WITHOUT (A) THE PROPOSED ON-CHIP ACCESS REDUCTION SCHEME (MBPS).

	foreman		stefan		coastguard		average		reduction
	A	B	A	B	A	B	A	B	
1-Level	1.08	0.75	1.53	1.16	1.88	1.41	1.49	1.11	25.9%
2-Level	2.43	1.66	1.73	1.42	2.47	1.90	2.21	1.66	25.1%
3-Level	2.77	1.96	2.06	1.66	2.93	2.24	2.58	1.95	24.5%
4-Level	2.72	2.07	2.69	2.03	3.00	2.31	2.80	2.14	23.7%

There are often common overlapped blocks between the last processed block and the current block due to the similarity of MVs. Therefore, if the updated motion information is buffered for one cycle, the loading of blocks which are the same with last processed block can be skipped.

## V. EXPERIMENTAL RESULTS

Table I shows the off-chip memory bandwidth of conventional scheme and the proposed scheme. The conventional update step scheme is the one discussed in Section II-A and III-A, and the proposed scheme is discussed in Section II-B and III-B. The proposed scheme can reduce about 55% off-chip memory bandwidth. Furthermore, the irregular memory access is turned into regular access. Generally, the regularity of off-chip memory access also has great influence on off-chip memory access power.

Table II shows the comparisons of on-chip memory access of deriving inverse MVs. About 25% on-chip memory access is reduced by the proposed scheme.

Table III shows the implementation results of the proposed hardware architecture. Artisan 0.18  $\mu\text{m}$  cell library is utilized, and the power is estimated by Synopsys PrimePower. By use of the hardware reuse scheme, the hardware dedicated to the updated step is of only 13259 logic gates. According to the implemental results in [7], the MC module itself occupies more than 60000 logic gates. This means about 80% area

TABLE III

THE SUMMARY OF THE IMPLEMENTED *Inverse MV Engine*.

Cell Library	Artisan 0.18 $\mu\text{m}$
Working Freq.	50MHz
Gate Count	13259
Estimated Power	6.7 mW (Fully Operation @ 50MHz)
	41 $\mu\text{W}$ (Scaled to Support CIF@30fps)

is saved. The circuits operates at 50MHz because of the consideration of MCTF system, and it can be turned off when MCTF system is in other steps. If 4-level MCTF is adopted, the power needed to enable CIF@30fps is only 41  $\mu\text{W}$ .

## VI. CONCLUSION

In this paper, a scheduling and hardware architecture of update step in MCTF are proposed as an efficient VLSI implementation. The proposed prediction/update pipeline scheme can reduce 71% off-chip memory access of deriving inverse MVs, and the proposed ME-based level C+ MC can reduce 53% off chip-memory access of MC. The proposed hardware reuse scheme can save about 80% area of update step. Only *Inverse MV Engine* is the hardware dedicated to update step. The hardware architecture is also proposed along with the on-chip memory reduction scheme that reduces 25% on-chip memory access of *Inverse MV Engine*.

## REFERENCES

- [1] Joint Scalable Video Model (JSVM) 2.0 Reference Encoding Algorithm Description, ISO/IEC JTC1/SG29/WG11 N7084, Oct. 2004.
- [2] Lin Luo, Feng Wu, Shipeng Li, and Zhenquan Zhang, "Advanced lifting-based motion-threading (MTh) technique for the 3d wavelet video coding," in *Proceedings of SPIE Visual Communications and Image Processing* 2003, 2003, pp. 707–718.
- [3] T. Nishikawa and et al., "A 60 MHz 240 mW MPEG-4 video-phone LSI with 16 Mb embedded DRAM," in *Digest of Technical Papers of IEEE International Solid-State Circuits Conference (ISSCC)*, 2000, pp. 230–231.
- [4] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search blockmatching VLSI architecture," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 1, pp. 61–72, JAN 2002.
- [5] 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 *Proceedings of 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 93–96.
- [6] Yu-Wen Huang and et al., "A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications," in *Digest of Technical Papers of IEEE International Solid-State Circuits Conference (ISSCC)*, 2005, pp. 128–130.
- [7] To-Wei Chen and et al., "Architecture design of H.264/AVC decoder with hybrid task pipelining for high definition videos," in *Proceedings of 2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 2931–2934.