Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An adaptive fast search algorithm for block motion estimation in H.264*

Cong-dao HAN[†1,2], Ji-lin LIU[1,2], Zhi-yu XIANG[1,2]

(*[1]Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China*)

(*[2]Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China*)

[†]E-mail: hancongdao@163.com

Received Mar. 14, 2009; Revision accepted June 4, 2009; Crosschecked July 1, 2010

Abstract: Motion estimation is an important issue in H.264 video coding systems because it occupies a large amount of encoding time. In this paper, a novel search algorithm which utilizes an adaptive hexagon and small diamond search (AHSDS) is proposed to enhance search speed. The 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.

Key words: Adaptive hexagon and small diamond search (AHSDS), Search pattern, Mean absolute difference (MAD), Optimal point

doi:10.1631/jzus.C0910561 **Document code:** A **CLC number:** TN919.81

1 Introduction

Multimedia technology has been widely used in many areas of modern communication, such as video meetings, visual telephone, and telemedicine. It has developed rapidly, especially in the area of mobile video telecommunications. Huge amounts of voice and video information are transmitted in communication networks. To remove the temporal and spatial information redundancy of video sequences, video compression as an important and fundamental research topic has received attention from many researchers over the last 20 years.

Recently, H.264 has become a popular video compression standard (ITU-T and ISO/IEC, 2003). Saving transmission bandwidth and enhancing video quality are the two essential goals of video compression. In H.264, block motion estimation occupies a large fraction of the total encoding time. For full search, the fraction can be as large as 60%. In contrast, some real-time encoding systems like telemedicine and video surveillance have rigid restrictions on encoding time and the quality of the reconstructed frames. Consequently, researchers have urgently sought efficient search methods and improved the distortion performance.

Many fast block matching methods have been proposed to speed up the search process. The classical methods include new three-step search (NTSS) (Li *et al*., 1994), four-step search (FSS) (Po and Ma, 1996), hexagon based search (Zhu *et al*., 2002; Tsai and Pan, 2004), and diamond based search (DS) (Tham *et al*., 1998; Zhu and Ma, 2000). Ghanbari (1990) presented the cross-search algorithm for motion estimation. This method experiences a logarithmic step, and checks four points in each step. Cheung and Po (2002) proposed a new search algorithm combining cross and diamond search together, which uses a cross-search pattern as the initial step; then a large or small diamond search pattern was chosen as the subsequent

* Project (Nos. 60505017 and 60534070) supported by the National Natural Science Foundation of China

step. Banh and Tan (2004) devised 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*. (2000); 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*. (2005) proposed an enhanced-PMVFAST (EPMVFAST) algorithm to further raise the speed. Unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) (Chen *et al*., 2006) 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.

Approaches utilizing various pieces of block information, such as temporal and spatial correlation of motion vectors (Nam *et al*., 2000) and predictive search area (Chung and Chang, 2003), have been advanced to improve the search efficiency. In addition, several new search schemes have been proposed (Jung *et al*., 2002; Tu *et al*., 2003), for example, adjustable partial distortion search (Cheung and Po, 2003), adaptive search (Nie and Ma, 2002; Han and Chun, 2003; Huang *et al*., 2005), fuzzy search (Roan and Chen, 2000), and probabilistic approach (Lengwehasarit and Ortega, 2001).

In this paper, we present a novel search algorithm, adaptive hexagon and small diamond search (AHSDS), which is a combination of two classical search methods. Firstly, we choose the search pattern according to the motion strength of the current block. When the block is in active motion, the hexagon search is a good alternative because it covers a wide search area. Three neighboring points around the center are examined to determine the accurate optimal point when the minimal distortion is at the center of the hexagon. When the block is inactive, only the small diamond search is adopted.

2 Early termination strategy

In H.264, the performance of the candidate search point is often evaluated by rate-distortion (RD) cost (Wong *et al*., 2005), which is composed of two parts:

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

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

Assuming the current block size is $M\times N$, SAD reflects the difference between the current block and the referenced block, while MAD (mean absolute difference) reveals the average difference of each pixel between the two blocks (Wong *et al*., 2005):

$$\begin{aligned}\mathrm{MAD}(\mathrm{mv}_x, \mathrm{mv}_y) = \frac{1}{M\times N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\big|F_t(x+m, y+n)\\ -F_{t-1}(x+m+\mathrm{mv}_x, y+n+\mathrm{mv}_y)\big|,\end{aligned} \tag{2}$$

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

$$\mathbf{mv}_{\mathrm{st}} = \mathrm{median}(\mathbf{fmv}_{\mathrm{l}}, \mathbf{fmv}_{\mathrm{t}}, \mathbf{fmv}_{\mathrm{tr}}), \tag{3}$$

where $\mathbf{fmv}_{\mathrm{l}}$, $\mathbf{fmv}_{\mathrm{t}}$, and $\mathbf{fmv}_{\mathrm{tr}}$ are the final motion vectors of the adjacent left, top, and top-right blocks, 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 $\mathbf{mv}_{\mathrm{st}}$ is close to the final search point **fmv**. For the Foreman sequence, the $\mathbf{mv}_{\mathrm{st}}$ 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 approach, motion strength mos is defined as the MAD of the block with the motion vector **cmv**:

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

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., $\mathbf{cmv}=\mathbf{mv}_{\mathrm{st}}$, 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 determined 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*. (2000) listed several forms of adaptive threshold. In our approach, the threshold Th_k is calculated as follows:

$$\begin{cases} \mathrm{Th}_k = a_k \times \mathrm{CMAD} + b_k, \\ \mathrm{CMAD} = \mathrm{mean}(\mathrm{MMAD}_{\mathrm{l}}, \mathrm{MMAD}_{\mathrm{t}}, \mathrm{MMAD}_{\mathrm{tr}}), \end{cases} \tag{5}$$

where $\mathrm{MMAD}_{\mathrm{l}}$, $\mathrm{MMAD}_{\mathrm{t}}$, and $\mathrm{MMAD}_{\mathrm{tr}}$ are the minimum MAD of the left, top, and top-right blocks respectively, and CMAD is the mean value of $\mathrm{MMAD}_{\mathrm{l}}$, $\mathrm{MMAD}_{\mathrm{t}}$, and $\mathrm{MMAD}_{\mathrm{tr}}$.

To prevent the threshold Th_k from being too large or too small, we use some limiting parameters to restrict the threshold:

$$\mathrm{Th}_k = \min\left(\max(a_k \times \mathrm{CMAD} + b_k, T_{jk}), T_{mk}\right), \tag{6}$$

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

4 The proposed algorithm

The proposed approach 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

$$\mathrm{MAD}(\mathbf{mv}_{\mathrm{st}}) > \mathrm{Th}_2, \tag{7}$$

where Th_2 is a threshold which can be adaptively set according to the minimum MAD of the adjacent blocks, and $\mathbf{mv}_{\mathrm{st}}$ is the motion vector of the starting point.

If Eq. (7) is satisfied, the current block is active; otherwise, it is inactive.

Step 2: Choose 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.

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. 2a. Similarly, Fig. 2b depicts the case where the points A and B are in a diagonal direction. Go to Step 5.

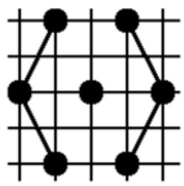

Fig. 1 The distribution of seven points in hexagon

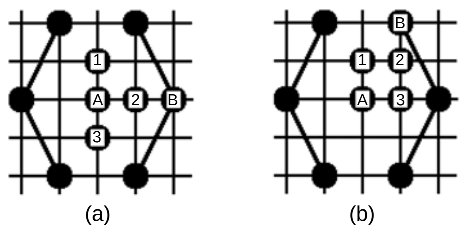

Fig. 2 The position of three neighboring points for different sub-optimal points

Center point A and sub-optimal point B are in a horizontal direction in (a) and a diagonal direction in (b)

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.

Our algorithm integrates the characteristics of hexagon search and small diamond search. The former can perform a rapid scan for long-distance searches, whereas the latter consumes few points for short-distance searches. The diagram of our algorithm is depicted in Fig. 3. The starting point comes from the predictive motion vector of the current block.

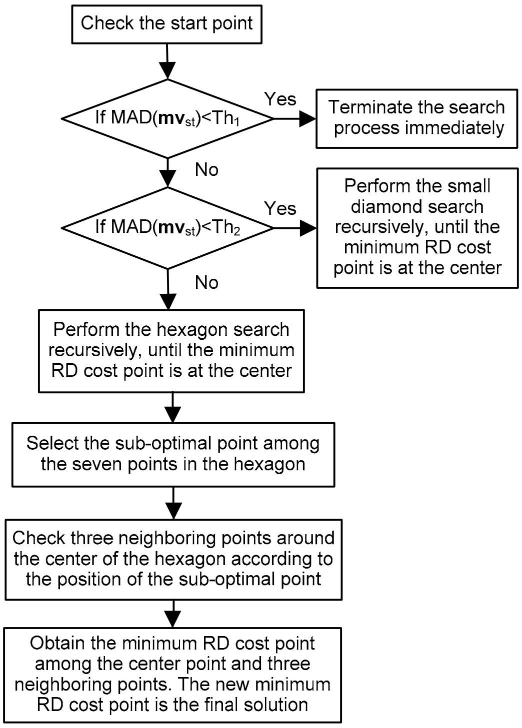

Fig. 3 Flowchart of the proposed adaptive hexagon and small diamond search (AHSDS) algorithm

To further illustrate the process of the AHSDS algorithm, an example of block matching is depicted in Fig. 4.

In our approach, 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.

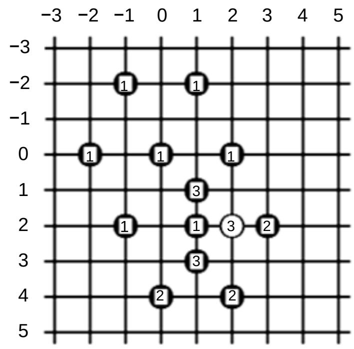

Fig. 4 An example of search steps using the adaptive hexagon and small diamond search (AHSDS) approach
The search process involves three steps: first it checks the starting point (0, 0) to determine the motion strength of the block. The hexagon search pattern is chosen because the MAD of the starting point is greater than Th_2. In the second step, point (1, 2) has the minimal RD cost; meanwhile, (3, 2) is a sub-optimal point. In the third step, the process checks three neighboring points around point (1, 2), and compares their RD costs with that of point (1, 2). Point (2, 2) has the minimal RD cost and becomes the final optimal point

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

$$P_n = P\left\{\left|\mathrm{fmv}_x - \mathrm{pmv}_x\right| \le n,\ \left|\mathrm{fmv}_y - \mathrm{pmv}_y\right| \le n\right\}, \quad (8)$$

where fmv_x and fmv_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 Experiments and results

The proposed algorithm AHSDS was simulated on the platform H.264 reference software JM14.2. We used different CIF video sequences, which contain various motion content, to test the performance of our method. 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.

5.1 Threshold setting

In our approach, threshold Th_1 is calculated as follows:

$$\begin{cases}\text{Th}_1 = \min(1.0\times\text{CMAD}, T_{m1}), \\ T_{m1} = 1.5\times\text{PMAD},\end{cases} \tag{9}$$

where CMAD is the mean value of MMAD_l, MMAD_t, and MMAD_tr, which come from the adjacent left, top, and top-right blocks, while PMAD is the mean MMAD of blocks in the previous frame.

Similarly, Th_2 can be expressed as

$$\begin{cases}\text{Th}_2 = \min\left(\max(2.0\times\text{CMAD}, T_{j2}), T_{m2}\right), \\ T_{j2} = 1.0\times\text{PMAD},\ T_{m2} = 3.0\times\text{PMAD}.\end{cases} \tag{10}$$

We did not set a lower bound on Th_1 because it is a threshold for early termination. To analyze the relationship between the magnitude of the motion vector and the MAD value of each block, we recorded the initial MAD value when the block was at the starting search point. Because the starting point comes from the predictive motion vector of the current block, we define the new motion vector **rmv** as follows:

$$\begin{cases}\mathbf{rmv} = \mathbf{fmv} - \mathbf{pmv}, \\ |\mathbf{rmv}| = \sqrt{\text{rmv}_x^2 + \text{rmv}_y^2}.\end{cases} \tag{11}$$

For all blocks that had the same magnitude |**rmv**|, we calculated their ratios of MAD to CMAD, and worked out the mean of these ratios. Fig. 5 shows the relationship between the ratio and the magnitude of motion vector **rmv** for the Foreman sequence. The lowest four points had different |**rmv**| values 0, 1, $\sqrt{2}$, and 2.

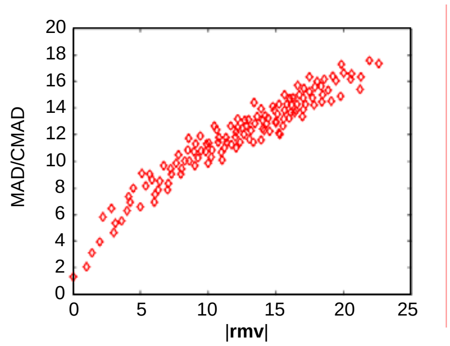

Fig. 5 The ratios with the change of |rmv| for the Foreman sequence using a statistical method
MAD: mean absolute difference; CMAD: the mean value of the minimum MADs of three adjacent blocks; |**rmv**|: magnitude of motion vector **rmv**

The ratio increased with the increase of |**rmv**|. Note that Fig. 5 is plotted with a statistical approach. The blocks with low |**rmv**| values occupied a large proportion. For the Foreman sequence, about 86% of blocks had the common property that their |**rmv**| values are not greater than 3.0. In other words, blocks with a large magnitude had only a small proportion, which leads to the randomness of statistical results.

In our approach, we adopted a conservative principle to calculate the thresholds. To demonstrate the determination of thresholds and bounds, we took the Foreman sequence as the instance to describe the details.

5.1.1 Threshold analysis

Assume the threshold Th_1 can be denoted as

$$\text{Th}_1 = k_1\times\text{CMAD}. \tag{12}$$

In Fig. 5, 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].

5.1.2 Bound analysis

Assume the upper bound T_{m1} can be expressed as

$$T_{m1} = m_1 \times \text{PMAD}. \tag{13}$$

The restrictions on threshold Th_1 can effectively prevent the local threshold from being too large, and help to stabilize the algorithm. PMAD can be taken as CMAD on the average. Because the average ratio was about 1.3 when |**rmv**| equals zero, m_1 was usually set to not greater than 1.5.

The lower bound T_{j2} and upper bound T_{m2} for threshold Th_2 have the corresponding parameters j_2 and m_2. They should not be assigned high values because the proportion of small diamond search increases with the increase of these two parameters. j_2 and m_2 were usually set not to exceed 1.5 and 3.0, respectively.

The setting of bounds is a supplement to the threshold. It helps to improve the robustness of the algorithm. When the thresholds are assigned relatively high or low values, our approach could still yield a stable performance.

To test the stability of the algorithm, we conducted experiments for two cases with the Foreman sequence when the quantization parameter (QP) equals 28: (1) k_2=2.0, m_2=3.0; (2) k_2=3.0, m_2=4.0. The difference of PSNR was only 0.01 dB.

5.2 Number of search points (NSP)

With the increase of QP, the PSNR of the reconstructed 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 QPs (28, 30, 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 Average number of search points (NSP) at five QPs of 28, 30, 32, 34, and 36 with 100 frames for each QP, for different video sequences and methods

Sequence	Average NSP			
	UMHS	SUMH	EPMVFAST	AHSDS
Foreman	18	13	9	7
Coastguard	32	18	10	8
Flower	26	20	9	7
Mobile	30	19	8	6
Football	41	25	13	10
Silent	12	10	7	5

UMHS: UMHexagonS; SUMH: simplified UMHexagonS; EPMVFAST: enhanced predictive motion vector field adaptive search technique; AHSDS: adaptive hexagon and small diamond search

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. Fig. 6 plots the NSP curves with EPMVFAST and AHSDS for the Foreman sequence when QP equals 32. We chose 100 frames for testing. The figure shows that the AHSDS method can speed up the process of block matching compared with EPMVFAST.

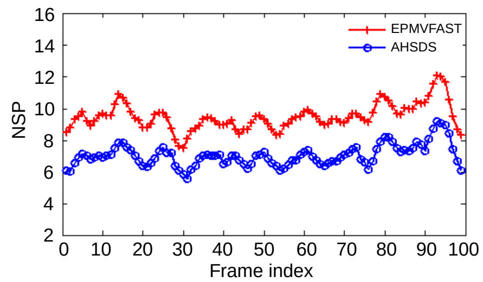

Fig. 6 Comparison of the number of search points (NSP) using EPMVFAST (enhanced predictive motion vector field adaptive search technique) and AHSDS (adaptive hexagon and small diamond search) for the Foreman sequence at QP=32

The main reason lies in the following two aspects:

1. 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.

2. For long-distance searches, EPMVFAST still uses DS as the search pattern; in contrast, AHSDS adopts the hexagon search. This helps to decrease the search cost.

5.3 PSNR

The average PSNR performance of each frame is summarized in Table 2 for different sequences.

Table 2 Average PSNR of each frame for AHSDS and UMHS

Sequence	Average PSNR (dB)			
	QP=28		QP=36	
	UMHS	AHSDS	UMHS	AHSDS
Foreman	40.53	40.52	37.09	37.07
Coastguard	41.01	41.01	37.49	37.48
Flower	36.46	36.46	31.19	31.18
Mobile	35.96	35.95	30.97	30.96
Football	39.98	39.97	35.75	35.73
Silent	39.18	39.18	35.61	35.60

UMHS: UMHexagonS; AHSDS: adaptive hexagon and small diamond search

The proposed AHSDS had little difference in PSNR compared with UMHS. Fig. 7 shows the PSNR curves using these two approaches, as conducted on the Silent sequence when QP equals 32.

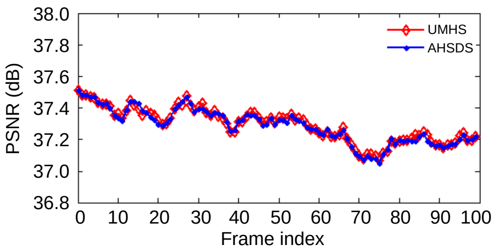

Fig. 7 PSNR performance comparison using UMHS (unsymmetrical-cross multi-hexagon-grid search) and AHSDS (adaptive hexagon and small diamond search) for the Silent sequence at QP=32

6 Conclusions

A new, fast approach AHSDS for block motion estimation is considered for the coding standard H.264. 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.

References

Banh, X.Q., Tan, Y.P., 2004. Adaptive dual-cross search algorithm for block-matching motion estimation. *IEEE Trans. Consum. Electron.*, **50**(2):766-775. [doi:10.1109/TCE.2004.1309460]

Chen, Z., Zhou, P., He, Y., 2006. Fast integer-pel and fractional-pel motion estimation for H.264/AVC. *J. Vis. Commun. Image Represent.*, **17**(2):264-290. [doi:10.1016/j.jvcir.2004.12.002]

Cheung, C.H., Po, L.M., 2002. A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, **12**(12):1168-1177. [doi:10.1109/TCSVT.2002.806815]

Cheung, C.H., Po, L.M., 2003. Adjustable partial distortion search algorithm for fast block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, **13**(1):100-110. [doi:10.1109/TCSVT.2002.808091]

Chung, K.L., Chang, L.C., 2003. A new predictive search area approach for fast block motion estimation. *IEEE Trans. Image Process.*, **12**(6):648-652. [doi:10.1109/TIP.2003.812756]

Ghanbari, M., 1990. The cross-search algorithm for motion estimation. *IEEE Trans. Commun.*, **38**(7):950-953. [doi:10.1109/26.57512]

Han, K., Chun, B., 2003. Adaptive Hexagon Search Pattern for Block Motion Estimation. Proc. IEEE Int. Conf. on System, Man and Cybernetics, **2**:1406-1409.

Huang, S.Y., Cho, C.Y., Wang, J.S., 2005. Adaptive fast block-matching algorithm by switching search patterns for sequences with wide-range motion content. *IEEE Trans. Circ. Syst. Video Technol.*, **15**(11):1373-1384. [doi:10.1109/TCSVT.2005.856931]

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

Jung, S.M., Shin, S.C., Baik, H., Park, M.S., 2002. Efficient multilevel successive elimination algorithms for block matching motion estimation. *IEE Proc.-Vis. Image Signal Process.*, **149**(2):73-84. [doi:10.1049/ip-vis:20020189]

Lengwehasarit, K., Ortega, A., 2001. Probabilistic partial-distance fast matching algorithms for motion estimation.

IEEE Trans. Circ. Syst. Video Technol., **11**(2):139-152. [doi:10.1109/76.905981]

Li, R., Zeng, B., Liou, M.L., 1994. A new three-step search algorithm for block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, **4**(4):438-442. [doi:10.1109/76.313138]

Nam, J.Y., Seo, J.S., Kwak, J.S., Lee, M.H., Ha, Y.H., 2000. New fast search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector. *IEEE Trans. Consum. Electron.*, **46**(4):934-942. [doi:10.1109/30.920443]

Nie, Y., Ma, K.K., 2002. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Trans. Image Process.*, **11**(12):1442-1449. [doi:10.1109/TIP.2002.806251]

Po, L.M., Ma, W.C., 1996. A novel four-step search algorithm for fast block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, **6**(3):313-317. [doi:10.1109/76.499840]

Roan, Y.T., Chen, P.Y., 2000. A fuzzy search algorithm for the estimation of motion vectors. *IEEE Trans. Broadcast.*, **46**(2):121-127. [doi:10.1109/11.868927]

Tham, J.Y., Ranganath, S., Ranganath, M., Kassim, A.A., 1998. A novel unrestricted center-biased diamond search algorithm for block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, **8**(4):369-377. [doi:10.1109/76.709403]

Tourapis, A.M., Au, O.C., Liou, M.L., 2000. Fast Block-Matching Motion Estimation Using Predictive Motion Vector Field Adaptive Search Technique (PMVFAST). ISO/IEC JTC1/SC29/WG11 MPEG2000/m5866. Noordwijkerhout, NL.

Tsai, T.H., Pan, Y.N., 2004. A Novel Predict Hexagon Search Algorithm for Fast Block Motion Estimation on H.264 Video Coding. Proc. IEEE Conf. on Asia-Pacific Circuits and Systems, **1**:609-612. [doi:10.1109/APCCAS.2004.1412836]

Tu, Y.K., Yang, J.F., Shen, Y.N., Sun, M.T., 2003. Fast Variable Size Block Motion Estimation Using Merging Procedure with an Adaptive Threshold. Proc. IEEE Int. Conf. on Multimedia and Expo, **2**:789-792. [doi:10.1109/ICME.2003.1221735]

Wong, H.M., Au, O.C., Ho, C.W., Yip, S.K., 2005. Enhanced Predictive Motion Vector Field Adaptive Search Technique (E-PMVFAST)-Based on Future MV Prediction. Proc. IEEE Int. Conf. on Multimedia and Expo, p.1-4. [doi:10.1109/ICME.2005.1521524]

Zhu, C., Lin, X., Chau, L.P., 2002. Hexagon-based search pattern for fast block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, **12**(5):349-355. [doi:10.1109/TCSVT.2002.1003474]

Zhu, S., Ma, K.K., 2000. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. Image Process.*, **9**(2):287-290. [doi:10.1109/83.821744]