# A Novel Adaptive Search Range Algorithm for Motion Estimation Based on H.264

Youwei Yuan, Wuyi Li, Shiyu Wang

School of Computer Science and Technology,
Hangzhou DianZi University,
Xiasha, Hangzhou, Zhejiang, P.R China,310018

Corresponding email: y.yw@163.com

## Abstract

*Motion estimation (ME) is very vital to video compression. Due to the adoption of the high precision of motion vector (MV) in H.264 encoder, the computational cost increases rapidly, and ME takes about 60% of the whole encoding time. In order to accommodate the new variable block size motion estimation strategy adopted in H.264, this paper proposes a novel adaptive search range(ASR) algorithm as a optimized part based on UMHexagonS. Not only we utilize the median_MVP and interframe information in our ASR algorithm but also a penalty function is included. Experimental results indicate that our proposed method reduces the computational complexity in a certain degree and enhances encoding efficiency but has few changes in the reconstructed image quality and bit rate.*

**Keywords:** H.264/AVC, Motion estimation, ASR, VBSME, UMHexagonS

## 1. Introduction

Compared with previous video CODEC standards, H.264/AVC is not the first but the newest video standard which was published in 2003 by JVT (Joint Video Team) [1]. With high data compression ratio and good quality image, H.264/AVC has gained wide applications. While its superior performance is at the expense of rapidly increasing in computing complexity, which hinder its further application in real-time coding domain. During the whole video encoding process, motion estimation (ME) takes up about 60% of the total encoding time, so it becomes one of the most important and challenging part in the encoding and up to now various kinds of algorithms have been proposed to reduce the computational complexity. For example, full search(FS)、two dimensional logarithmic search(2-D LOGS)[2]、three step search(3SS)[3]、four step search(FSS)[4]、Hexagon-based search(HEXBS)[5] 、Block-based gradient decent search (BBGDS)[6] and diamond search (DS)[7]are all typical of block-matching algorithms. Compared to FS, other algorithms enhance search efficiency in a certain extent with dramatically diminishing computation. However, these ME module are more suitable for application in estimating big range motion vector occasions, they are easy to fall into local optimum in small range situation, causing poor precision of matching and reducing image quality. So many researchers are trying their best to modify the search patterns in order to boost encoding efficiency.

Another way to reduce the complexity of ME process is the adoption of adaptive search range algorithm, especially in the occasion of video sequence based on H.264. Because variable block size is an innovative strategy accepted in H.264. Specifically, there are up to seven kinds of block size mode (BSM): 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4(Fig 1). And they are divided into two levels: one is macro block (MB) level with 16x16, 16x8, 8x16 and 8x8; the other is 8x8 block level including 8x8, 8x4, 4x8 and 4x4. Four MB are carried out one by one and the second level is processed sequentially. Here is a question: In the reference model JM of H.264, it employs a fixed range of search window, which means all kinds of block size will be put in the same search window. It is obviously an unscientific strategy because seven kinds of block have different size. Take 16x16 and 4x4 for example, putting the 4x4 block in the same search window as 16x16 can inevitably cause useless points' search.

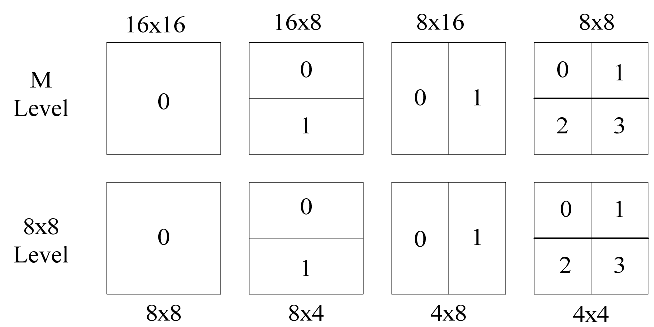


**Fig 1.** Seven block size modes in H.264

This paper proposes an efficient adaptive search range algorithm based on H.264/AVC encoder. In this method, information of interframe and neighboring blocks are both used to inspire new search range. Additionally, we integrate the algorithm with UMHexagonS[8], which is a classic mixed motion search algorithm adopted in H.264. Experiments show that our algorithm gets improvements in reducing ME time and maintaining the certain PSNR. The rest of this paper is organized as follows: Section Ⅱ describes the UMHexagonS simply and our novel algorithm. Section Ⅲ shows the simulation results. Finally, we conclude the paper and give future direction in section Ⅳ.

## 2. UMHexagonS and the proposed algorithm

### *A. Details of the UMHexagonS in H.264*

UMHexagonS is abbreviation of unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) algorithm (Fig 2). It is proposed by Chen, Zhou, and He[8] , which is a great contribution to the development of video encoder. Its specialty is using hybrid and hierarchical motion search strategies. "Hybrid" is because it includes four sub-steps with different search model:1)Predictor candidates selection and reorder the prediction result;2)Unsymmetrical-cross search; 3)Uneven multi-hexagon-grid search; 4)Extended hexagon-based search.

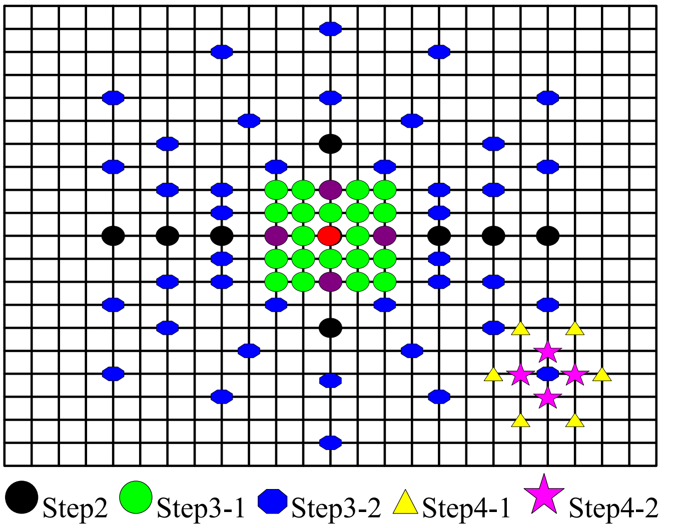


**Fig 2.** Realization steps of UMHexagonS algorithm

### *B. The proposed adaptive search range algorithm*

As mentioned above, it is not wise to use a fixed search window in motion estimation. Our adaptive search range algorithm consists of two parts:

$$ASR = Dynamic\_Searchrange + R \quad (1)$$

ASR means the adaptive size of the search window, and R is the penalty function. Actually the part of Dynamic_Searchrange is made of another two components: the fixed-part and the dynamic-part. Fig 2 shows the relationship among fixed-part, dynamic-part and Dynamic_Searchrange: Dynamic_Searchrange=fixed-part+dynamic-part.

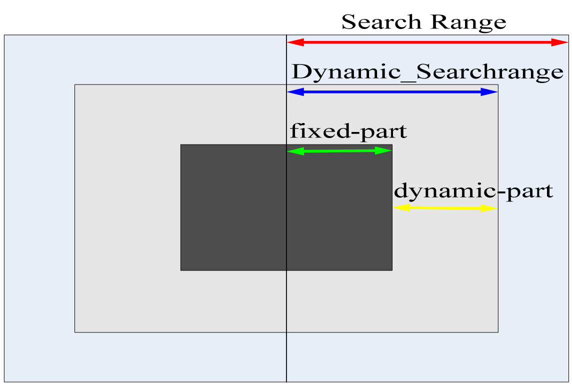


**Fig 3.** Dynamic Searchrange

From the Fig3, we know that Dynamic_Searchrange=fixed-part+dynamic-part, and the fixed-part=search_ range/8, here search_range is a parameter which is set in advance and its value can be 16/32/48/64. Considering the fact that an object usually occupies more than one block of an image and motions of the neighboring blocks are highly correlated, and so does the blocks between frames, consequently we adopt the median_MVP and COL_MV in the dynamic part. The expression is as follows:

$$\text{dynamic-part}=\max\{|MVP\_X-COL\_MV\_X|,|MV-COL\_MV\_Y|\} \quad (2)$$

Where MVP means the median value of the current block's neighboring blocks (Fig 4), and COL_MV represent the block's motion vector in the previous frame which has the same location with the current block (Fig 5). A vector followed by "_X(Y)" stands for its X(Y)-axis projected length [9]. In order to make sure our search range is more accurate, we select a penalty function R [10]. In the formula of R, we have used the part initial cost of the search center. Exactly speaking, it's the sum of absolute differences (SAD) between the current block and candidate block according to the search center. If the SAD is low, it means that the search center vector is probably close to best MV. On the other hand, if SAD is high, it means that search center vector is not so correlated with current MV, and thus the size of the search range should be larger. Based on this observation R is defined as follows:

$$R = \begin{cases} 0 & if\ E <= 2 \\ 1 & otherwise \end{cases} \quad (3)$$

And E=SAD/ (M*N) (4)

Where E is the average pixel error, SAD is the cost of the search center (0,0) and M*N is the size of the current block..

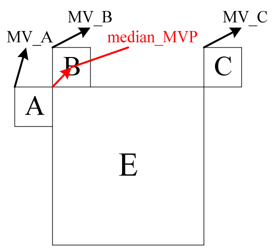


**Fig 4**.Concept of median_MVP

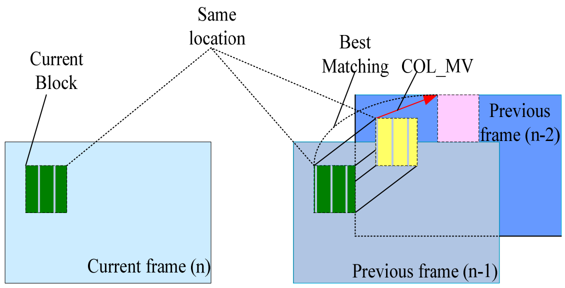


**Fig 5**.Concept of COL_M

## 3. SIMULATION RESULTS

In order to evaluate the performance of our proposed method, JM 10.1 reference software is used in the experiments and different types of video sequences with QCIF format are adopted as test materials. We set each sequence 100 frames to test because they represent a wide range of motion contents and video formats, which can make the simulation results more correct. The test conditions are as follows: all video sequences are tested under the windows XP operating system, and image's YUV=4:2:0, some parameters' setting like this: InputFile ="video sequence needed to test", FramesToBeEneoded = 100, FrameRate = 30.0, SourceWidth =176, SourceHeight=144, UseHadamard = l, SearehRange = 16, UseFME=1, NumberRefereneeFrames = 5, QP=28, others parameters are default setting. Here are our test consequences (Table1):

**Table 1.** Test consequences

| | *UMHexagonS* | | | | *UMHexagonS+Proposed ASR* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *Video sequence* | *PSNR (dB)* | *bit rate (kbit/s)* | *Enc.T (s)* | *ME.T (s)* | *PSNR (dB)* | *bit rate (kbit/s)* | *Enc.T (s)* | *ME.T (s)* | *Rate.Enc (%)* | *Rate.ME (%)* |
| Mobile | 33.09 | 265.72 | 334.913 | 151.752 | 33.08 | 266.42 | 305.427 | 121.862 | 8.804% | 19.697% |
| Coastguard | 34.28 | 169.42 | 307.382 | 162.820 | 34.26 | 169.22 | 278.985 | 134.781 | 9.238% | 17.211% |
| Silent | 36.13 | 66.63 | 253.376 | 118.317 | 36.10 | 67.69 | 238.413 | 107.080 | 5.905% | 9.497% |
| Container | 36.42 | 31.15 | 233.054 | 102.311 | 36.42 | 31.27 | 230.841 | 100.395 | 0.950% | 1.873% |
| Highway | 37.64 | 53.26 | 225.267 | 111.595 | 37.63 | 54.99 | 218.664 | 105.244 | 2.931% | 5.691% |

Enc.T indicates the whole time spent on encoding process, and ME.T has the similar meaning of motion estimation. Rate.Enc means the saving rate compared the original UMHexagonS with UMHexagonS+Proposed ASR on encoding time, and so does the Rate.ME.

From the table, we can get the information that our proposed method does make sense. It reduces both encoding time and ME time in a certain degree but maintaining relatively stable PSNR and bit rate. Figure 6-8 is a typical demonstration of our simulation results, from which we can conclude that our proposed algorithm get better performance not only in reducing the total encoding time but also in motion estimation time compared to UMHexagonS. In addition, the new proposed method keeps a similar PSNR results on average except some single frames. Specially, the proposed algorithm behaves well on mobile and coastguard video sequence, which are the typical of big moving sequences. It will expand H.264's application. However, the proposed ASR method shows not very good performance for "Container" sequence, which is thought as a hinder for how to improve the proposed ASR algorithm in the future work.

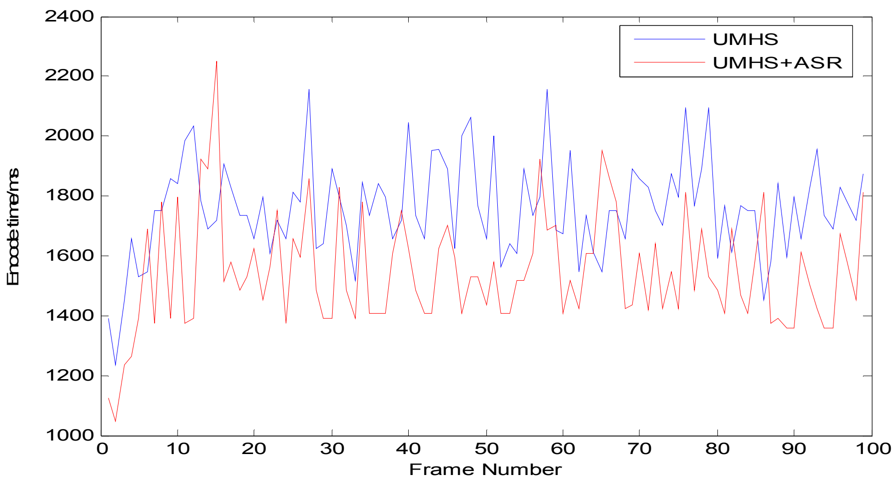


**Fig 6.** Enc.T of mobile_qcif sequence

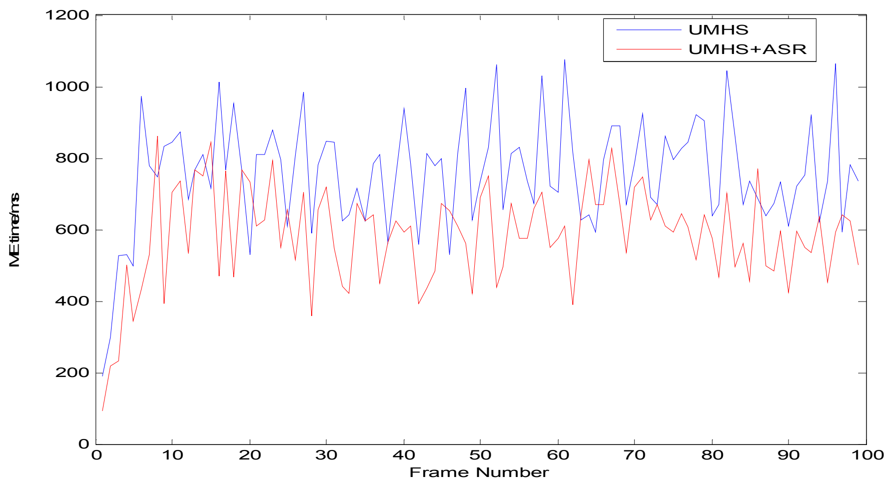


**Fig 7.** ME.T of mobile_qcif sequence

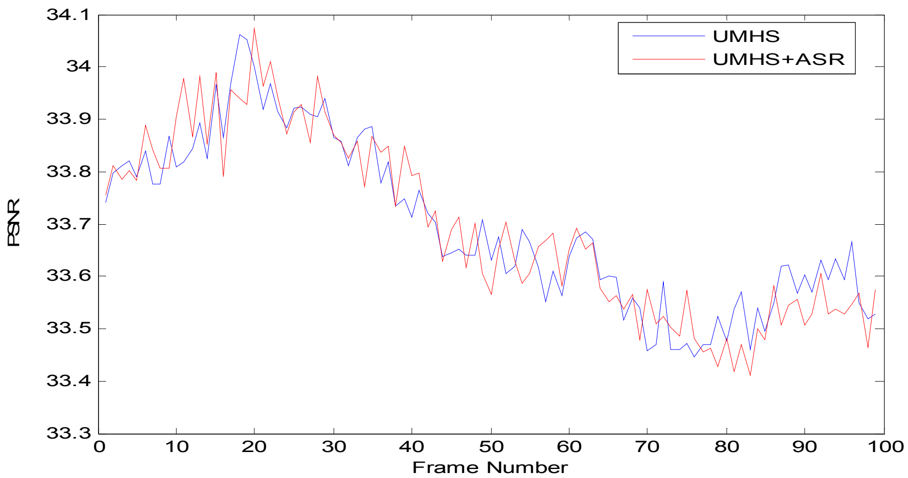


**Fig 8.** PSNR of mobile_qcif sequence

## 4. CONCLUSION

This paper proposes a novel adaptive search range algorithm for variable block size motion estimation to achieve a new tradeoff between encoding efficiency and control overhead. Simulation results show that our proposed method provides almost the same encoding quality but consumes less time both in encoding and motion estimation compared with the original UMHexagonS algorithm. To sum up, the adaptive search range algorithm determines the size of the search window dynamically, and above all it’s very simple and it not only cuts down the computational complexity but also maintains a similar picture quality.

## 5. ACKNOWLEDGEMENT

The work was supported by NSFC (Grant No. 60873023) and also supported by the science foundation of Hangzhou Dianzi University (No. KYS055608080, KYS225608037).

## REFERENCES


[1] T. Wiegand, G.J. Sullivan, A. Luthra, Overview of the H.264/AVC Video Coding Standard, IEEE Trans. On Circuits and System for Video Technology, 2003, 13(7): 560-576.

[2] J. Jain, A. Jain, “Displacement measurement and its application in interframe image coding”, IEEE Transactions on Communications, vol. COM-29, pp. 1799-1806, Dec. 1981.

[3] T. Koga, K. Iinuma, A. Hirano, Y. Lijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing”, in Proceedings Nat. Telecommunications Conf. 81, New Orleans, LA, pp. G5.3.1-G5.3.5, Nov. 1981

[4] L. M. Po, W. C. Ma, “A novel four-step search algorithm for fast block motion estimation”, IEEE Transactions on CSVT, vol. 6, no. 3, pp. 313-317, June 1996.

[5] Ce Zhu, Xiao Lin, and Lap-Pui Chau, “Hexagon-Based Search Pattern for Fast Block Motion Estimation”, IEEE Trans On CSVT, pp. 349-355, Vol. 12, No. 5, May, 2002.

[6] L. Liu, E. Feig, “A block-based gradient descent search algorithm for block motion estimation in video coding”, IEEE Transactions on CSVT, vol. 6, no. 4, pp. 419-422, June 1996.

[7] J.Y. Tham, S.Ranganath, A.A.Kassim. A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation, IEEE Transactions on Circuits and Systems for Video Technology, 1998. 369-377.

[8] Z.Chen, J. Xu, Y. He, J. Zheng. Fast integer-pel and fractional- pel motion estimation for H.264/AVC, Journal of Visual Communication and Image Representation, vol. 17, Special Issue on Emerging H.264/AVC Video Coding Standard, 2006. 264–290.

[9] Zhenxing. CHEN, Yang SONG, Takeshi. IKENAGA, Satoshi. GOTO. A Macroblock Level Adaptive Search Range Algorithm For Variable Block Size Motion Estimation In H.264, Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication System. Nov.28-Dec.1, pp. 598-601.

[10] Mohammed Golam Sarwer, Q.M.Jonathan Wu, Adaptive Search Area Selection of Variable Block-Size Motion Estimation of H.264/AVC Video Coding Standard, IEEE International Symposium on Multimedia, vol. 11, 2009. 100-105.