Avoiding Unnecessary Frame Memory Access and Multi-Frame Motion Estimation Computation in H.264/AVC

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 chchen@mail.ncku.edu.tw

***Abstract*—H.264/AVC video compression standard uses multiple reference frame motion estimation (MRF-ME) to enhance the coding performance. The required frame memory accesses and the computational cost of the MRF-ME, however, are very high. This paper proposes an approach which exploits the characteristic of a constant luminance macroblock (L-MB) to avoid unnecessary frame memory accesses and MRF-ME computation. Simulation results show that the proposed scheme can reduce about 43% of the frame memory accesses and 33% of the MRF-ME computation for low motion video sequences without any PSNR degradation and bit-rate increase.**

I. Introduction

H.264/AVC is currently the most efficient and state-of-the-art video compression standard, jointly developed by the Video Coding Experts Group (VCEG) of the ITU-T and the Moving Picture Experts Group (MPEG) of ISO/IEC. Compared to previous standards, it can achieve 50% coding gain over MPEG-2 and 24% coding gain over H.263 high profile encoders while keeping the same video quality. In order to provide such high coding performance, H.264/AVC adopts a number of relatively new technical advances such as MRF-ME, variable block-size motion compensation, quarter-sample accuracy motion compensation, in-loop deblocking filtering, and so on.

In MRF-ME, more than one prior-coded picture can be used as a reference. Thus, ME, the most computationally intensive task in the encoding process, is performed on each of the reference frames when encoding an inter-frame. This means more candidate L-MBs should be loaded from the external frame memory and compared. These high computation and memory access demands consequently induce large power consumption for the encoder.

Many methods have been proposed to reduce the computation [1-11] and the memory access [12, 13] for the MRF-ME. In [1-4], the motion vectors between two successive reference frames are used to predict the optimal motion vectors between the current frame and the reference frames (second to the oldest). These approaches effectively simplify the complexity of the MRF-ME but are more suitable for software-oriented implementations. Several researches [5-7] use a threshold condition to determine whether to terminate the ME process or to skip remaining reference frames early so the computational cost can be reduced. Ting et al. [8] propose a center-based frame selection algorithm which only searches on a selected path in each reference frame to determine a final frame for full search. Chang et al. [9] describe a fast multi-frame selection approach to increase ME speed. In this approach, they only enable one reference frame selected from a reference frame group, the center MBs of which have the same sub-pixel location, for ME. In [10], Liang et al. exploit the motion intensity from a video sequence at block, frame, and sequence levels to control different fast ME techniques. Li et al. [11] take advantage of the Lagrange function costs of the neighboring MBs to determine the best reference frame. These methods make the MRF-ME faster, however, at the cost of degraded video quality.

Chen et al. [12] propose a novel scheme that reuses once-fetched search window data to perform ME for multiple current MBs, existing in consecutive original-frames, by rescheduling the MRF-ME procedures at frame level. This method can reduce on-chip search window buffer size and system bandwidth, but increase the off-chip original-frame buffer size and sacrifice coding efficiency due to suboptimal ME. In [13], Shin and Kyung apply the level-C date reuse scheme [14] on the MRF-ME and modify the reference frame search order to reduce memory access.

In this paper, our experiments show that encoding low activity videos, which contain highly correlated images and large static video objects, results in many constant L-MBs. In MRF-ME, when these constant L-MBs exist in reference frames, repeatedly transferring the same image data from the external frame memory to on-chip local buffers and performing ME on a search window that is full of constant L-MBs are both redundant and should be avoided. The simulation results show that for low motion video sequences, the frame memory access and ME computation can be reduced by as much as 43% and 33%, respectively, without any sacrifice in image quality and coding efficiency.

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

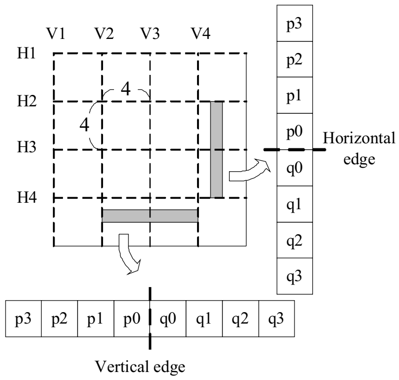

Fig. 1. Edges to be filtered in an L-MB and pixels involved in the in-loop deblocking filter operation.

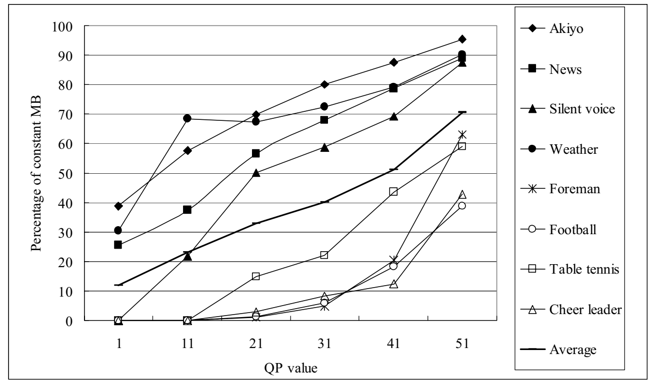

Fig. 2. Percentage of constant L-MBs versus QP value.

The remaining of the paper is organized as follows. Section II introduces the constant L-MB and presents the statistic analysis. Section III describes the proposed algorithm. Simulation results are discussed in Section IV. Finally, Section V concludes this paper.

II. CONSTANT MB

A constant L-MB should satisfy the following three conditions: (1) this MB has zero-valued motion vector and zero-valued luminance residual, (2) the reference frame chosen is the nearest reference frame, and (3) the in-loop deblocking filter does not alter the luminance component of this MB. This work only focuses on the constant L-MB which refers to the first reference frame since (1) our experimental results indicate that over 95% of the MBs satisfying the first and third conditions refer to the first reference frame, and (2) the hardware overhead of the table used to record the states of MBs can be minimized. Furthermore, an L-MB specified in the H.264 may be partitioned into sub-blocks of different sizes (such as 8×16, 16×8, 8×8, 4×8, 8×4, and 4×4). Based on the results of our experiments, a sub-block being a constant block rarely appears. Thus, we only focus on the 16×16 constant L-MB.

In H.264/AVC, the deblocking filter is performed to reduce the blocking artifacts after the inverse transformation. Fig. 1 illustrates the case that the filter is applied to the vertical and horizontal edges of the 4×4 luminance blocks in an MB. The inputs of a filtering operation include the adjacent p and q blocks (pixel p0, p1, p2, p3, q0, q1, q2 and q3) across a vertical or horizontal edge. The filter may alter the values of the six pixels (p0, p1, p2, q0, q1, and q2).

A parameter called boundary strength (BS) which decides the filtering strength is assigned to each edge with an integer value from one to four. A zero BS value means that no filtering is required for this specific edge. The BS value is set to zero if the following conditions are all satisfied: p and q are both inter coded; p and q do not contain coded coefficients; p and q have the same reference frame; p and q have motion vectors that differ less than one luminance pixel.

Whether the samples p0 and q0 as well as p1 and q1 are filtered or not is determined by the BS value, dependent threshold α and β, and the content of the picture itself. The filtering of p0 and q0 only takes place if the following content activity check operations are satisfied:

$$BS \neq 0 \quad \text{---Eq. (1)}$$
$$|p0-q0| < \alpha \quad \text{---Eq. (2)}$$
$$|p1-p0| < \beta \text{ and } |q1-q0| < \beta \quad \text{--- Eq.(3)}$$

where α and β depend on the average quantization parameter (QP) of the two blocks p and q; they increase with the increase of the QP and vice versa. Correspondingly, the filtering of p1 or q1 takes place if the condition below is satisfied:

$$|p2-p0| < \beta \text{ or } |q2-q0| < \beta \quad \text{--- Eq.(4)}$$

The occurrence of constant L-MBs not only depends on the inherent feature of a video sequence, but also on the QP. Using a larger QP to encode a video sequence will typically result in lower bit-rate, poorer image quality, but more constant L-MBs. Fig. 2 shows the percentage of constant L-MBs and the impact of the QP on it for eight video sequences. The results are obtained from running H.264 JM 8.5 [15] with four reference frames. On average, about 40% of the L-MBs are constant when QP is equal to 31.

A higher QP generally results in more constant L-MBs despite the adverse effect of more intense deblocking operations. To explain this, Fig. 3 presents the variation of constant L-MB distribution (denoted by the black dots) in *Akiyo's* 12[th] frame for different QPs (16 and 31). The gray blocks including those with black dots represent the L-MBs satisfying the first and second conditions. When the QP is increased, the number of the gray blocks increases because more residuals are quantized to zero. Thus, more filtering operations among the gray block edges are turned off due to the zero-BS, which makes more gray blocks satisfy the third condition.

On the other hand, a larger QP implies that the blocking artifacts may become more significant; therefore, α and β are increased and potentially more edges with a non-zero-BS are filtered. Because the larger α and β mainly influence the edges around and within the 'white block regions,' the gray blocks surrounding the white block regions are less likely to become

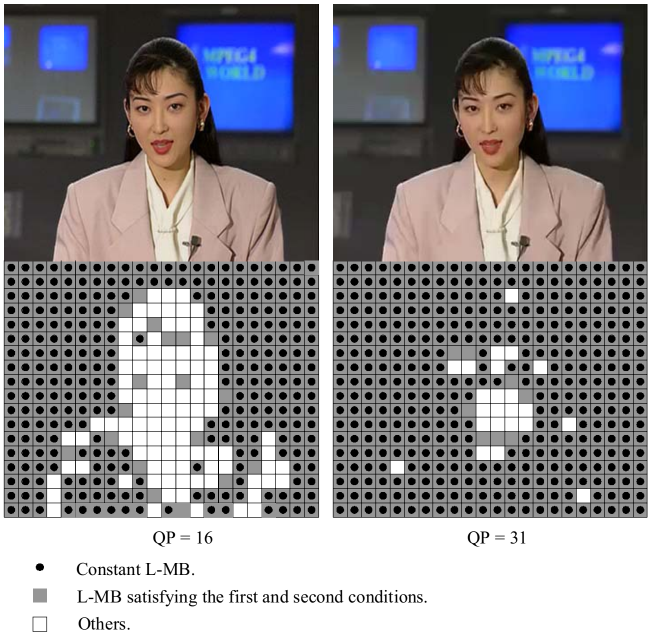

Fig. 3. Distribution of constant L-MBs as QP increases.

a constant L-MB. The white block regions, however, shrink with the increase of QP so the number of the affected neighboring gray blocks decreases. Thus, generally, as the QP becomes larger, the increased constant L-MBs outnumber those gray blocks adversely affected by the in-loop filtering operations.

III. Proposed Algorithm

The frame memory in a large frame application is much too large to fit into an embedded memory, and therefore is typically located in the external memory. For ME, the luminance pixels of a reference frame located within a search range should be loaded from the external memory. These bulk data transfers may cause an I/O bound problem and consume excessive power. To overcome this difficulty, a local buffer is usually employed to hold reusable data located within overlapped search area between two adjacent MBs in the same horizontal row, as shown in Fig.4. When ME is performed on the current MB, only the search range data in B are loaded into the local buffer and overwrite the data in A.

In MRF-ME, multiple local buffers are used to store search range data in different reference frames, as shown in Fig. 5. In this figure, we assume that the number of reference frames is four and the search range is −16 to 15 pixels (nine reference MBs). The current frame is encoded at time slot t whereas the reference frame-1, -2, -3 and -4 are reconstructed at time slot t-1, t-2, t-3 and t-4, respectively. Because two sequential search windows in one reference frame share six reference MBs, only three right L-MBs of each search window are loaded into a local buffer. Some of these data transfers are redundant if the L-MB to be transferred in the reference frame-1, -2 or -3 is constant. For example, in Fig. 5, the four top right L-MBs in the search windows are exactly the same. Thus, only the first L-MB is transferred from the external memory to the first local buffer and broadcast to the other

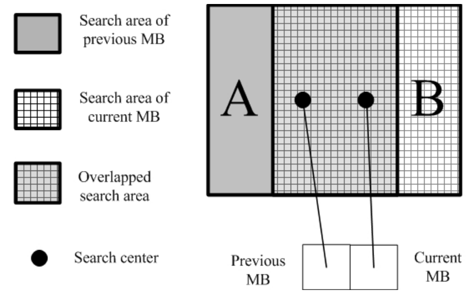

Fig. 4. Search area overlapping.

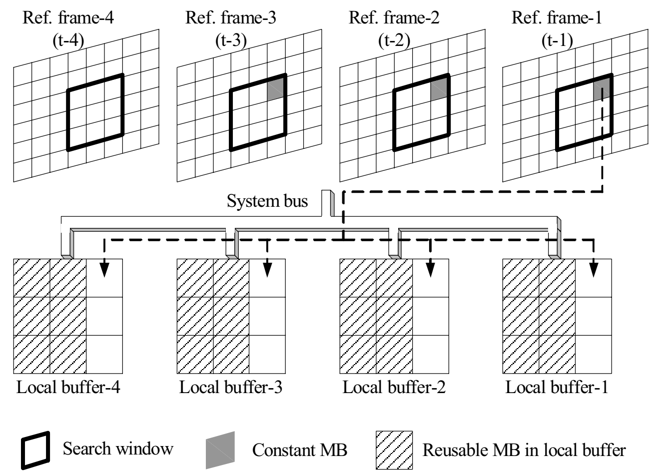

Fig. 5. Broadcasting the constant MB to avoid unnecessary data transfer.

three local buffers. On the other hand, if the L-MBs stored in a local buffer are all constant, performing ME with same search window data on the next reference frame is redundant since the ME results are the same.

The method of avoiding unnecessary frame memory access is summarized by using a flow chart shown in Fig. 6. The outer loop checks whether the r co-located MBs are loaded into the local buffers, whereas the inner FOR loop decides which local buffers to receive the broadcast L-MB. This work employs an MB state bitmap (MSB) to keep track of the status of L-MBs in each reference frame. An MSB has as many entries as the number of MBs in a frame; each entry contains two states (non-constant and constant). The hardware cost of one MSB is 49.5 bytes for CIF (352×288) image resolution and is less than 2.2% of that of a local buffer holding nine L-MBs. Hence, its power consumption is negligible.

IV. Simulation Results

The proposed algorithm is implemented in H.264 encoder reference software JM 8.5 [15]. The motion estimation uses full search with a search range of [−16, 15] and the number of reference frames is four. The frame pattern is IPPP… structure while the Intra-period is 30 frames. The test video sequences are classified into two groups: low motion group (*Akiyo*, *News*, *Silent voice* and *Weather*) and fast motion group (*Foreman*, *Football*, *Table tennis* and *Cheerleader*). The frame size of *Football*, *Weather* and *Table tennis* is 352×240 pixels whereas the rest are 352×288 pixels.

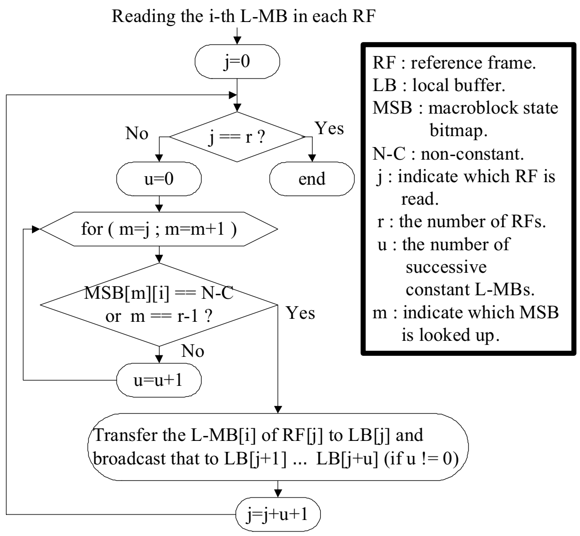

Fig. 6. Flow chart of the proposed approach.

Performance comparisons are done on the first 150 frames of *Cheerleaders*, 120 frames of *Football*, and 300 frames of the others.

Table I shows the percentage of reduced frame memory accesses and MRF-ME computation in three levels of video quality, high, normal, and low where the peak signal-to-noise ration (PSNR) of luminance is around 40, 34 and 28 dB, respectively. The higher the video quality is, the less the access and computation are saved. This is because smaller QPs, decided by rate control, are used in the encoder. However, in the low motion video sequences, over 43% of the frame memory accesses are saved even at the high video quality level. This is because encoding a low motion video (taken by a still camera with almost inactive background) results in many constant L-MBs. On average, 26.8% of frame buffer accesses and 21.8% of MRF-ME computation can be saved at the normal video quality level.

V. CONCLUSION

In this paper, we propose a constant L-MB detector to remove redundant frame memory access caused by transferring the same L-MB multiple times and avoid unnecessary MRF-ME performed on the same search range data. Simulations of the proposed method show that the approach can eliminate around 43% of the frame memory accesses as well as 33% of the MRF-ME computations for low motion videos at the normal video quality level, without affecting the image quality. Through the experiments, we found that the inherent video features have made the proposed scheme quite attractive to use in power-aware H.264/AVC video encoders.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan under NSC-96-2221-E-006-192-MY3 and by the Program for Promoting Academic Excellence of Universities in Taiwan.

Table I. Percentage of reduced frame memory accesses and MRF-ME computation in different bit rates.

Benchmark	Akiyo	News	Silent voice	Weather	Foreman	Football	Table tennis	Cheer leader	Average
PSNR (dB)	40.34	39.50	40.23	40.16	40.08	39.91	40.05	39.71	39.99
Bitrate (kB/s)	16.76	40.40	91.12	65.00	148.72	547.92	299.88	642.62	231.55
% of RFMA	50.6	41.6	34.5	45.6	0.9	0.8	9.9	2.4	23.3
% of RMMC	41.7	29.0	21.8	43.0	0.1	0.1	4.3	0.2	17.5
PSNR (dB)	34.32	33.42	34.01	34.21	34.30	34.14	34.20	34.26	34.11
Bitrate (kB/s)	5.13	14.28	28.20	32.64	41.48	272.34	121.39	361.15	109.58
% of RFMA	56.3	48.0	39.0	48.2	3.3	2.1	13.3	4.4	26.8
% of RMMC	52.4	37.4	29.3	44.9	0.6	0.3	8.8	0.5	21.8
PSNR (dB)	28.35	28.26	28.43	28.52	27.96	27.60	27.78	28.10	28.13
Bitrate (kB/s)	2.01	5.94	7.50	15.63	10.75	81.54	26.72	152.31	37.80
% of RFMA	61.3	53.7	46.0	51.2	15.5	7.7	25.9	6.9	33.5
% of RMMC	60.8	47.3	39.0	47.2	6.7	2.1	18.8	1.3	28.0

RFMA: Reduced frame memory access   RMMC: Reduced MRF-ME computation

References

[1] S.-E. Kim, J.-K. Han, and J.-G. Kim, "An Efficient Scheme for Motion Estimation Using Multireference Frames in H.264/AVC," *IEEE Transactions on Multimedia*, Vol. 8, No. 3, pp. 457-466, June 2006.

[2] M.-J. Chen, G.-L. Li, Y.-Y. Chiang, and C.-T. Hsu, "Fast Multiframe Motion Estimation Algorithms by Motion Vector Composition for the MPEG-4/AVC/H.264 Standard." *IEEE Transactions on Multimedia*, Vol. 8, No. 3, pp. 478-487, June 2006.

[3] G. N. Rao and P. S. S. B. K. Gupta, "Temporal Motion Prediction for Fast Motion Estimation in Multiple Reference Frames," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 817-820, August 27-30, 2006.

[4] Y. Su and M.T. Sun, "Fast Multiple Reference Frame Motion Estimation for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 447-452, March 2006.

[5] A. K. Mahajan, S. Kondayya, and X. Su, "Exploiting Reference Frame History in H.264/AVC Motion Estimation," *IEEE Asia Pacific Conference on Circuits and Systems*, pp. 410-413, December 4-7, 2006.

[6] S.-F. Lin, M.-T. Lu, H. Chen, and C.-H. Pan, "Fast Multi-Frame Motion Estimation for H.264 and Its Applications to Complexity-Aware Streaming," *IEEE International Symposium on Circuits and Systems*, pp. 1505-1508, May 23-26, 2005.

[7] Y.-W. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 4, pp. 507-522, April 2006.

[8] C.-W. Ting, L.-M. Po, and C.-H. Cheung, "Center-Biased Frame Selection Algorithms for Fast Multi-Frame Motion Estimation in H.264," *IEEE International Conference on Neural Networks and Signal Processing*, pp. 1258-1261, December, 2003.

[9] A. Chang, O. C. Au, and Y.-M. Yeung, "A Novel Approach to Fast Multi-Frame Selection for H.264 Video Coding," *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 704-707, May 25-28, 2003.

[10] Y. Liang, I. Ahmad, J. Luo, Y. Sun, and V. Swaminathan, "On Using Hierarchical Motion History for Motion Estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 12, pp. 1594-1603, December 2005.

[11] H.-J. Li, C.-T. Hsu, and M.-J. Chen, "Fast Multiple Reference Frame Selection Method for Motion Estimation in JVT/H.264," *IEEE Asia-Pacific Conference on Circuit and System*, pp. 605-608, December 6-9, 2004.

[12] T.-C. Chen, C.-Y. Tsai, Y.-W. Huang, and L.-G. Chen, "Single Reference Frame Multiple Current Macroblocks Scheme for Multiple Reference Frame Motion Estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, No. 2, pp. 242-247, February 2007.

[13] H. Shim and C.-M. Kyung, "Data Reuse Algorithm for Multiple Reference Frame Motion Estimation," *Electronics Letters*, Vol. 43, No. 7, pp. 382-383, March 2007.

[14] 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.

[15] Joint Video Team Reference Software JM8.5, September 2004