

An Improved Adaptive Hexagon and Small Diamond Search

Fu Mo^(✉) and Kangshun Li

Department of Computer Science, Guangdong University of Science and Technology,
Dongguan, China
zdb317710990@sohu.com

Abstract. Motion estimation have a significant impact in H.265 video coding systems because it occupies a large amount of time in encoding. So the quality of motion search algorithm affect the entire encoding efficiency directly. In this paper, a novel search algorithm which utilizes an adaptive hexagon and small diamond search is proposed to overcome the drawbacks of the traditional block matching algorithm implemented in the most current video coding standards. The adaptive search pattern is chosen according to the motion strength of the current block. When the block is in active motion, the hexagon search provides an efficient search means; when the block is inactive, the small diamond search is adopted. Simulation results showed that our approach can speed up the search process with little effect on distortion performance compared with other adaptive approaches.

Keywords: Adaptive hexagon search · Diamond search · Motion estimation · Motion strength

1 Introduction

Now, H.265 is a popular video codec standard, motion estimation is the key technology in video coding [1]. Motion estimation directly affects the coding efficiency and the quality of image restoration. For the video image sequence, as between adjacent frame macroblock significant time correlation, by reducing the time redundancy, can improve the video coding efficiency. Motion estimation based on block matching algorithm is an effective method, has been adopted by many video coding standards. Motion estimation occupies whole encoder 60 %–80 % [2] computation. Largely determines the encoder speed quality and compression efficiency encoding [3–5].

Many fast block matching methods have been proposed to speed up the search process [6–8]. The classical methods include new three-step search (NTSS), four-step search (FSS), hexagon based search, diamond based search. Ghanbari presented the cross-search algorithm for motion estimation. This method experiences a logarithmic step, and checks four points in each step. Cheung and Po proposed a new search algorithm combining cross and diamond search together [9], which uses a cross search pattern as the initial step; then a large or small diamond search pattern was chosen as the subsequent step. Banh and Tan designed an adaptive dual-cross search method composed of three steps. This algorithm searches for three motion vectors, and chooses

the one with the least sum of absolute difference (SAD) as the initial search center. By comparing the SAD of the initial center with a pre-determined threshold, an early search termination strategy is used to reduce the computational cost. Finally, the global minimum is located through the dual-cross search scheme. The technique of block matching motion estimation has developed rapidly in recent years. The predictive motion vector field adaptive search technique (PMVFAST) was advanced by Tourapis et al. [11]; it uses several predictive motion vectors as the initial candidate vectors, and selects the best predictor to continue the diamond search. This method enhances the search speed greatly. Wong et al. [10] proposed an enhanced-PMVFAST (EPMVFAST) algorithm to further raise the speed. Unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [9] is a hierarchical approach consisting of four main steps, and most importantly, it has little loss in rate distortion performance compared to the full search method.

In this paper, we present a novel search algorithm, adaptive hexagon and small diamond search (AHSDS).

2 Early Termination Strategy

In H.265, the performance of the candidate search point is often evaluated by rate-distortion (RD) cost [10], which is composed of two parts:

$$J = SAD + \lambda_{motion}R \tag{1}$$

where SAD denotes the distortion measured as the sum of absolute differences between the original and referenced blocks, λ_{motion} is the Lagrange multiplier, and R represents the bits used to encode the motion information.

Assuming that the current block size is $M \times N$, SAD is the current block and reference block rate distortion difference, while the mean absolute error (Mean Absolute Difference) is the average difference of each pixel between the two blocks:

$$MAD(mv_x, mv_y) = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |F_t(x + m, y + n) - F_{t-1}(x + m + mv_x, y + n + mv_y)| \tag{2}$$

where F_t and F_{t-1} represent the current and the previous frames respectively, and (mv_x, mv_y) is the current motion vector. The motion search begins at the starting point. It comes from the predictive motion vector of the current block, and can be denoted as

$$mv_{st} = median(fmv_l, fmv_t, fmv_{tr}) \tag{3}$$

where fmv_l, fmv_t, fmv_{tr} are the final motion vectors of the adjacent left, top, respectively. As the search proceeds, the current search point progressively approaches the final optimal point. cmv and fmv denote the motion vectors at the current search point and the final search point, respectively. In most cases, the predictive motion vector mv_{st} is close to the final search point fmv . For the Foreman sequence, the mv_{st} of about 58 % of

blocks is exactly equal to the fmv . To reduce the search burden, the early termination strategy is applied when the MAD at the starting point is less than threshold Th_1 .

3 The Choice of Search Pattern

In our algorithm, the motion strength mos is defined as follows:

$$mos = MAD(cmv) \tag{4}$$

The motion strength, a dynamic variant, does not represent the absolute magnitude of the motion vector cmv , but reflects the deviation of cmv from fmv . When the block is at the starting point, i.e., $cmv = mv_{st}$, the value of mos is relatively high; when the current search point is close to the final optimal point, mos decreases gradually.

The choice of the search pattern is decided by the comparison between the motion strength and the threshold. A fixed threshold is not suitable for all sequences. Too large or too small a threshold will unnecessarily increase search costs, and may even lead to an error in the final optimal point. An adaptive threshold is adopted in our approach to meet the changes of sequences with various motion characters and different video quality. Tourapis et al. [11] present several forms of adaptive threshold. In our algorithm, the threshold Th_k is computed as follows:

$$\begin{cases} Th_k = a_k \times CMAD + b_k, \\ CMAD = mean(MMAD_l, MMAD_t, MMAD_{tr}) \end{cases} \tag{5}$$

where $MMAD_l, MMAD_t, MMAD_{tr}$ are the minimum MAD of the left, top, and top-right blocks respectively, and $CMAD$ is the mean value of $MMAD_l, MMAD_t$, and $MMAD_{tr}$. a_k, b_k are both constant, K is the number of search. To prevent the threshold Th_k from being too large or too small, we use some limiting parameters to restrict the threshold:

$$Th_k = \min (\max (a_k \times CMAD + b_k, T_{jk}), T_{mk}) \tag{6}$$

where T_{mk} and T_{jk} are the upper and lower bounds for threshold Th_k , respectively. T_{mk} and T_{jk} are set usually according to the mean distortion of the previous frame.

4 The Design of AHSDS

The proposed algorithm AHSDS can be described as follows:

Step 1: Determine the activity of the current block.

In our algorithm, we use the MAD of the starting search point to describe the motion activity of the current block according to:

$$Th_k = \min (\max (a_k \times CMAD + b_k, T_{jk}), T_{mk}) \tag{7}$$

where Th_2 is a threshold which can be adaptively set according to the minimum MAD of the adjacent blocks, and mv_{st} is the motion vector of the starting point. If Equation a is satisfied, the current block is active; otherwise, it is inactive.

Step 2: The choice of the search pattern.

If the MAD at the starting point is smaller than Th_1 , the search terminates immediately. If the block is active (which means there is a high probability that the block has a large motion vector), start with the hexagon search, and go to Step 3; otherwise, go to Step 4.

Step 3: Perform the hexagon search.

Repeat the hexagon search until the best point is at the center of the hexagon. Select the sub-optimal point among the seven points in Fig. 1.

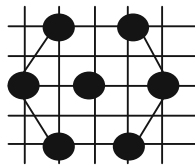


Fig. 1. Hexagon seven point distribution

If the center point A and the sub-optimal point B are in a horizontal direction, choose the newest and best point among point A and its three neighboring points as shown in Fig. 2(a). Similarly, Fig. 2(b) depicts the case where the points A and B are in a diagonal direction. Go to Step 5.

Step 4: Perform the small diamond search recursively until the best point with the minimum RD cost is at the center of the small diamond.

Step 5: Obtain the final optimal search point, and the whole search process then terminates.

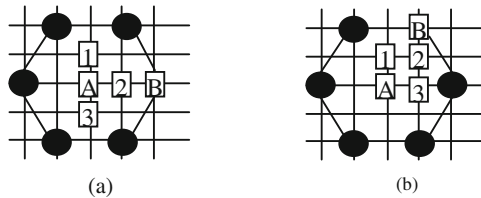


Fig. 2. Different second optimal point of the three adjacent point position

In our algorithm, the motion vector at the starting search point comes from the predictive motion vector pmv , which is the median motion vector of three spatially adjacent blocks (the left, top, and top-right blocks). This simplifies the search procedure greatly. For a large majority of blocks, the final motion vector fmv is close to pmv , which means that small diamond is the main search pattern.

To analyze the similarity of motion vectors pmv and fmv , we calculate the probability P_n that the point (pmv_x, pmv_y) appeared in a square area, centered at (pmv_x, pmv_y) , by

$$P_n = P \left\{ |fmv_x - pmv_x| \leq n, |fmv_y - pmv_y| \leq n \right\} \quad (8)$$

where pmv_x and pmv_y denote the x- and y-component of the motion vector fmv , respectively. P_n is worked out with a statistical method. For the Foreman sequence, when n equals 2, the corresponding probability is about 0.83. It does not mean that all these blocks have small final motion vectors. If the current block and the adjacent blocks all have similar large motion vectors, the small diamond search could still be used. Short-distance searches often occur whenever the final motion vector is close to the predictive motion vector, regardless of the absolute magnitude of the final motion vector. In contrast, long-distance searches usually appear in cases where the final motion vector of the current block is distinct from those of the adjacent blocks.

5 Implementation and Simulation

The proposed algorithm AHSDS was simulated on the platform H.265 reference software JM 14.2. We used different CIF video sequences, which contain various motion content, to test the performance of our algorithm. The search window was $[-16, 16]$. The number of reference frames was set to 5. The frame structure in our experiment was IPPPP..., and our proposed algorithm was compared with UMHexagonS (UMHS), simplified UMHexagonS (SUMH), and EPMVFAST in the same environment.

For different block matching algorithms, the number of search points (NSP) required for each block-matching was used to measure the computational efficiency. The average PSNR per frame is an important factor in evaluating the distortion performance. Different block matching approaches are compared in the following aspects.

A. Search threshold selection

Assume the threshold Th can be denoted as

$$Th_1 = k_1 \times CMAD \quad (9)$$

In Fig. 3., when $|rmv|$ is equal to zero, the statistical ratio was near 1.3. We adopted the conservative threshold. The parameter k_1 was usually set to not greater than 1.0. The choice of search pattern is determined by Th_2 . The small diamond search can achieve a smaller NSP, but it is not fit for long-distance searches because it might become trapped in a local optimal point and lead to the decrease of PSNR. In contrast, the hexagon search consumes more search points for short-distance searches.

A small diamond is composed of one center point and four edge points. When the minimum RD cost is located at one of the four edge points, the search will correctly detect the optimal point. When the distance between the real optimal point and the starting point is greater than 1, however, the search may not find the real optimal point successfully. Therefore, the hexagon search should be chosen when $|rmv|$ is greater than 1.0. The suggested range for k_2 is $[2.0, 2.6]$.

B. Number of search points (NSP)

With the increase of QP , the $PSNR$ of the re-constructed frame dropped off progressively. Some details might be lost in the reference frame, affecting the number of search points. We tested each sequence on five different QP (24, 28, 32, 34, and 36), 100 frames for each QP , and then calculated the mean value of their NSPs and denoted it as an integer. The average NSPs for different video sequences and different methods are listed in Table 1.

Table 1. Three different QP of the average search points in different sequence and methods

Average search points Sequence type	$QP=24$		$QP=28$		$QP=36$	
	<i>UMHS</i>	<i>SUMH</i>	<i>UMHS</i>	<i>SUMH</i>	<i>EPMVFAST</i>	<i>AHSDS</i>
Foreman	22	16	18	13	9	7
Coastguard	36	20	32	18	10	8
Flower	30	22	26	20	9	7
Mobile	34	25	30	19	8	6
Football	43	18	41	25	13	10
Silent	14	16	12	10	7	5

From Table 1, AHSDS generally had the lowest number of search points compared with other methods. For the different sequences, Silent had fewer average search points than Football, for example, because the latter had more blocks in active motion. Figure 4 plots the NSP curves with EPMVFAST and AHSDS for the Foreman sequence when QP equals 28. We chose 100 frames for testing. The figure shows that the AHSDS method can speed up the process of block matching compared with EPMVFAST. The main reason lies in the following two aspects:

- For EPMVFAST, several predictors are checked to obtain the optimal starting point, while in AHSDS, only the predictive motion vector is chosen as the starting point. On the other hand, in most cases, the final motion vector has a high level of similarity with the predictive one.

6 Conclusions

A new, fast approach AHSDS for block motion estimation is considered for the coding standard H.265. Using an initial classification on motion strength of the block according to its MAD, different search patterns were chosen to determine the optimal point with

the minimum RD cost. When the block is in high activity, hexagon search is the preferred scheme. If the best point is located at the center of the hexagon, three points around the center are checked to obtain the optimal solution. Simulation was conducted with different methods and video sequences. Compared with EPMVFAST, AHSDS can enhance the search speed. Our approach is suitable for hardware realization due to its simplicity in algorithm complexity, and can be applied to real-time encoding systems owing to its small number of search points.

Acknowledgment. This project is supported by Science and Technological Program for Dongguan's Higher Education, Science and Research, and Health Care Institutions. Project number: 2012108102028.

References

1. Ugur, K., Alshin, A., Alshina, E., Bossen, F.: Motion compensated prediction and interpolation filter design in H.265/HEVC. *IEEE J. Mag.* **7**(6), 946–956 (2013). Nokia Corp., Tampere, Finland
2. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. Image Process.* **9**(2), 287–290 (2000)
3. Han, C.D., Liu, J.L., Xiang, Z.Y.: An adaptive fast search algorithm for block motion estimation in H.264. *J. Zhejiang Univ. Sci. C* **11**(8), 637–644 (2010)
4. Sourabh, R., Neeta, T.: Hexagonal based search pattern for motion estimation in H.264/AVC. *World Comput. Sci. Inf. Technol. J.* **1**(4), 162–166 (2011)
5. Yan, Y.X., Meng, S.L.: A new hybrid search scheme for video motion estimation. *J. Convergence Inf. Technol.* **6**(3), 106–112 (2011)
6. Goel, S., Ismail, Y., Bayoumi, M.A.: Adaptive search window size algorithm for fast motion estimation in H.264/AVC standard. In: *The 48th Midwest Symposium on Circuits and System*, vol. 2, pp. 1557–1560 (2005)
7. Huang, Y.W., Hsieh, B.Y., Chien, S.Y., et al.: Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **16**(4), 507–522 (2006)
8. Kim, S.E., Han, J.K., Kim, J.G.: An efficient scheme for motion estimation using multireference frames in H.264/AVC. *IEEE Trans.* **8**(3), 457–466 (2006)
9. Zhu, C., Lin, X., Chua, L.P.: Hexagon-eased search pattern for fast block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.* **12**(50), 349–355 (2002)
10. Wong, H.M., Au, O.C., Ho, C.W., et al.: Enhanced predictive motion vector field adaptive search technique (E-PMVFAST) -based on future mv prediction. In: *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 1–4 (2005)
11. Tourapis, A.M., Au, O.C., Liou, M.L.: Fast block-matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST). In: *ISO/IEC JTC1/SC29/WG11 MPEG99/m5866*, Noordwijkerhout, The Netherlands, March 2000
12. Cheung, C.H., Po, L.M.: Novel cross-diamond-hexagonal search algorithms for fast block motion estimation. *IEEE Trans. Multimedia* **7**(1), 16–22 (2005)