An Efficient Search Range Decision Algorithm for Motion Estimation in H.264/AVC

Mohammed Golam Sarwer, and Q. M. Jonathan Wu

***Abstract*—**Variable block size motion estimation (VBME) is a new feature introduced in H.264/AVC video coding standard. VBME plays a significant rule in achieving outstanding performance in compression efficiency and video quality. However, the VBME is the most time consuming part of H.264/AVC encoder. In order to reduce computations in motion estimation module, this paper presents a novel adaptive search area selection method by utilizing the information of the previously computed motion vector differences (MVDs). The direction of picture movement of the previously computed blocks is also considered for search area selection. In this algorithm, narrow search ranges are chosen for areas in which little motion occurs and wide ranges are chosen for areas of significant motion. Experimental results show that the proposed algorithm provides significant improvement in coding speed with negligible objective quality degradation compared to the full search motion estimation method adopted by H.264/AVC reference software.

***Keywords*—**H.264/AVC, motion estimation, search range, video coding.

I. Introduction

As a result of rapid development of digital techniques and increasing use of internet, image and video compression plays a more and more important role in our life. H.264/AVC is the newest video coding standard which offers a significant performance improvement over previous video coding standards such as H.263++ and MPEG-4 part 2 [1]. New and advanced techniques are introduced in this new standard, such as intra prediction for I-frame encoding, variable block-size motion estimation (VBME), small block-size transform coding, context-adaptive arithmetic entropy coding, de-blocking filtering, etc. These advanced techniques are incorporated in new standards and provide approximately 50% bit rate saving for equivalent perceptual quality relative to the performance of prior standards [2]. Among several new features that are introduced in H.264/AVC; motion estimation (ME) is computationally intensive in comparison to traditional algorithms and accounts for about 80% of the total computation complexity. Therefore, development of efficient algorithms for ME of H.264/AVC is one of the most challenging themes. Various algorithms have been proposed to reduce the computational complexity of ME module such as three step search (3SS) [3], and four step search (4SS) [4]. These algorithms utilize monotonic property of block distortion measure (BDM) used to estimate motion vector. However, this property does not always hold true for real-world video sequences, especially with high and irregular motion videos at low frame rates. As a consequence, video quality degradation introduced by these algorithms is relatively high [5].

In order to accelerate H.264/AVC video coding, a number of researches have been made to explore fast algorithms in motion estimation and mode decision [5-15, 24-28]. One of the methods of reducing computation is to stop the ME process early. The variable block-size zero motion detection and variable block-size best motion detection algorithms compare the rate-distortion cost function of the two blocks instead of the sum of absolute differences (SAD) to detect the zero motion blocks or best motion vector for the variable block-size H.264/AVC video coding is proposed in [10]. The efficiency of early termination algorithms is improved by determining the adaptive threshold based on the rate-distortion (RD) cost of highly correlated blocks [11].

Another way to reduce complexity of ME process is to adaptively change the search range [16-21]. The summation of absolute value of previously computed motion vectors and summation of prediction errors are used to change the search range adaptively [19]. In [20], the search range is determined at each block level by using local statistics of neighboring blocks motion vectors (MVs). Unlike in [19] the MVs are directly used irrespective of their size, in [21], three MVs are firstly used together to calculate two statistics, and then based on these statistics, decision on search range is made. However, the fact is, high probability of large MV does not mean a large search window is required, because the search window may not be centered on the (0, 0) vector, it may centered on the search center placed by median Motion Vector Predictor (MVP).

This paper proposes an efficient adaptive search area selection algorithm of H.264/AVC encoder. In this method, instead of MVs, motion vector differences (MVDs) are used. The inter mode correlation is also considered to select the search area. Based on the MVDs of previously computed modes, the search area is changed in block level. Search area

Manuscript received July 5, 2009. This work was supported in part by the National Sciences and Engineering Research Council of Canada.

The authors are with the Computer Vision and Sensing Systems Laboratory, Department of Electrical and Computer Engineering, University of Windsor, ON, N9B3P4, Canada (e-mail: sarwer@uwindsor.ca, jwu@uwindsor.ca).

is also inspired by the orientation of the previously computed MVs. The rest of the paper is organized as follows. Review of motion estimation method of H.264/AVC is given in section II. Section III describes the proposed search area selection method. Simulation results are detailed in section IV. Finally, section V concludes the paper.

II. Motion Estimation in H.264/AVC

In order to best represent the motion information, H.264/AVC allows partitioning a macroblock (MB) into several blocks with variable block sizes, ranging from 16 pixels to 4 pixels in each dimension. For example, one MB of size 16x16 may be kept as is, decomposed into two rectangular blocks of size 8x16 or 16x8, or decomposed into four square blocks of size 8x8. If the last case is chosen (i.e. four 8x8 blocks), each of the four 8x8 blocks can be further split to result in more sub-macroblocks. There are four choices again, i.e. 8x8, 8x4, 4x8 and 4x4. The possible modes of different block sizes are shown in Fig. 1. Each block with reduced size can have its individual motion vectors to estimate the local motion at a finer granularity. Though such finer block sizes incur overhead such as extra computation for searching and extra bits for coding the motion vectors, they allow more accurate prediction in the motion compensation process and consequently residual errors can be considerably reduced, which are usually favorable for the final RD performance.

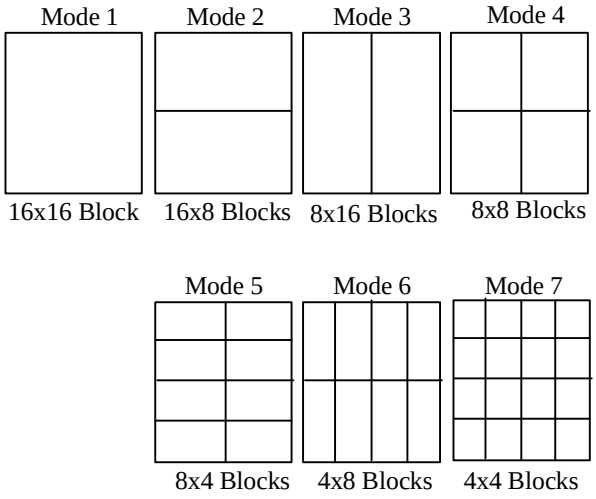


Fig. 1 Block sizes for motion estimation of H.264/AVC

Earlier encoders typically computed the sum of absolute differences (SAD) between the current block and candidate blocks and simply selected the MV yielding the least distortion. However, this approach will often not give the best image quality for a given bit rate, because it may select long motion vectors that need many bits to transmit. It also does not help to determine how subdivision should be performed, because the smallest blocks will always minimize distortion, even if multiple MVs may use larger amount of bits and increase the bit rate. For this reason, H.264/AVC uses the cost function J, rather than SAD, as the measure of prediction error in selecting the best matching block. The RD cost function J is defined as

$$J(m, l) = SAD(s, c(m)) + l\, R(m - mvp) \quad (1)$$

where $m = (m_x, m_y)^T$ is the current MV, $mvp = (mvp_x, mvp_y)^T$ is the predicted MV, and $SAD(s, c(m))$ is the sum of absolute differences between current block s and candidate block c for a given motion vector m, l is the Lagrangian multiplier and $R(m - mvp)$ is the number of bits required to code the MV.

As shown in Fig. 1, the total number of blocks for all modes is 41. Each block can have its individual motion vector. If the encoder selects one reference frame, to encode a MB 41 MVs should be computed. If full search method is utilized, the number of search point for a macro block is $41(2p+1)^2$, where, p is the search range. Therefore, the complexity and computation load of ME in H.264/AVC increases drastically compared to previous standards.

III. Proposed Adaptive Search Area Selection

In H.264/AVC video coding, the search range p is fixed throughout the encoding process. This constant search range is inefficient for slow motion macroblocks because the distance between best MV and search center is very small. But for large motion MBs, a large search range is more efficient. Therefore, an adaptive search range decision algorithm is necessary for H.264/AVC video coding.

Since an object usually occupies more than one block of an image, motions of the neighbouring blocks are highly correlated. Also due to the inertia of the moving objects, there is a correlation among motion vectors of blocks of consecutive frames. Consequently if the MVs of the previously computed MBs are all high then current MB is highly probable to have large MV. However, the fact is, high probability of large MV does not mean a large search window is required, because the search window is not centered on the (0, 0) vector, it is centered on the search center placed by median motion vector predictor (MVP). For example, if the MVs of all adjacent MBs is (12, 11) and the MV of current MB is also (12, 11), the real situation is that the search center is calculated out as (12, 11) thus a search range of zero will be large enough for finding the correct MV (12, 11). Based on this idea, the previously computed motion vector differences (MVDs) are used to predict the search area of a current MB or mode. In the proposed method, the adaptive search range (ASR) is defined as the weighted average of the previously calculated MV differences.

$$Av\_MVD_x = \left\lceil \frac{\sum_{i=1}^{5} w_i (MV_{x(i)} - MVP_x)}{\sum_{i=1}^{5} w_i} \right\rceil \quad (2)$$

$$Av\_MVD_y = \left\lceil \frac{\sum_{i=1}^{5} w_i (MV_{y(i)} - MVP_y)}{\sum_{i=1}^{5} w_i} \right\rceil \quad (3)$$

$$ASR = \max(|Av\_MVD_x|, |Av\_MVD_y|) + \Delta R \quad (4)$$

Where, (MV_x, MV_y) are the horizontal and vertical components of previously computed MVs , (MVP_x , MVP_y) is the motion vector predictor, (Av_MVD_x, Av_MVD_y) is the weighted average motion vector difference, ASR is the adaptive search range, w is the weighting factors, and ΔR is the penalty function. The notation $\lceil\ \rceil$ denotes rounding up to next integer. Table 1 shows the candidate MVs and their corresponding weighting factors for ASR calculation of different modes.

Table 1. Previously computed motion vectors (MVx, MVy) and corresponding weighting factors (Wi) of equation (2) and (3)

Mode	Previously computed MVs and corresponding weighting factor (w_i)
16x16	Collocated (2), up (2), left (2), up-left (1), up-right (1)
16x8	16x16 (2), Collocated (2), up (2), left (1), up-left (1)
8x16	16x16 (2), 16x8 (2), Collocated (2), up (1), left (1)
8x8	16x16 (2), 16x8 (2), 8x16(2), Collocated (1), up (1)
8x4	8x8 (2), 16x16 (2), 16x8 (2), 8x16(1), Collocated (1)
4x8	8x8 (2), 8x4 (2), 16x16 (2), 16x8 (1), 8x16(1)
4x4	8x8 (2), 8x4 (2), 4x8 (2), 16x16 (1), 16x8 (1)

In H.264/AVC motion estimation process, the processing order of different modes is 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4. So, during motion estimation process of 16x8 mode, the information about the best MV of 16x16 mode is available. Similarly while processing 4x4 block, the best MVs and best RD costs of all other modes are already computed. So we can estimate the search area of lower size block from motion information of the higher size block of same MB. In case of 16x16 mode, the information about MV of other modes for same MB is not available. But MVs of neighboring MBs and collocated MBs in the previous frames are already computed.

In Table 1, the corresponding weighting factors of the set of the MVs are given within braces. Motion vector of 16x16 mode is highly correlated with that of collocated, up and left MB as compared to up-left and up-right MVs [11]. Therefore, weighting factors of collocated, up and left modes should be larger than that of up-left and up-right MVs. Similarly, motion vector of 4x4 mode is highly correlated with that of 8x8, 8x4, and 4x8 mode [11]. That's why the weighting factors of these modes are higher than other modes. Similar observation is applied to select weighing factors of all other modes.

In order to select a penalty function ΔR, we have used the initial cost of the search center (J_{MVP}). If J_{MVP} is low, it means that MVP is probably close to best MV. On the other hand, if J_{MVP} is high, it means that MVP is not so correlated with current MV, and thus the size of the search range should be larger. Based on this observation ΔR is defined as follows:

$$\Delta R = \begin{cases} 0 & \text{if E <= 2} \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

$$\text{and} \qquad E = \frac{J_{MVP}}{M \times N} \quad (6)$$

where E is the average pixel error, J_{MVP} is the RD cost of the search center and MxN is the size of the block.

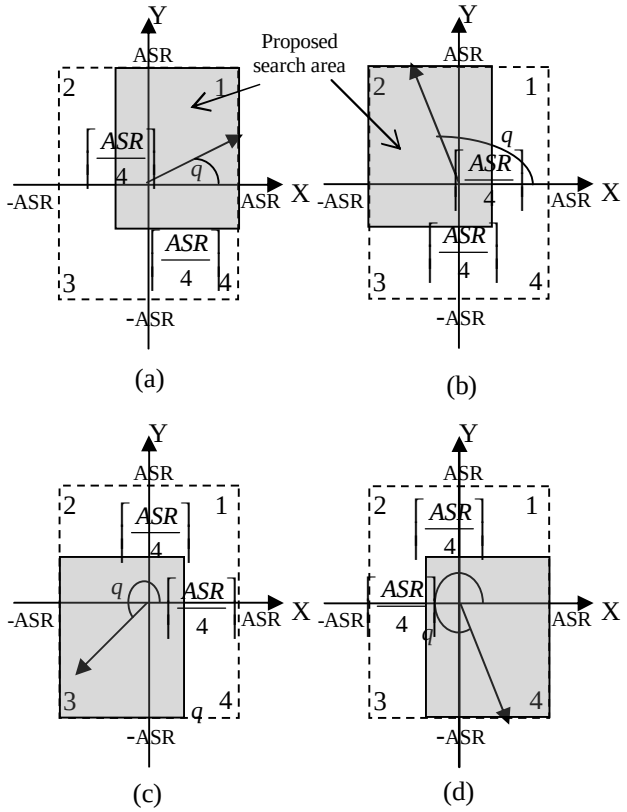


Fig. 2 Proposed search area with (a) most probable quadrant = 1, (b) most probable quadrant = 2, (c) most probable quadrant = 3, (d) most probable quadrant =4

Dotted squares in fig. 2 indicate adaptive search window. Search window is square which implies that the search range is same in both horizontal and vertical directions but motion vectors of real video sequences are directional oriented. If all of the correlated blocks move in a particular direction, the movement of current block should be also in the same direction. Therefore a square search window is not efficient and in order to tackle this issue, we have used orientation of weighted average motion vector, calculated as follows:

$$q = \tan^{-1}(Av\_MVD_y / AV\_MVD_x) \quad (7)$$

If q falls in first quadrant, the MV of current block is most likely to be in the first quadrant. In this paper, the quadrant at which q falls is called the most probable quadrant. Therefore, search ranges of other quadrants are reduced from ASR to $\lceil ASR/4 \rceil$. Here $\lceil\ \rceil$, means rounding up to integer value. Grey area in fig. 2 shows the modified search window for different values of most probable quadrants.

IV. Simulation Results

To evaluate the performance of the proposed method, JM 9.6 [22] reference software is used in simulation. Different types of video sequences with CIF format are used as test materials. Each sequence has 100 frames, hence, they represent a wide range of motion contents.

Table 2. Simulation Conditions

GOP Structure	IPPPP../IBPBP..
Quantization Parameters	12/16/20/24
Number of Reference Frame	1
Hadamard Transform	OFF
Search range for FS	16
Symbol mode	CAVLC
RDO optimization	ON
Frame rate	30/60 fps
Number of frames	100

A group of experiments were carried out on the test sequences with different quantization parameters and frame rates. All the simulations are conducted under Windows Vista operating system, with Pentium 4 2.2 G CPU and 1 G RAM. The conditions of the simulation are listed in Table 2. The comparison results are produced and tabulated based on the average difference of integer pixel ME time, the average PSNR differences ($\Delta PSNR$), and the average bit rate differences ($\Delta Rate\%$). PSNR and bit rate differences are calculated according to the numerical averages between RD curves derived from the original and proposed algorithm, respectively. The detail procedure to calculate these differences can be found in [23].

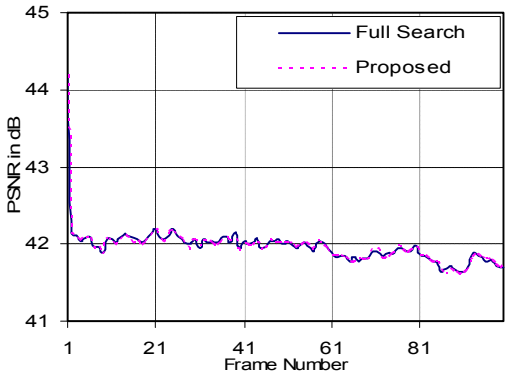
A. Comparison with Full Search ME

In this experiment, the proposed method is compared with full search ME of H.264/AVC. Both IPPPP and IPBPBP sequences are used in simulations. Comparison results with IPPPP sequences are tabulated in Table 3. The positive values mean increments whereas negative values mean decrements. The complexity reduction $\Delta T_1\%$ is calculated as follows and average values presented.

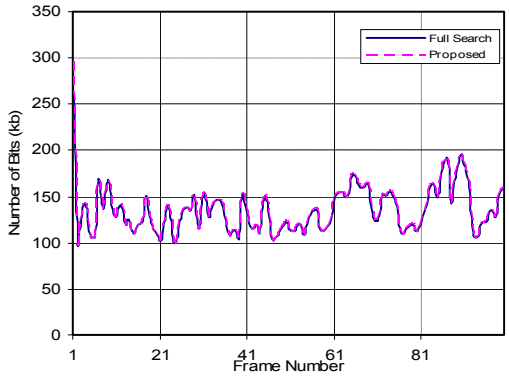
$$\Delta T_1\% = \frac{T_{proposed} - T_{FS}}{T_{FS}} \times 100\% \quad (8)$$

where, T_{FS} denotes the integer pixel ME time of the JM 9.6 encoder with full search ME and $T_{proposed}$ is the integer pixel ME time of the encoder in the proposed method.

Fig. 3 shows the frame by frame comparison of the proposed method with FS with IPPPP.. sequences in terms of PSNR and number of bits. Both PSNR and bit rate of the proposed method are almost same as FS. In these plots, 100 frames of *Flower* video sequence are encoded with quantization parameter equal to 20. The PSNR and rate of the first frame is higher than others because first frame is encoded with intra-coding and remaining frames are encoded with inter-coding.

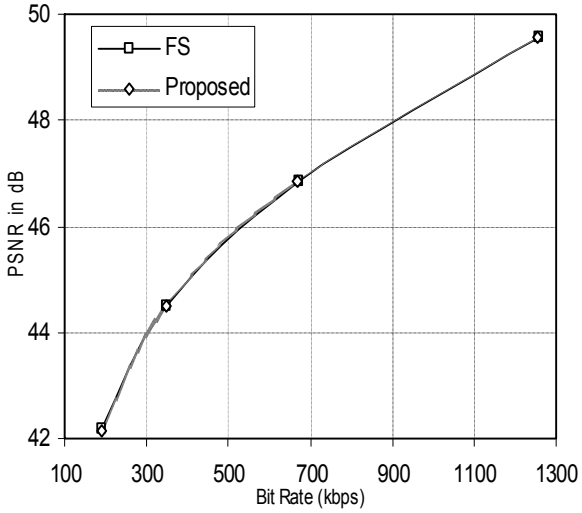


(a)

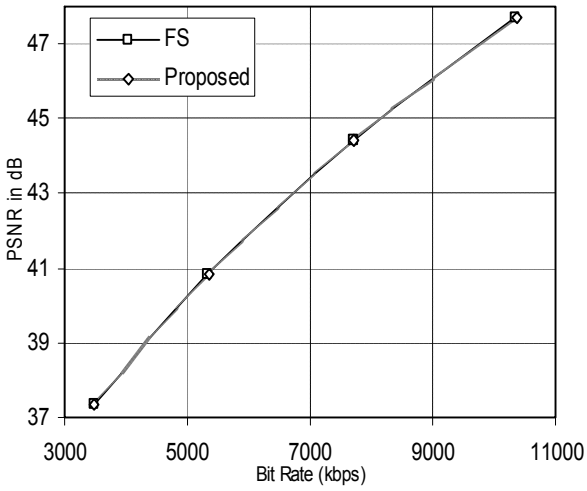


(b)

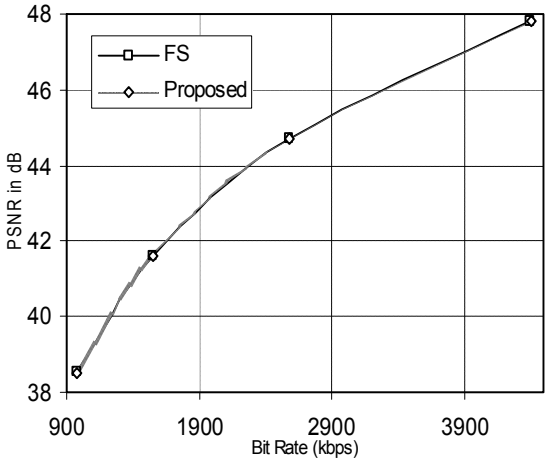
Fig. 3 Frame by frame comparison of the proposed method with FS motion estimation (a) PSNR Comparison (b) Bit rate Comparison



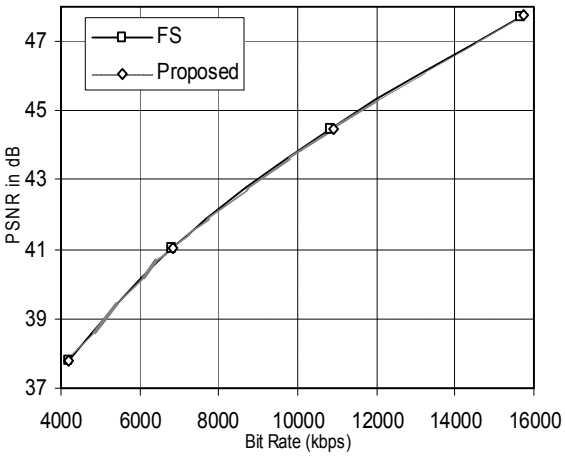
(a) Akiyo@30 fps



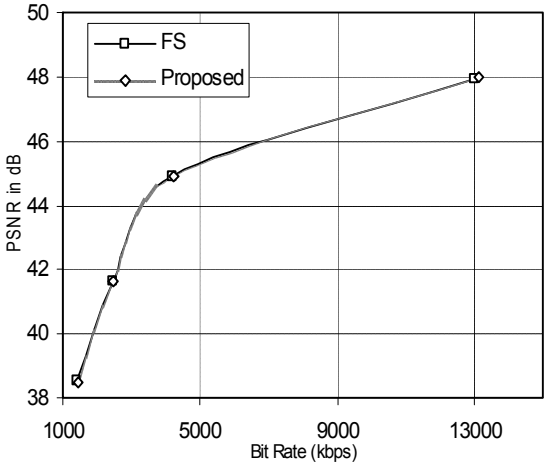
(b) Mobile@30 fps



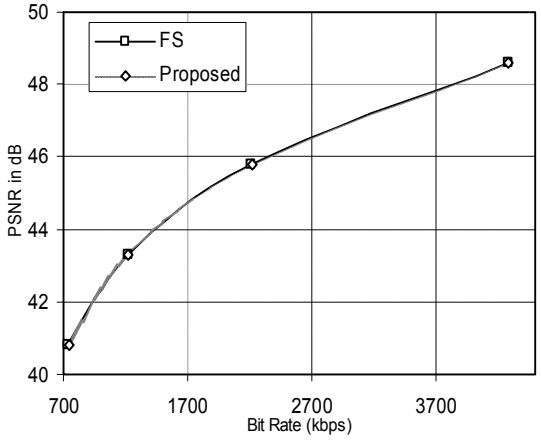
(c) Paris@30 fps



(d) Bus@60 fps

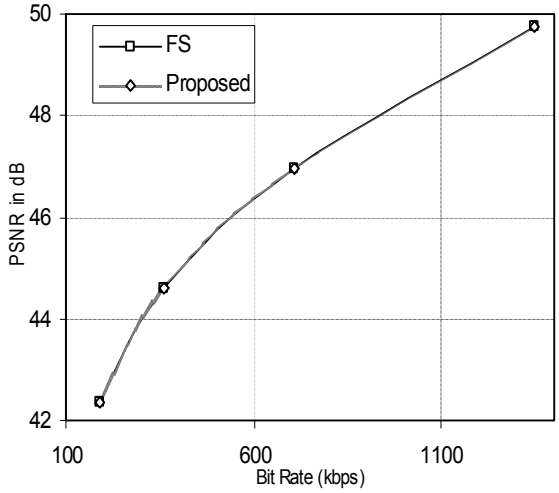


(e) Tennis@60 fps

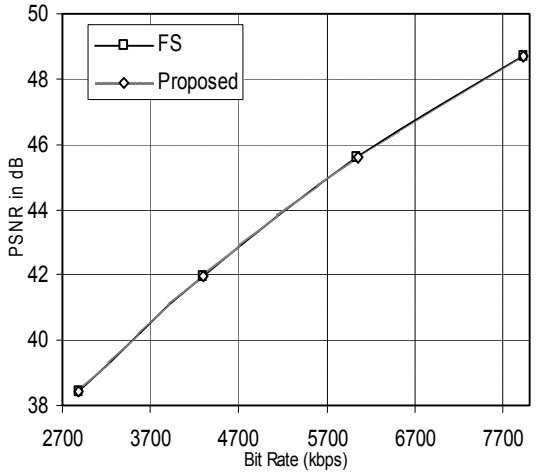


(f) News@60 fps

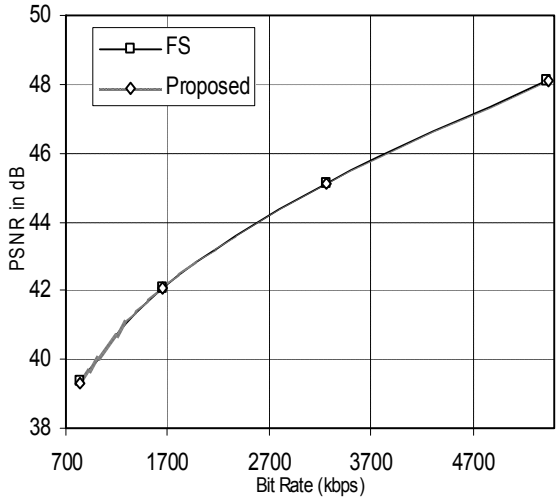
Fig. 4 Rate-distortion (RD) curves of different sequences with IPPP.. frames



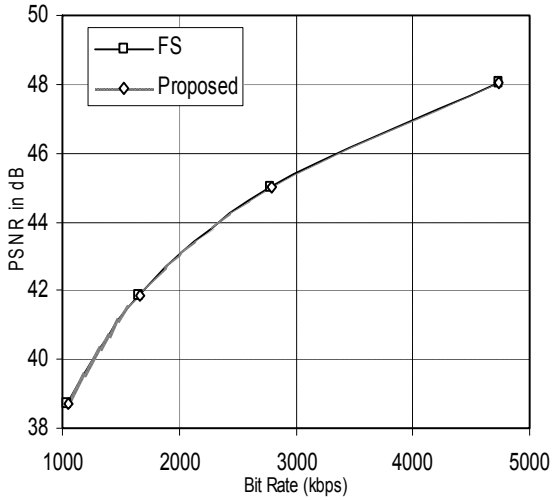
(a) Akiyo@30 fps



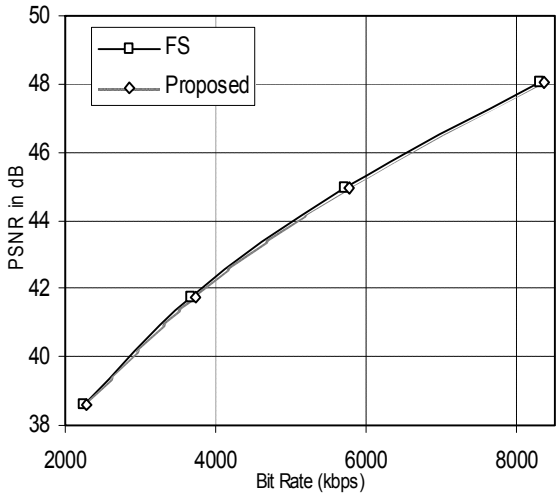
(b) Flower@30 fps



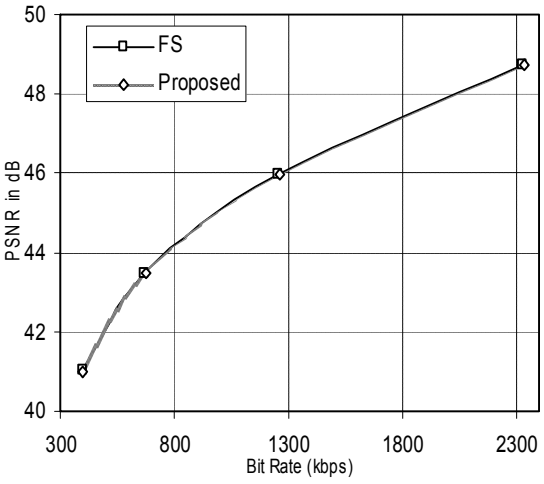
(c) Foreman@30 fps



(d) Paris@30 fps



(e) Stefan@30 fps



(f) News@30 fps

Fig. 5 Rate-distortion (RD) curves of different sequences with IPBPBPB.. frames

From Table 3, it is clear that the proposed algorithm yields up to 92% of integer pixel ME time savings, compared to a full search ME of IPPP frames with a negligible PSNR reduction (0.015 db) and bit rate increment (0.47%). The speed up of the proposed algorithm is quite significant for various motion videos and on average the proposed method saves 84% computational cost in comparison with full search ME. The computation reduction is more pronounced and an utmost 92% of FS for low motion sequence such as Akiyo and News. Most of the motion vectors of these sequences are around the search center so the adaptive search range is low.

Table 3 Comparison with full search ME with IPPP.. sequences

Frame rate	Sequence	Δ PSNR	Δ Rate%	ΔT_1 %
30 fps	Foreman	-0.003	0.178 %	-82.28 %
	Paris	-0.018	0.591 %	-84.24 %
	Stefan	-0.023	0.689 %	-74.79 %
	Flower	-0.015	0.317 %	-81.27 %
	Akiyo	-0.007	0.005 %	-92.65 %
	Mobile	-0.002	0.060 %	-86.57 %
60 fps	News	-0.017	0.773 %	-88.37 %
	Bus	-0.031	0.860 %	-82.95 %
	Tennis	-0.019	0.797 %	-84.98 %
Average		-0.015	0.474 %	-84.23 %

Table 4 Comparison with full search ME with IPBPBP.. sequences

Sequence	Δ PSNR	Δ Rate%	ΔT_1 %
Foreman	-0.010	0.618	-81.26
Paris	-0.034	1.140	-86.28
Stefan	-0.046	1.206	-73.18
Flower	-0.013	0.280	-82.24
Akiyo	-0.006	0.359	-92.84
News	-0.041	1.910	-85.25
Average	-0.025	0.918	-83.50

The RD curves for full search ME and the proposed method of different sequences at dissimilar frame rates are presented in fig. 4. In this figure IPPP sequences are used. We can see that the RD curves of proposed method closely follow the original full search method. As shown in Table 4, the proposed method saves about 83% of computation of IBPBPBPBP sequences with negligible degradation in RD performance. The RD curves comparison of B frame coding is presented in fig. 5. The RD curves of the proposed method closely match with the original standard.

B. Comparison with other methods

In this experiment, the proposed method is compared with two different methods in terms of RD performance and complexity. The complexity reduction ΔT_2 % is calculated using eq. (9) and average results presented.

$$\Delta T_2 \% = \frac{T_{proposed} - T_{oth}}{T_{oth}} \times 100\% \quad (9)$$

where, T_{oth} denotes the integer pixel ME time of the JM 9.6 encoder with other method and $T_{proposed}$ is the integer pixel ME time of the encoder in the proposed method. Table 5 shows the comparison of our proposed method with search range decision method presented in [19] and [21]. It is shown that as compared with reference [19], our algorithm reduces bit rate by about 2.58 % and achieve 0.16 dB improvement in PSNR on average with up to 26% computation reduction. The proposed algorithm not only saves up to 36% computation time of [21] but also improves the rate-distortion performance.

Table 5 (a) Comparison with [19]

Sequence	Δ PSNR	Δ Rate%	ΔT_2 %
Foreman	0.18	-3.77 %	-26.05 %
Paris	0.19	-2.23 %	-22.27 %
Stefan	0.10	-2.93 %	-25.28 %
Flower	0.13	-2.76 %	-20.24 %
Akiyo	0.20	-1.24 %	-8.17 %
Average	0.16	-2.58 %	-20.40 %

Table 5 (b) Comparison with [21]

Sequence	Δ PSNR	Δ Rate%	ΔT_2 %
Foreman	0.03	-1.47 %	-36.34 %
Paris	0.03	-1.24 %	-23.58 %
Stefan	0.06	-1.90 %	-18.88 %
Flower	0.01	-0.31 %	-21.81 %
Akiyo	0.005	-0.28 %	-18.84 %
Average	0.135	-1.04 %	-23.89 %

V. CONCLUSION

In this paper, we have proposed an adaptive search range decision algorithm for variable block size motion estimation strategy based on motion vector differences of neighboring and collocated blocks. The adaptive search range method determines the size of the search range dynamically. The proposed method is very simple and efficient and experimental results confirm that the proposed method not only cuts down the computational complexity but also improves coding efficiency and picture quality as well. The proposed method can be easily applied to many mobile video application areas such as a digital camera.

ACKNOWLEDGMENT

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, *ITU-T Rec. H.264/ISO/IEC 14496-10 AVC*, 2003.

[2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan," Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits and Syst. Video Technol.*, vol 13, no. 7, pp.688-703, July 2003.

[3] R. Li, B. Zeng, and M. L. Liuo, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Tech..*, vol. 4, no. 4, pp.438-442, August 1994.

[4] L.M. Po and W.C. Ma, "A novel four step search algorithm for fast block motion estimation," *IEEE Trans. on circuit and system for video. Tech.*, vol. 6, no. 3, pp.-313-317, June 1996.

[5] Zhibo Chen, Peng Zhou, Yun He, "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," *Joint Video Team (JVT) Docs*, JVT-F017, Dec. 2002.

[6] Mohammed Golam Sarwer, and Lai Man Po "Fast bit rate estimation for mode decision of H.264/AVC" *IEEE Trans. on circuit and system for video. Tech*., volume 17, number 10, October 2007, pp 1402-1407.

[7] Yu-Kuang Tu, Jar-Ferr Yang, Ming-Ting Sun, and Yuesheng T. Tsai, "Fast variable block motion estimation for efficient H.264/AVC encoding." *Signal Processing: Image Communication* 20 (2005) 595-623.

[8] Feng Pan, Hongtao, and Zhiping Lin, "Scalable fast rate-distortion optimization for H.264/AVC," *EURASIP journal of applied signal processing*, vo. 2006, Article ID 37175, pages1-10.

[9] Yao-Chung Lin, and Torsten Fink, Erwin Beller, "Fast mode decision for H.264 based on rate-distortion cost estimation," *ICASSP-2007*, pp 1137-1140

[10] Yang L., Yu K., Li J., Li S. "An effective variable block size early termination algorithm for H.264 video coding," *IEEE Trans. Circuits Syst. Video Tech..*, vol. 15, no. 6, pp.784-788, June 2005.

[11] M.G. Sarwer, T. M. Nguyen, Q.M.J. Wu, " Fast Motion estimation of H.264/AVC by adaptive early termination," Proceedings of the 10th IASTED *International Conference Signal and Image Processing (SIP 2008)*, August 18-20, 2008, HI, USA, pp. 140-145.

[12] Xiao Sul, Sweta Singh, and Yan Bai, "Local reference with early termination in H.264 Motion estimation," ICME 2007, pp.384-387

[13] Hasan F. Ates, and Yucel Altunbasak, "Rate-distortion and complexity optimized motion estimation for H.264/AVC video coding", IEEE Trans. Circuits Syst. Video Tech.., vol. 18 no. 2 pp.159-171, Feb-2008.

[14] Chun-Man Mak, Chi-Keung Fong, and Wai-Kuen Cham, "Fast Motion Estimation for H.264/AVC in Walsh–Hadamard Domain", IEEE Trans. Circuits Syst. Video Tech.., vol. 18 no. 6 pp.735-745, June-2008.

[15] Wenqi You, Yang Song, Takeshi Ikenaga, and Satoshi Goto, "A High Quality Fast Motion Estimation Algorithm for H.264/AVC", 2008 Congress on Image and Signal Processing, pp. 375-379.

[16] Si-Woong Lee, Seong-Mo Park, and Hyun-soo Kang, " Fast Motion estimation with adaptive search range adjustment", Optical Engineering Letter, vol. 46, no. 4, April 2007.

[17] Z. Chen, Y. Song, T. Ikenaga, and S. Goto, " Adaptive search range algorithms for variable block size motion estimation in H.264/AVC", IEICE Trans. Fundamental, vol. E91-A, no.4, pp. 1015-1022, April 2008.

[18] Goel, S.; Ismail, Y.; Bayoumi, M.A., "Adaptive search window size algorithm for fast motion estimation in H.264/AVC standard", 48th Midwest Symposium on Circuits and Systems, 2005, Volume, Issue 7-10 Aug. 2005 Page(s): 1557 – 1560.

[19] T. Yamada, et al., "Fast and accurate motion estimation algorithm by adaptive search range and shape selection," *Proc. ICASSP*, vol.2, pp. 897-900, March-2005.

[20] M. C. Hong and H.H. Oh, "Range Decision for Motion estimation of VCEG-N33," *JVT B022*, Switherland, February 2002.

[21] X. Xu and Y.HE, "Modification of Dynamic Search range for JVT," *JVT Q088-L*, USA, Oct. 2005.

[22] http://iphome.hhi.de/suehring/tml/, JVT Reference Software version 9.6

[23] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," presented at the *13th VCEG-M33 Meeting*, Austin, TX, April 2001.

[24] Mohammed Golam Sarwer, and Q.M. Jonathan Wu "Adaptive Variable Block-Size Early motion estimation termination algorithm for H.264/AVC Video Coding Standard." IEEE Transaction on Circuit and System for Video Technology, volume 19, number 8, August 2009, pp 1196-1201.

[25] Mohammed Golam Sarwer, Lai Man Po and Jonathan Wu, "Fast Sum of Absolute Transformed Difference based 4x4 Intra Mode Decision of H.264/AVC Video Coding Standard," Elsevier Journal of Signal Processing: Image Communications, Volume 23, Issue 8, September 2008, Pages 571-580.

[26] C. L. Lin, J. J. Leou, An Adaptive Fast Search Motion Estimation Algorithm for H.264, WSEAS Transactions on Communications, Issue 7, Volume 4, July 2005, pp. 396-406.

[27] H. Nam, S. Lim, A New Motion Estimation Scheme Using a Fast and Robust Block Matching Algorithm, WSEAS Transactions On Information Science & Applications, Issue 11, Volume 3, November 2006, pp. 2292-2299.

[28] Y. Shi, C. Yi and Z. Cai, "Multi-Direction Cross-Hexagonal Search Algorithms for Fast Block Motion Estimation," *WSEAS Trans. on Computers*, Issue 6, Volume 6, June 2007, pp. 959-963.

Mohammed Golam Sarwer was born in Comilla, Bangladesh. He received his B.Sc. and M.Phil degree both in electronic engineering from Khulna University of Engineering and Technology and City University of Hong Kong in 2001 and 2007, respectively. He was working as a full time faculty member in Khulna University of Engineering and Technology from 2001 to 2005. Currently, he is a PhD candidate in the department of electrical and computer engineering, University of Windsor, Canada. He has published more than 25 peer-reviewed papers in international journals and conferences. His research interest includes video coding, motion estimation, fast mode decision of H.264/AVC.

Mr. Sarwer is the student member of IEEE and the member of International Association of Engineers (IAENG). In 2003, he won the prime minister gold medal due to outstanding academic performance of undergraduate level. In 2009, he won prestigious Fredrick Atskin graduate award, University of Windsor, due to outstanding academic performance of the PhD level.

Q.M. Jonathan Wu received his Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990.

From 1995, he worked at the National Research Council of Canada (NRC) for 10 years where he became a senior research officer and group leader. He is currently a Professor in the Department of Electrical and Computer Engineering at the University of Windsor, Canada. Dr. Wu holds the Tier 1 Canada Research Chair (CRC) in Automotive Sensors and Sensing Systems. He has published more than 100 peer-reviewed papers in areas of computer vision, image processing, intelligent systems, robotics, micro-sensors and actuators, and integrated micro-systems. His current research interests include 3D image analysis, active video object tracking and extraction, vision-guided robotics, sensor analysis and fusion, wireless sensor network, and integrated micro-systems.

Dr. Wu is an Associate Editor for IEEE Transaction on Systems, Man, and Cybernetics (part A). Dr. Wu has served on the Technical Program Committees and International Advisory Committees for many prestigious conferences.