A DATA-REUSE SCHEME FOR AVOIDING UNNECESSARY MEMORY ACCESSES IN MPEG-4 ASP VIDEO DECODER

Wei-Cheng Lin and Chung-Ho Chen

Department of Electrical Engineering and Institute of Computer and Communication Engineering
National Cheng-Kung University
No.1 University Road, Tainan, Taiwan 70101, R.O.C.
(kevin@casmail.ee.ncku.edu.tw and chchen@mail.ncku.edu.tw)

ABSTRACT

Due to the inherent characteristics of MPEG video decoder processing which involves bulk data transfers, a large portion of the energy used comes from memory accesses. In this paper, we propose a data-reuse scheme which recognizes the reusable data in frame buffers and a display RAM to avoid unnecessary memory accesses. Extensive experiments for eighteen video sequences exhibiting various image features show that the proposed approach on average reduces the frame buffer accesses by one half and the display RAM accesses by one quarter.

I. INTRODUCTION

The MPEG video decoding system decodes and displays the video streams. This system general consists of a decoder chip and a display unit which comprises an LCD/OLED panel and LCD/OLED driver with a built-in display RAM [1, 2], as shown in Fig. 1. The decoder employs two frame buffers to store anchor (I or P) frames for reference and display. The frame buffers and display RAM dominate total memory usage in the system due to high video compression rate. For example, in an MPEG-4 advance simple profile (ASP) video decoder [3], these two storage units are responsible for around 98.5% of the total memory accesses when the video compression rate is 25. Thus, reducing the number of frame buffer accesses and display RAM accesses is an attractive target for the power-aware video decoders.

In Fig.1, when a B-frame is decoded, the reconstructed macroblocks can be directly transformed into the RGB format and then written to the display RAM by specifying the appropriate row (column) start (end) address of "active window." When an anchor frame is decoded, the reconstructed macroblocks are written into the frame buffer which stores the displayed anchor frame, and the reference frame stored in the other

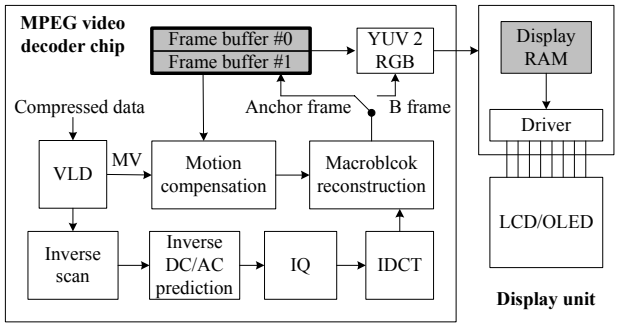

Fig. 1: An MPEG video decoder architecture with a display unit.

frame buffer is read out for both reference and data transformation (for display). Hence, the reconstructed (transformed) macroblock always overwrites a macroblock belonging to a previous frame in the frame buffer (display RAM). If the contents of these two macroblocks are totally equivalent due to co-located stationary macroblocks in consecutive frames, the macroblock in the memory is clearly reusable.

This work exploits the reusable data to reduce memory accesses. The proposed scheme is evaluated with eighteen benchmarks including sport, news, surveillance, street and scenes. On the average, this work has obtained about 45% saving in frame buffer accesses and 26% in display RAM accesses.

II. RELATED WORK

A number of researches explore reusable data at the algorithmic level to avoid repeated memory accesses for data-dominated operations such as the motion estimators [4, 5] or the deblocking filter [6]. To optimize memory hierarchy usage and reduce external memory accesses for processor-based implementation, several recent researches improve cache performance [7, 8] or use specialized on-chip software-controlled streaming memories [9].

978-1-4244-1593-9/07/$25.00 ©2007 IEEE

Several works [10, 11] try to reduce frame buffer accesses by compressing the image data prior to storing. These compression methods reduce frame buffer size, but degrade the resulting image quality. Bit-compression techniques for reducing the bit-line switching activity in the embedded frame buffer, based on the spatial correlation of the pixel in the video sequence, are reported in [12].

Moshnyaga et al. [13] aim to decrease computations for the B-frame in an MPEG-2 video decoder. They propose that current macroblock image data in the B-frame can be derived directly from the co-located macroblock, which is stationary and exists in the future reference frame, without performing decoding. However, reusable macroblocks in the memory may still exist even if the future reference frame macroblock is not stationary. Our approach differs from [13], in that it uses the reference relationship between the reconstructed (or transformed) macroblock and the macroblock to be overwritten in the memory to identify reusable data. Unnecessary accesses can thus be reduced as much as possible for both the P- and B-frames.

III. METHODOLOGY

A. Stationary Macroblock

In the MPEG-4 ASP, the I-frame is decoded independently without reference to other frames. The P-frame is decoded by taking reference frame information from the last decoded anchor frame while the B-frame is based on the previous and future reference frames (i.e., the last two decoded anchor frames). The B-frame uses backward prediction; therefore the future reference frame must be decoded prior to B-frame decoding. The decoding order of frames is thus different from the display order as illustrated in Fig. 2, where the arrows indicate the prediction directions and anchor frames are in bold.

Each coded P-frame macroblock is either in Inter or Intra mode. An Intra-macroblock in the P-frame is also decoded independently without reference to other video frames. Decoding an Inter-macroblock implies the addition of a reference macroblock with a residual macroblock. The reference macroblock is extracted from the reference frame according to the motion vector(s) while the residual macroblock is derived from texture decoding. A macroblock in the B-frame is decoded in the Inter mode. Its reference macroblock(s) are obtained from the previous reference frame in the forward prediction mode, the future reference frame in the backward prediction

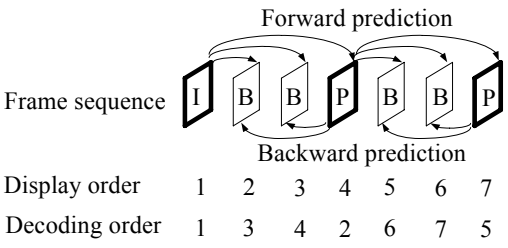

Fig. 2: Display and decoding orders.

mode, or the average of two reference regions in the previous and future reference frames in the bi-direction prediction mode.

The decoded macroblock is stationary if it exactly equals its reference macroblock. In the P-frame, if the residual macroblock and 16×16 motion vector of an Inter-macroblock are all zero, this macroblock (which is skipped and therefore not coded) is stationary and referred to as the *forward stationary macroblock*. For the B-frame, the stationary macroblock only exists in the forward and backward prediction modes. If the residual macroblock and the forward (backward) 16×16 motion vector of a macroblock in the B-frame are all zero, this macroblock is exactly the same as its previous (future) reference macroblock and referred to as the *forward (backward) stationary macroblock*. Hence, each decoded macroblock is classified as non-stationary, forward-stationary, or backward-stationary. Clearly, the macroblock in Intra mode is non-stationary. Note that if a macroblock in the future reference frame is skipped, the co-located macroblocks in the B-frames are also skipped (forward-stationary) in MPEG-4 video coding.

The occurrence of stationary macroblocks not only depends on the inherent video sequence, but also on the encoder-defined quantization parameter (QP). Using a larger QP to encode a video sequence will typically result in lower bit-rate, poorer image quality, but more stationary macroblocks.

B. Detection of Reusable Macroblocks

Based on the MPEG video decoder architecture described in Fig. 1, frame buffers store two anchor frames. Consequently, the new reconstructed macroblock in an anchor frame always overwrites a macroblock (referred to as the victim macroblock) in a previous anchor frame (referred to as the victim frame). The victim macroblock in the frame buffer is completely reusable when these two macroblocks are exactly the same. Similarly, reusable data also exist in the display RAM for B-frame decoding and anchor frame displaying.

978-1-4244-1593-9/07/$25.00 ©2007 IEEE

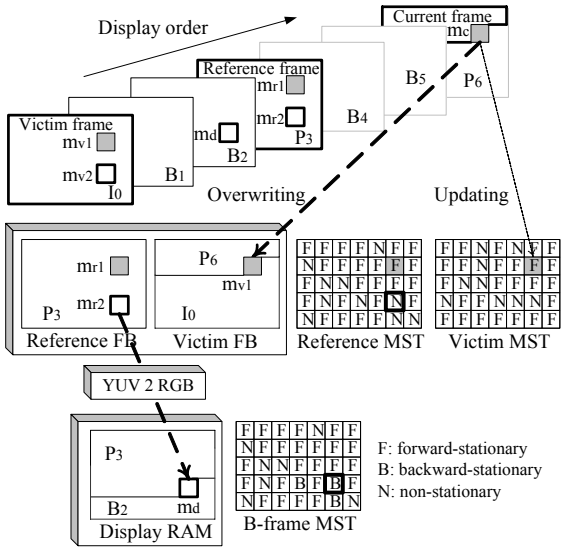

Fig. 3: Detection of reusable macroblocks.

Fig. 3 gives an example illustrating how to identify the reusable macroblock. The four frames (I_0, B_1, B_2 and P_3) and the upper part of the current frame P_6 are assumed to be completely decoded. The frame buffers store the victim frames I_0 and the reference frame P_3. The roles of the victim frame buffer and the reference frame buffer are exchanged at the end of each anchor frame decoding. The current macroblock m_c will overwrite the victim macroblock m_{v1}. If m_c and the co-located macroblock m_{r1} in the reference frame P_3 are both forward-stationary ($m_c = m_{r1} = m_{v1}$), then m_{v1} in the buffer is reusable; thus reading m_{r1} (for reconstructing m_c) and writing m_c from/to frame buffers are both redundant.

While an anchor frame is decoding, the reference frame in the frame buffers is transformed into the RGB format and then written to the display RAM. In Fig. 3, we assume that the macroblock m_{r2} in P_3 is read for data transformation and the RGB format macroblock m_d of frame B_2 in the display RAM is going to be overwritten by the transformed data. If m_d is backward-stationary ($m_d = m_{r2}$) or m_{r2} is forward-stationary (i.e., also m_d is forward-stationary, $m_d = m_{v2} = m_{r2}$), then m_d in the display RAM is reusable; thus reading m_{r2} from the frame buffers and writing transformed m_{r2} into the display RAM are both redundant.

Then, we assume that frame P_6 is completely decoded and a macroblock in frame B_4 is going to be decoded. If this macroblock is forward-stationary, the co-located macroblock in P_3 stored in the display RAM is reusable. After the frame B_4 is completely decoded and stored in the display RAM, a macroblock in frame B_5 is going to be decoded. If this macroblock and the co-located macroblock in the last decoded B-frame B_4 are both forward-stationary or both backward-stationary, the latter stored in the display RAM is reusable. The method of detecting reusable macroblocks is summarized using two flow charts as shown in Fig. 4.

This work employs two macroblock state tables (reference MST, victim MST) to keep track of the "stationary-information" of macroblocks stored in the frame buffers. The B-frame MST is used to record the states of macroblocks in the last decoded B-frame. An MST has as many cells as the number of macroblocks in a frame; each cell in the reference and victim MSTs contains two states (forward- and non-stationary) and the cell in the B-frame MST contains three states (forward-, backward- and non-stationary). The hardware cost of these three MSTs is 198 bytes for CIF (352×288) image resolution and is less then 0.07% of that of the frame buffers. Hence, its power consumption is less considerable. Accessing the MST consumes additional bus bandwidth; this overhead is negligible because only five bits of MST data at most are transferred for a macroblock decoding.

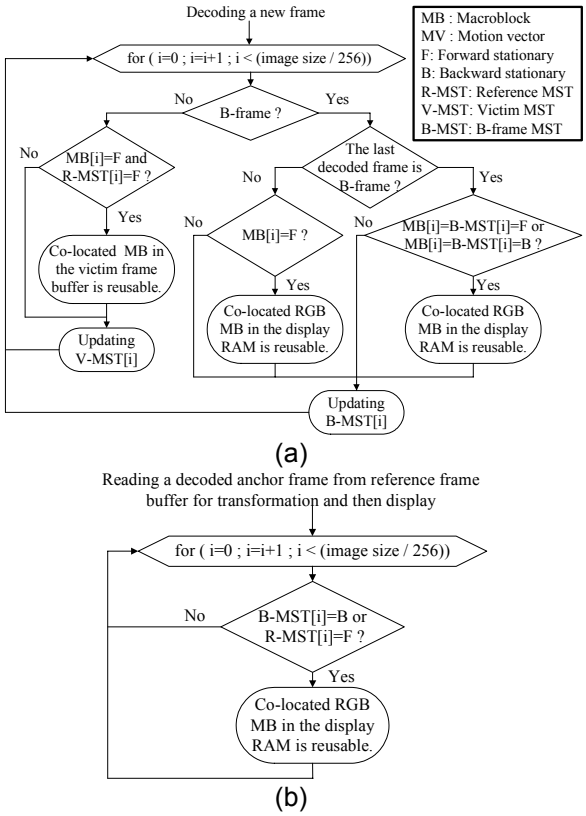

Fig. 4: The flow charts of detecting reusable macroblocks for (a) frame decoding and (b) anchor frame displaying.

978-1-4244-1593-9/07/$25.00 ©2007 IEEE

Table 1: The percentage of reduced accesses.

Class	QP value	% of reduced frame memory acceses					% of reduced display RAM acceses				
		4	8	16	24	28	4	8	16	24	28
I	Mother & daughter	29.8	51.1	61.9	65.4	65.9	17.0	30.1	37.0	39.0	39.3
	Akiyo	61.1	70.4	76.6	78.2	78.4	34.0	39.5	42.9	43.9	44.1
	Hall monitor	0	25.4	69.3	76.0	76.3	0	15.3	38.3	41.7	41.9
	Container ship	17.5	47.0	68.1	74.7	76.0	11.6	30.7	40.0	42.6	43.4
	Sean	56.1	69.4	75.3	77.0	77.1	31.5	38.8	42.1	43.2	43.3
	Bridge_far	0.7	57.8	87.1	87.9	86.8	0.5	32.9	47.1	47.3	46.9
II	Foreman	1.7	8.6	19.0	23.1	24.6	1.2	6.4	13.4	16.2	17.2
	News	44.0	58.7	67.2	69.9	70.5	25.2	34.0	38.8	40.2	40.5
	Silent voice	36.4	53.7	61.4	63.3	63.6	21.5	30.7	35.4	36.5	36.7
	Coastguard	0	0.9	8.7	16.6	19.7	0	0.7	6.8	12.3	14.2
	Weather	64.2	67.2	70.6	72.1	72.7	34.8	36.9	39.2	40.4	40.8
	Weather_foreground	64.4	67.3	70.9	72.9	73.5	35.0	37.1	39.4	40.7	41.1
III	Bus	0	0.8	2.4	5.0	6.1	0	0.6	1.9	3.8	4.6
	Stefan	3.6	10.4	14.3	16.0	16.6	2.8	7.0	9.2	10.4	10.8
	Mobil & calendar	0.4	1.3	3.0	5.5	6.9	0.3	0.9	2.1	4.0	5.1
	Football	0.1	2.4	12.5	20.6	23.1	0.1	1.9	9.1	13.9	15.0
	Table tennis	6.9	18.8	37.1	43.9	45.4	4.1	11.3	22.8	26.4	27.2
	Cheer leaders	1.0	6.3	12.0	15.3	16.4	0.7	4.2	7.6	9.7	10.4
Average		21.5	34.3	45.4	49.1	50.0	12.2	19.9	26.3	28.5	29.0

IV. SIMULATION RESULTS

This work uses an MPEG-4 advance simple profile at level 3 video coding/decoding environment to evaluate the performance of the proposed approach. The three-step-search method with search range [-16, +15] is adopted in this paper and the I-frame period has 30 frames. The encoder first generates bit-streams, running on video sequences by setting the frame pattern in IBBPBBP. The decoder then decodes bit-streams with the detection of reusable macroblocks.

Experiments run on eighteen video sequences classified into three classes [3] according to spatial frequency and amount of movement, as shown in Table 1. The frame size of *Football*, *Sean*, *Weather*, *Weather_foreground,* and *Table tennis* is 352×240 pixels, while the remainder is 352×288 pixels. Performance comparisons are done on the first 150 frames of *Bus* and *Cheerleaders*, 120 frames of *Football*, and 300 frames of the others.

Table 1 shows the percentage of reduced frame buffer access and display RAM accesses with different QP values. We assume that the decoding rate of the decoder is equal to the display rate, i.e., the number of reads and writes from/to the display RAM are equivalent. The reduced memory accesses increase with the increase of the QP value because a higher QP value results in more zero-residuals. When the QP value is equal to sixteen, about 73.1%, 49.6%, and 13.6% (41.2%, 28.8%, and 8.8%) frame buffer (display RAM) accesses can be saved for Class I, II, and III video sequences, respectively. Class I video sequences have much lower spatial frequency and movement, leading to larger amounts of zero-residuals and zero-motion vectors. Thus, the reduction in frame buffer accesses is the largest. Our approach is especially effective for low motion videos as a result. On the average, 45.4% of frame buffer accesses and 26.3% of display RAM accesses can be saved when the QP value equals to sixteen.

V. CONCLUSION

This paper presents a data-reuse scheme to reduce redundant memory accesses for the MPEG-4 ASP video decoder. Simulation results have shown that the approach can eliminate around one half of the frame memory accesses as well as over one quarter of the display RAM accesses. Through the experiments, we found that the inherent video features have made the proposed scheme quite attractive to use in power-aware video decoders for embedded applications.

ACKNOWLEDGEMENT

The work in this paper is in part supported by the National Science Council, Taiwan, under NSC 94-2220-E-006-004.

REFERENCES

[1] SSD1339, 132RGB x 132 with 2 smart Icon lines Dot Matrix OLED/PLED Segment/Common Driver with Controller, April 2005, http://www.solomon-systech.com.

[2] PCF8881, 132 × 132 RGB single chip TFT driver with DC-to-DC converter, August 2004, http://www.semiconductors.philips.com.

[3] ISO/IEC JTC1/SC29/WG11, "MPEG4 Video Verification Model Version 18.0," January 2001.

[4] 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, January 2002.

[5] Y.-K. Lai, Y.-L. Lai, Y.-C. Liu, P.-C. Wu, and L.-G. Chen, "VLSI Implementation of the Motion Estimator with Two-Dimensional Data-Reuse," *IEEE Transactions on Consumer Electronics,* Vol. 44, No.3, pp. 623-629, August 1998.

[6] C.-M. Chen and C.-H. Chen, "An Efficient Pipeline Architecture for Deblocking Filter in H.264/AVC," *IEICE Transactions on Information and Systems*, Vol. E90-D, No.1 pp.99-107, January 2007.

[7] Z. Xu, S. Sohoni, R. Min, and Y. Hu, "An Analysis of Cache Performance of Multimedia Applications," *IEEE Transactions on Computers,* Vol. 53, No. 1, pp. 20-38, January 2004.

[8] P. Pakdeepaiboonpol and S. Kittitornkun, "Energy Optimization for Mobile MPEG-4 Video Decoder," *International Conference on Mobile Technology, Applications and Systems*, pp. 1-6, November 2005.

[9] A. Ramachandran and M.F. Jacome, "Energy-Delay Efficient Data Memory Subsystems," *IEEE Signal Processing Magazine*, pp. 23-37, May 2005.

[10] C.-W. J. Shih, N. Ling, and T. Ogunfunmi, "Memory Reduction by Haar Wavelet Transform for MPEG Decoder," *IEEE Transactions on Consumer Electronics*, Vol. 45, No. 3, pp. 867-873, August 1999.

[11] P. H. N. de With, P. H. Frencken, and M. V. D. Schaar-Mittera, "An MPEG Decoder with Embedded Compression for Memory Reduction," *IEEE Transactions on Consumer Electronics*, Vol. 44, No. 3, pp. 545-555, August 1998.

[12] V. G. Moshnyaga, "Reducing Energy Dissipation of Frame Memory by Adaptive Bit-Width Compression," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 8, pp. 713-718, August 2002.

[13] V. Moshnyaga and K. Wakisaka, "Reducing Computations in MPEG2 Video Decoder," *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, Island of Kos, Greece, pp. 1820-1823, May 2006.

978-1-4244-1593-9/07/$25.00 ©2007 IEEE