Fast Motion Estimation Using Hierarchical Motion Intensity Structure

Yongfang Liang, Ishfaq Ahmad, Jiancong Luo, Yu Sun
Department of Computer Science and Engineering
The University of Texas at Arlington, TX 76019
{yliang, iahmad, jluo, sunyu}@cse.uta.edu

Abstract

The embedded motion compensation model of the new H.264/AVC video-coding standard dramatically increases the computational complexity of motion estimation. In this paper, we propose a fast motion estimation algorithm using hierarchical motion intensity structure to lower the computational complexity of the motion estimation in H.264/AVC. The proposed algorithm is mainly based on a multi-level motion intensity structure. It determines the motion intensity at three levels and accordingly uses different motion estimation techniques to find more accurate motion vector (MV) with faster speed. Experimental results show that the proposed algorithm provides promising performance in terms of the computational speedup and video reconstruction quality.

1. Introduction

H.264, also known as MPEG-4 AVC (Advance Video Coding), is a new video coding standard [2] in response to the growing need for higher compression required by the new applications, including high definition movies, mobile videophones and digital cinema. Due to its flexibility and clean slate design, H.264/AVC is much more complex than the prior standards, which limits its practical application. From the motion estimation/compensation side, the feature of multi-reference frames selection and variable block sizes increases the computational complexity of motion estimation (ME) dramatically.

In H.264/AVC, a much larger number of different block sizes and shapes (7 segmentations: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4) within one macroblock (MB) are available for performing ME. Moreover, H.264/AVC offers the option of having up to 16 reference frames in inter-picture coding. To choose the coding block size and reference frame for the best coding efficiency, each block size needs to be investigated in every reference frame. This huge computational complexity leads to a new research challenge in finding an effective fast ME algorithm, which is suitable for the new motion compensation model in H.264/AVC.

Many fast ME algorithms have been proposed in the literature to reduce the complexity of ME, such as the heuristic search [4][5][6][10] and the hierarchical search [1][7]. These techniques make the search faster at the expense of less motion accuracy and video quality. Recently, new ME algorithms using the spatial-temporal correlations were presented, such as [3][9], to achieve better video quality. However, most of these algorithms can get trapped into local minima. Moreover, these techniques are not designed for H.264/AVC, without consideration of the characteristics of its new motion compensation model.

This paper proposes a fast ME algorithm using hierarchical motion intensity structure (FMEHMI) to reduce the computational complexity of ME for H.264/AVC, while keeping the video quality close to the full search algorithm (FS). A hierarchical structure is employed to determine the motion intensity (MI) at three levels. By applying different techniques at different level accordingly, the speedup and accuracy performance is improved. The novel hierarchical motion intensity structure and the techniques designed specially for H.264/AVC distinguish our algorithm from existing ME algorithms.

This paper is organized as follows. The hierarchical motion intensity structure is introduced in Section 2. Section 3 describes the detailed techniques employed at each level. Simulation results are given in Section 4. Finally, the conclusions are followed in Section 5.

2. Hierarchical motion intensity structure

In our framework, the MI of a video sequence is represented using a hierarchy of three levels.

2.1. Block level motion intensity

At the block level, we can obtain detailed motion information on a block, including motion history and motion activity.

0-7803-8603-5/04/$20.00 ©2004 IEEE

One Motion History Matrix (MHM) is maintained to obtain the motion history distribution. The value of each entry in MHM indicates the cumulative number of times this block has no motion activity. The i^{th} entry of MHM is updated as following:

$$MHM(i) = \begin{cases} MHM(i)+1 & d=0 \\ 0 & else \end{cases}, \qquad (1)$$

where $d=|x|+|y|$, denoted as the Manhattan distance of its associated MV: (x,y). Fig. 1 gives an example to illustrate the efficiency of MHM. For display purpose, the MHM is linearly mapped to gray level values. The brightness represents the continuous number of frames with no motion activity. The brighter the block in the figure, the longer it has no motion activity in the motion history. We can observe that the MHM contains not only the motion history but also the locations of motions, which can used to predict the static blocks.

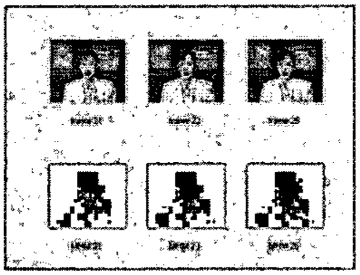

Figure 1: MHM of sequence "akiyo".

High motion correlation exists between current block and its neighboring blocks. The smoother the local MV field (LMVF), which is defined as the set of MVs of the adjacent blocks, the higher the spatial motion correlation is. A smoothness measure of LMVF is the consistency of the MVs from neighbor blocks:

$$MA = max\{var_i \mid var_i = |x_i - x_0| + |y_i - y_0|\}, \qquad (2)$$

where (x_i,y_i) is the neighboring block's MV and (x_0,y_0) is their average MV. According to the regularity of the LMVF, the block level motion intensity (BMI) is determined by Eq. (3). Here, L_1 and L_2 are determined empirically.

$$BMI = \begin{cases} lowMI & MA \le L_1 \\ \text{mediumMI} & L_1 < MA \le L_2 \\ highMI & MA > L_2 \end{cases}, \qquad (3)$$

2.2. Frame level motion intensity

Motion vector prediction is an efficient technique to improve the performance of ME. A good motion vector predictor is usually close to the global optimum. However, the performance of motion vector prediction depends on the motion correlations. Fig. 2 and Fig. 3 give examples of the distribution of motion activity (DMA) for "Akiyo" and "Stefan" respectively. In Fig. 2, the motion vector field is regular, thus the spatial correlation of MVs is high. In Fig. 3, the motion vector field is disorder and the correlation is small. Hence, the regularity of the motion activity can be considered as a measure of the effectiveness of the motion vector predictor, which usually is predicted by its adjacent blocks.

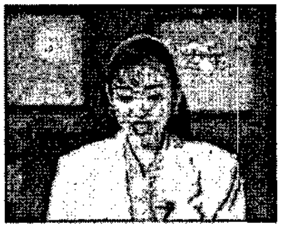

Figure 2: DMA for "Akiyo" at Frame 37.

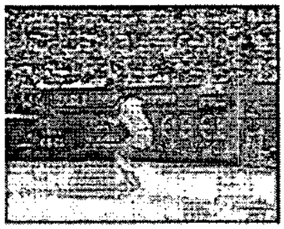

Figure 3: DMA for "Stefan" at Frame 91

We classify the motion activity x of each block as Eq. (4):

$$x = \begin{cases} lowMI, & d = 0 \\ \text{mediumMI}, & 0 < d \le 4 \\ highMI, & d > 4 \end{cases}, \qquad (4)$$

The frame level motion intensity (FMI) is determined according to the regularity of the motion activities. FMI represents the spatial distribution attribute of motion activities. According to the distribution of the motion activities in the frame, the expected FMI can be obtained by:

$$EFMI = \sum(x*p(x)), \qquad (5)$$

where $p(x)$ is the corresponding percentage of motion activity in one frame. However, *EFMI* is not very precise since it is an average value. We use the entropy to further refine the expected motion intensity.

$$FMI = \begin{cases} EFMI, & En \le c1 \\ mediumMI, & c1 < En \le c2 \\ highMI, & En > c2 \end{cases}, \qquad (6)$$

where $E_n=\sum p(x)log(1/p(x))$, is the entropy of the motion activity distribution. The thresholds above are obtained empirically.

2.3. Sequence level motion intensity

The motion intensity histogram (MIH) of FMIs characterizes the attribute of temporal motion intensity distribution of the sequence. It represents the motion property at the sequence level. Because the motion of one sequence tends to be stable continuously in time, we use the dominant motion intensity level of MIH as the sequence level motion intensity (SMI), representing the motion characteristic at the sequence level.

3. The proposed FMEHMI

The performance of ME highly depends on the motion characteristics. With the hierarchical MI structure, we can get fine motion details to improve the ME performance. Considering the new motion estimation/compensation model of H.264/AVC, we jointly apply different ME techniques based on the proposed structure, from block level to sequence level.

3.1. Reference frame selection

Multiple reference frames are employed in H.264/AVC. By adaptively limiting the number of reference frames, a significant amount of ME computations can be saved without loss of accuracy: When the motion intensity is high at each level, we only use several most recent frames, say the last 2 frames, as the reference frames; When BMI, FMI are low and its MHM value is bigger than a predefined threshold, we predict this block has minor motion and the frame with minimum SAD (sum of absolute difference) at MV(0,0) is chosen as the reference frame; Otherwise, all the reference frames are checked.

3.2. Adaptive search range

A good MV predictor is close to the optimum MV. Around this predictor, the search range can be decreased. In FMEHMI, since the accuracy of the prediction can be estimated from the hierarchical motion intensity structure, the search range can be adjusted adaptively according to the MIs. When BMI is low and FMI is medium, or when BMI is medium and FMI is either low or medium, this block tends to have small motion, we reduce the search range to a predefined value. When BMI and FMI are low, this block is a static block with high probability and the search range can be further decreased.

3.3. Extended motion vector predictor set

Motion correlation also exists among different block size modes in H.264/AVC. For example, MV of a 8x8 block is a good MV predictor for the 4x8, 8x4, 16x8 and 8x16 blocks. To get more accurate MV predictor, the MV candidate list (MVCL) is enlarged in FMEHMI. The MVCL consists of the median MV predictor [8], the motion inertia MV predictor [9], and the MVs of other block sizes. The candidate in MVCL with minimum SAD value is the refined MV predictor.

3.4. Early search termination

To avoid unnecessary SAD computations, an adaptive threshold *TH0* is used to terminate the search process quickly. *TH0* can be predicted by the SADs of adjacent blocks, denoted as *TH1*. In H.264/AVC, *TH0* can also be predicted by the SADs of different block size modes, denoted as *TH2*. In our algorithm, *TH0* is chosen according to the MI at different levels. If BMI is low and FMI is low or medium, the probability is high that the distortion surface around the MV is flat and *TH0* is set to the maximum value of *TH1* and *TH2*; otherwise, it is their minimum value. To avoid the extremely low or high value of *TH0* caused by noise, we bound *TH0* within a certain range.

3.5. Combined search pattern

In the proposed combined search pattern (CSP), as shown in Fig. 4, the area with higher probability to find the optimum MV is checked in a more detailed way with SDP, while the area with lower probability is checked with LDP. The SWP is used to compensate the prediction error brought by SDP and LDP.

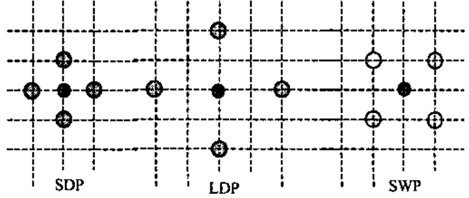

Figure 4: Combined Search Pattern (CSP).

We define a parameter ***pattern_factor*** to switch the search pattern, which is determined by BMI. We start with SDP. If the SAD of search center is found as the minimum, we stop. Otherwise if the number of successively executed SDPs exceeds ***pattern_factor***, the search pattern is changed to LDP. In LDP, if the SAD of search center is the minimum, we do one SDP search again, then terminate. Otherwise, the point with minimum SAD is used as the new search center of SWP. In SWP, we will switch back to LDP with the point with the minimum SAD as the search center.

3.6. Search process

By combining the above techniques, the proposed FMEHMI is summarized as following:

Step 1: Initially, the motion intensity at each level is considered as low motion intensity. The MHM is initialized to 0.

Step 2: For the incoming coding block, calculate the BMI, FMI and SMI.

Step 3: Perform the search termination threshold decision. The threshold is used in every MV candidate evaluation to terminate the search.

Step 4: Perform the reference frame selection and search range adjustment.

Step 5: Evaluate the extended motion vector predictor set to get the starting search center.

Step 6: Determine ***pattern_factor***, start CSP search.

Step 7: After obtaining the MV, update MHM. If motion estimation for current frame is not finished, go to step 2 to begin the ME for the next block. Otherwise update FMI and SMI.

4. Simulation results

The simulation experiments are taken on the H.264/AVC reference software version JM6.1e. The MVFAST [3] and AMSED [9] are implemented in H.264/AVC to perform the comparison. 16 sequences that cover a wide range of video content, including typical low motion talking head sequences such as Akiyo, and high motion sequences such as Stefan are tested in our experiments.

Fig. 5 shows the PSNR comparison for "Akiyo" in a frame-by-frame manner. The number of encoded bits used in these algorithms is roughly the same. We can see that the video quality performance of our proposed algorithm outperforms MVFAST, AMSED and is close to that of FS. Table 1 summaries the average video distortion and speedup performance of MVFAST, AMSED, and FMEHMI compared with FS in terms of PSNR gain and number of searched points per MV search for the tested sequences. The results indicate that FMEHMI has similar video quality performance to MVFAST and AMSED. However, FMEHMI achieves 50% speedup over MVFAST, 28% speedup over AMSED using 1 reference frame, and 80% speedup over MVFAST, 51% speedup over AMSED using 5 reference frames, while keeping similar, even better, video quality.

5. Conclusions

In this paper, we propose a novel fast ME algorithm using hierarchical motion intensity structure to address the problem of the huge computational complexity introduced by the new motion compensation model in H.264/AVC. By jointly applying different ME techniques at different motion intensity levels, the speed and accuracy of ME are improved. Experimental results show that our algorithm is very efficient in terms of the computational speedup and video reconstruction quality. We believe that our work provides promising performance and will be useful in the future development of H.264/AVC Codecs.

6. References

[1] C.K. Cheung and L.M.Po, "A Hierarchical Block Motion Estimation Algorithm Using Partial Distortion Measure," *Proceeding of ICIP'97*, Vol3, pp. 606-609, 1997.

[2] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC) - Joint Committee Draft," document JVT-G050r1.doc, May. 2003.

[3] ISO/IEC JTC1/SC29/WG11 N3324, "Optimization Model Version 1.0," Noordwijkerhout, NL, March 2000.

[4] J.R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1808, Dec. 1981.

[5] T. Koga, K. Ilinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. NTC81*, New Orleans, LA, pp. "C9.6.1-C9.6.5, Nov./Dec. 1981.

[6] R. Li, B. Zeng, and M. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, NO. 4, pp. 438-442, Aug. 1994.

[7] Xudong Song, Tihao Chiang, and Ya-Qin Zhang, "A Scalable Hierarchical Motion Estimation Algorithm for MPEG-2," *Proceeding of ICIP'98*, pp. IV126-IV129, 1998.

[8] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) - Enhancing Block Based Motion Estimation," proceedings *of Visual Communications and Image Processing 2001 (VCIP-2001)*, San Jose, CA, January 2001.

[9] W. Zheng, I. Ahmad, and M. Liu, "Adaptive Motion Search with Elastic Diamond for Mpeg-4 Video Coding," in Detection and Tracking of motion, *ICIP2001*.

[10] S. Zhu, K.K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. Image Processing*, vol. 9, NO. 2, pp. 287-290, Feb. 2000.

Table 1: Overall performance comparisons.

QCIF	Ref = 1		Ref = 5	
	PSNR gain (dB)	Speedup	PSNR gain (dB)	Speedup
MVFAST	-0.04	198	-0.03	175
AMSED	-0.04	234	-0.03	209
FMEHMI	-0.03	299	-0.04	316
CIF	Ref = 1		Ref = 5	
	PSNR gain (dB)	Speedup	PSNR gain (dB)	Speedup
MVFAST	-0.04	199	-0.03	176
AMSED	-0.04	226	-0.03	203
FMEHMI	-0.03	278	-0.03	275

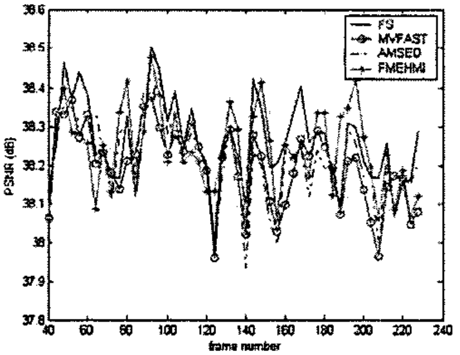

Figure 5: PSNR comparison (Akiyo).