SIViP (2010) 4:123–128
DOI 10.1007/s11760-009-0104-9

ORIGINAL PAPER

Fast motion estimation using modified orthogonal search algorithm for video compression

Shilpa P. Metkar · Sanjay N. Talbar

Received: 7 November 2007 / Revised: 19 December 2008 / Accepted: 28 December 2008 / Published online: 22 January 2009

Abstract This paper presents a novel method named as Modified Orthogonal Search Algorithm (MOSA) for the block based motion estimation. Recently fast search algorithm for video coding using Orthogonal Logarithmic Search Algorithm (OSA) has been proposed by Soongsathitanon et al. (IEEE Trans Consum Electron 51(2):552–559, 2005). We introduce the center biased search point pattern for the estimation of small motions and a half way stop technique to reduce the computational complexity in the existing OSA. This feature improves the speed performance of the algorithm by 80% as compared to the Full Search Algorithm, 50% over the Three Step Search algorithm and 2% faster than the OSA. However, the Mean Square Error and Signal to Noise Ratio did not show significant deviation from the orthogonal logarithmic search algorithm and the three step search algorithm. The experimental results based on the number of video sequences were presented to demonstrate the advantages of proposed motion estimation technique.

Keywords Block-Matching · Motion estimation · Orthogonal logarithmic search

1 Introduction

Recent years have witnessed a vital role played by video compression in data storage and transmission. Video compression involves the removal of the spatial and temporal (interframe) redundancies. Inter frame compression exploits the similarities between successive frames, known as temporal redundancy, to reduce the volume of data required to describe the sequence. Several inter-frame compression methods [2], of various degrees of complexity are presented in the literature such as sub-sampling coding, difference coding, block based difference coding and motion compensation.

The motion compensation prediction is commonly used in most of the video codec [2,3]. In order to carry out motion compensation the motion of the moving objects have to be estimated first, which is called as motion estimation [4]. Motion compensated prediction assumes that the current picture can be locally modeled as a translation of the pictures of some previous time. Block matching methods due to their less computational complexity are the most popular motion estimation methods which are adopted by various video coding standards such as MPEG-1 and MPEG-2 [17]. In block matching method, the current frame is divided into sub-blocks of $N \times N$ pixels. Each sub block is predicted from the previous or future frame, by estimating the amount of motion of the sub-block called as motion vector during the frame time interval. The video coding syntax specifies how to represent the motion information for each sub-block, but it does not however specify how such vectors are to be computed. The motion vector is obtained by finding a Block Distortion Measure (BDM) function which calculates the mismatch between the reference and the current block [5]. To locate the best matched sub-block which produces the minimum mismatch error, we need to calculate distortion function at several locations in the search range.

One of the first algorithm developed for block based motion estimation was the full search algorithm (FSA) or exhaustive search algorithm (ESA), which evaluates the block distortion measure (BDM) function at every possible

S. P. Metkar (✉)
College of Engineering, An Autonomous Institute of Government of Maharashtra, Pune, India
e-mail: shilpa_metkar@yahoo.com

S. N. Talbar
SGGS Institute of Engineering and Technology, Nanded, Maharashtra, India
e-mail: sntalbar@yahoo.com

pixel locations in the search area [5]. Although this algorithm is the best in terms of quality of the predicted frame and simplicity, it is computationally intensive. In the past two decades, several fast search methods for motion estimation have been introduced to reduce the computational complexity of block matching, for examples, two dimensional logarithmic search (LOGS) [5], three-step search (3SS) [6], four step search (4SS) [7]. As these algorithm utilized uniformly allocated search points in their first step, they all can achieve substantial computational reduction with a drawback of modest estimation accuracy degradation. Search with a large pattern in the first step is inefficient for the estimation of small motion since it will be trapped into a local minimum. In real world video sequences, the distortion of motion vectors is highly center biased which results in a center biased motion vector distribution instead of a uniform distribution [9]. This indicates that the probability increases to get the global minimum at the center region of the search window. To make use of this characteristic, center biased block matching algorithms were then proposed with search points much closer to the center which improves the average prediction accuracy especially for the slow motion sequences. Well known examples of this category are New Three-Step Search (N3SS) [8], Advanced Center Biased Three Step Search (ACBTSS) [9], Block-Based Gradient Descent Search (BBGDS) [10], Diamond Search (DS) [11], Cross Diamond Search (CDS) [12], Hexagon-Based Search (HS) [13], etc. There is considerable research effort being applied to the subject of motion estimation [14–18].

Among these algorithms, 3SS has become most popular one for low bit rate application owing to its simplicity and effectiveness. However 3SS and recently proposed OSA uses uniformly allocated searching points in their first step which becomes inefficient for the estimation of small motions since it gets trapped into local minimum. Having observed this problem, a Modified Orthogonal Search Algorithm (MOSA) is proposed in this paper. The proposed MOSA searches the additional central eight points in order to favor the characteristics of center biased motion. To speed up block matching process, MOSA terminated at intermediate step instead of adapting the entire steps of the algorithm referred as half way stop technique. The rest of the paper is organized as follows; Sect. 2 presents the OSA, then proposed MOSA is described in Sects. 3, and 4 brings about experimental results of our proposed method in comparison with FSA, 3SS and OSA. The experimental results are concluded in Sect. 5.

2 Orthogonal logarithmic search algorithm

The OSA proposed by [1] has the pairs of horizontal and vertical search points with a logarithmic decrease in the successive step size (‘*st*’). After the step size *st* has been initialized at $st = \lceil d/2 \rceil$ where d is the maximum motion displacement, the center block and two candidate blocks on either side of the x-axis at a distance (‘$\pm$ *st’)* are compared to the target block. The position of minimum distortion (BDM) will become the center of the vertical stage. During the vertical stage, two search points above and below the new center are examined and the values of the distortion function at these three positions are compared. The position with the minimum distortion will become the center of the next stage. After a horizontal and vertical iteration, the step size is halved, if it is greater than one the algorithm proceeds with another horizontal and vertical iteration otherwise it halts and declares one of the positions from the vertical stage the best match for the target block. Thus OSA estimate motion vectors by searching the window orthogonally with logarithmic reduction in the step size. The search patterns for OSA are shown in Fig. 1.

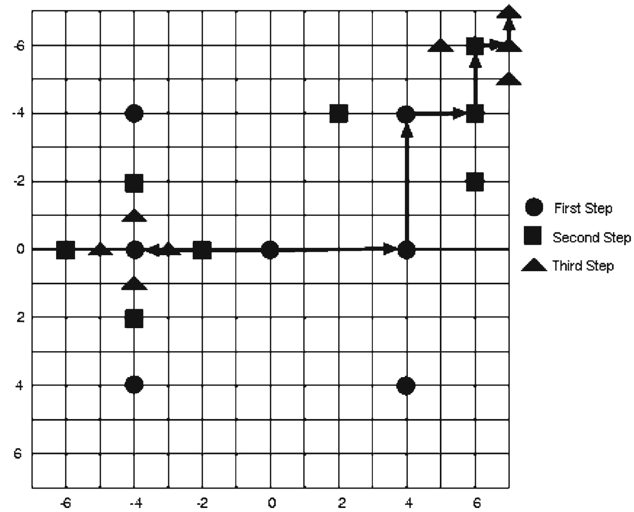

Fig. 1 Orthogonal logarithmic search

3 Modified orthogonal search algorithm

We present the modification in original OSA by utilizing a 3×3 grid search pattern at the center to exploit the center biased characteristics of motion vector distribution in real world video sequences. Fig. 2 shows the search pattern of MOSA. The search window size is $(2d + 1) \times (2d + 1)$ where d is the maximum displacement in horizontal and vertical direction. The step size is initialized at $st = \lceil d/2 \rceil$, where $\lceil\rceil$ is upper truncation function. We have assumed the search window size as 7×7 (i.e., maximum displacement of motion vector $d = \pm 7$).

The proposed MOSA is decomposed into following steps.

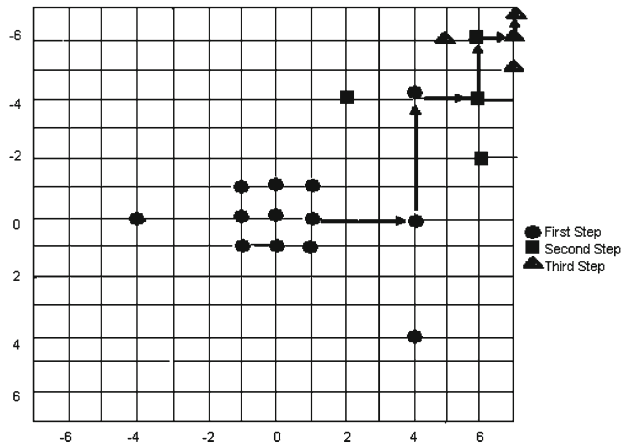

Fig. 2 Modified orthogonal logarithmic search

Step 1 In the initial stage of the algorithm, we propose to add eight extra search points around the center in addition to the original search points in horizontal direction used in OSA as shown in Fig. 2. The advantage of adding these extra search points is to capture the small motion at the center. If the minimum BDM at this stage occurs at the search window center i.e., at (0, 0), then assume the block is with zero motion and further search bring to an end (half way stop technique). Otherwise go to step 2.

Step 2 If the winning point of step 1 corresponds to one of the eight point of the center grid then step size is initialized to one and two positions horizontally on either side and two vertical positions above and below of the winning point are searched. The search point with minimum BDM corresponds to motion vector. The algorithm is halted at this stage or else go to step 3.

Step 3 Shift the center at the winning point of step 1 (having minimum BDM) for the vertical search and two positions are searched in the vertical direction (up and down) retaining the same step size.

Step 4 The step size '*st*' is reduced by half and the center is moved to the minimum BDM point of step 3. Again two positions in the horizontal direction are checked with step size *st* apart from the center.

Step 5 Repeat the step 3 and 4, till the step size *st* becomes one.

The proposed MOSA retained the simplicity and regularity of OSA. Although MOSA take eight additional search points at the initial stage as compared to OSA, the early termination of the algorithm at step 1 helped to reduce the computations significantly.

3.1 Analysis of the algorithm

For block matching algorithms, the number of search points required for each motion vector estimation measures the computational complexity. In this section, we estimate the number of search points needed to find motion vector for MOSA as compared with the OSA and 3SS algorithm.

The number of search points required for 3SS algorithm given by [6] is,

$$[1 + 8\log_2 d] \tag{1}$$

Hence 3SS requires total 25 search points with 7×7 search window.

The computational complexity of OSA algorithm measured by [1] is,

$$[1 + 4\lceil\log_2(d + 1)\rceil] \tag{2}$$

For 7×7 search window size , the OSA algorithm requires 13 search points.

A search example using the center biased MOSA for motion vector estimated at $(-7, 7)$ is shown in Fig. 2. The 21 search points are required to obtain the motion vector for this particular example. From Fig. 2, we can easily deduce that the total number of search points can vary from 11 (best case scenario) to 21 (worst case scenario). But experimental result indicates that average search points for 100 frames are less than 11 since for the boundary blocks the search window extends outside the frame boundary, as a result some search points gets omitted from the search pattern. Thus with this discussion, undoubtedly we claim that, with MOSA, motion vectors are found with fewer search points as compared to 3SS and OSA algorithm.

4 Experimental results

The algorithm is implemented on a Pentium IV, 2.4 GHz personal computer. In our experiment BDM is defined to be the mean square error (MSE) [19].

$$MSE(i, j) = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}(f(m, n) - f(m + i, n + j))^2$$
$$-d \le i, j \le d \tag{3}$$

where $f(m, n)$ represents the current sub-block of N^2 pixels at coordinates (m, n) and $f(m + i, n + j)$ represents the corresponding block in the previous frame at new coordinates $(m + i, n + j)$ Choosing large block size is computationally efficient but it results in the poor quality of the reconstructed frame so we choose block size as 8×8 as a tradeoff between computational complexity and the quality of the image [20]. The maximum displacement of block in horizontal and vertical direction is assumed to be

Table 1 Average MSE for different algorithms

Sequences	FSA	3SS	OSA	MOSA
Foreman	26.36	34.13	40.2	31.23
Claire	3.60	3.67	3.96	3.61
Carphone	20.6	24.52	26.6	22.61
Silent	12.54	14.87	16.42	17.90
News	17.10	18.86	19.98	19.98
Grandma	3.70	3.78	3.92	3.72
Suzie	15.09	18.32	22.33	20.42
Miss America	5.02	5.54	5.79	5.10
Salesman	6.51	6.97	7.43	6.82
Trevor	30.08	35.47	38.27	38.33

Table 3 Average search points for different algorithms

Sequences	FSA	3SS	OSA	MOSA
Foreman	205.18	25	12.40	11.14
Claire	205.18	25	12.39	10.66
Carphone	205.18	25	12.40	11.43
Silent	205.18	25	12.39	10.62
News	205.18	25	12.39	10.41
Grandma	205.18	25	12.40	10.63
Suzie	205.18	25	12.40	11.26
Miss America	205.18	25	12.40	11.06
Salesman	205.18	25	12.39	10.37
Trevor	205.18	25	12.39	10.76

Table 2 Average SNR for different algorithms

Sequences	FSA	3SS	OSA	MOSA
Foreman	34.2	33.2	32.6	33.59
Claire	43.25	43.20	42.99	43.24
Carphone	35.2	34.6	34.32	34.95
Silent	37.86	37.18	36.81	36.76
News	38.14	37.95	37.75	37.88
Grandma	43.65	43.59	43.53	43.62
Suzie	36.71	36.01	35.33	35.88
Miss America	41.54	41.18	41.02	41.48
Salesman	40.86	40.64	40.45	40.76
Trevor	35.55	35.21	34.87	35.33

$d = \pm 7$. For the performance evaluation of standard algorithms FSA, OSA, 3SS and MOSA, 100 frames of ten QCIF video sequences "Foreman", "Claire" ,"Carphone", "Silent", "News", "Grandma", "Suzie" "Salesman", "Trevor" and Miss America" are selected. These video sequences are categorized into three classes based on fast, moderate and slow motion. The video sequences of Silent, Claire and Grandma are all of slow object translation with low motion activities. News, Suzie and Miss America are with moderate motion while Foreman, Carphone, Salesman and Trevor sequences are of fast object translation with high motion activity. The motion compensated values of MSE and SNR are being used as the metrics to evaluate the performance of standard algorithms and proposed MOSA, with results presented in Tables 1 and 2. Computational complexity in terms of average number of search points per block is summarized in Table 3.

It is seen from Table 3 that proposed MOSA algorithm requires less number of search points as a result of half way stop technique and it also improves the MSE and SNR values as compared to OSA which is shown in Tables 1 and 2.

For "Foreman" and "Carphone" which contains large motion, the MOSA algorithm has obtained speed improvement of nearly 50% over 3SS. For "Claire" and "Grandma" sequences with motion vectors limited within a small region, the proposed MOSA algorithms achieves close to 55% speed improvement over 3SS as seen from Table 3. Tables 1 and 2 indicate that MOSA has given performance close to FSA for "Claire" and "Grandma" sequences with small motion content. And it performs better for "Foreman" "Salesman" and "Carphone" sequences of large motion over 3SS and OSA. 3SS and OSA utilize uniformly allocated search points in their first stage, which is not efficient to catch small motions appearing at search window center. By utilizing 3×3 pattern of search points at the center of search window we able to catch the small motion with MOSA. The advantage of the novel MOSA in terms of the speed of operation (computational complexity) is much more dramatic. Experimental result have shown that the MOSA performs better than OSA in terms of the motion compensation error and it is superior to 3SS in terms of computational complexity. For more clarification, the performance measured in terms of MSE and SNR for three test sequences each one representing different class of motion like "Foreman", "Claire" and "Miss America" are represented in Figs. 3, 4 and 5 respectively.

5 Conclusion

The proposed MOSA in block motion estimation results in the significant speed up gain over 3SS and FSA. Search for the center biased motion vectors have been facilitated by adding 8 extra search points, which are the neighbors of the window center. The introduction of the half way stop technique reduced the computational complexity and the time required for the computation. The strength of MOSA algorithm lies in its speed of operation which helps to reduce

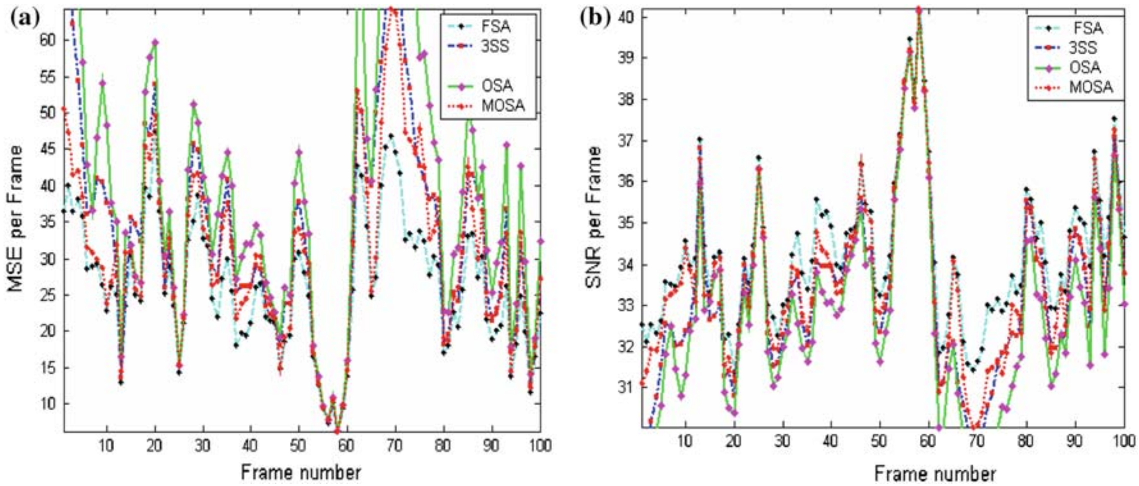

Fig. 3 Comparative MSE & SNR performance of "Foreman" using FSA, 3SS, OSA and MOSA

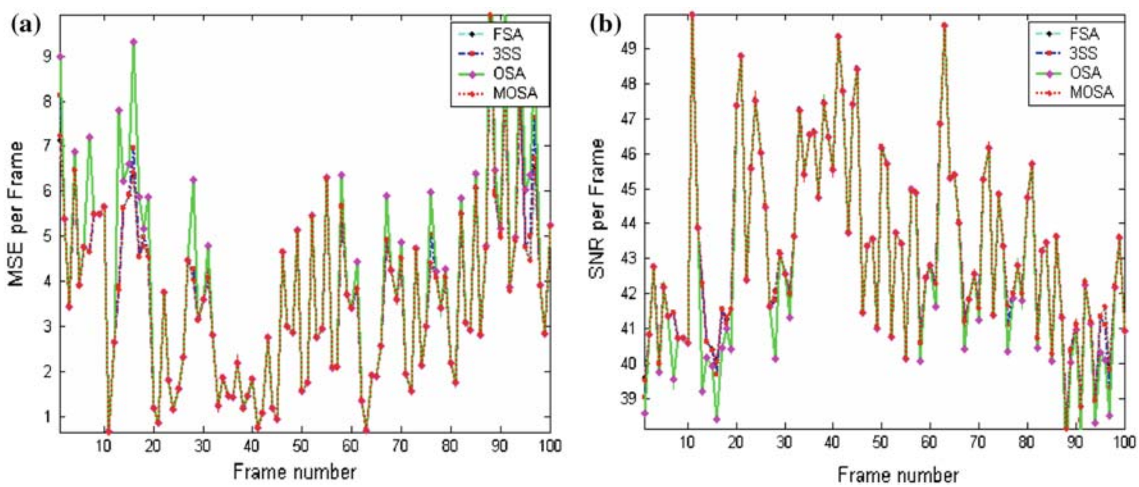

Fig. 4 Comparative MSE & SNR performance of "Claire" using FSA, 3SS, OSA and MOSA

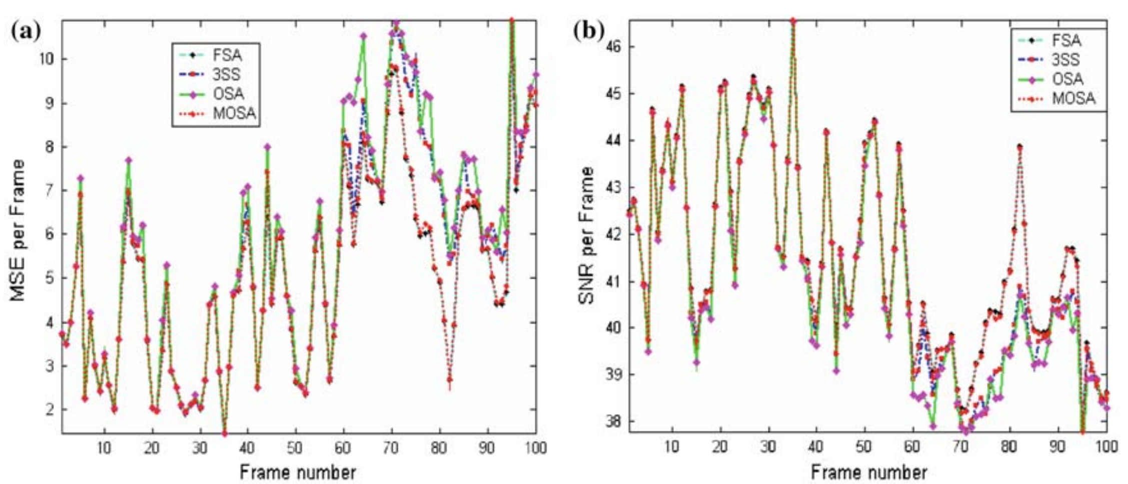

Fig. 5 Comparative MSE & SNR performance of "Miss America" using FSA, 3SS, OSA and MOSA

computational complexity. The speed of operation is almost 80 and 50% faster than FSA and 3SS respectively. The SNR performance ensures the good quality of reconstructed frame comparable to those of FSA, 3SS and OSA. Summing up; we conclude that MOSA is a good candidate to replace FSA and 3SS methods. Owing to its ability to reduce complexity, it can be recommended for efficient hardware implementation.

References

1. Soongsathitanon, S., Woo, W.L., Dlay, S.S.: Fast search algorithms for video coding using orthogonal logarithmic search algorithm. IEEE Trans. Consum. Electron. **51**(2), 552–559 (2005)
2. ISO/IEC 11 172-2, Information technology-coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-part 2: Video [MPEGl Video] (1993)
3. Koga, T., Iinuma, K., Hirano, A., Iijima, Y., Ishiguro, T.: Motion compensated interframe coding for video conferencing. In: Proc. Nat. Telecommun. Conf., New Orleans, LA, pp. G5.3.1–5.3.5, Nov 29–Dec 3 (1981)
4. Srinivasan, R., Rao, K.R.: Predictive coding based on efficient motion estimation. IEEE Trans. Commun. **COM-33**(8), 888–896 (1985)
5. Jain, J., Jain, A.: Displacement measurement and its application in interframe image coding. IEEE Trans. Commun. **COM-29**(12), 1799–1808 (1981)
6. Yeong-Kang, L.: A memory efficient motion estimator for three step search block-matching algorithm. IEEE Trans. Consum. Electron. USA **47**(3), 644–651 (2001)
7. Lai-Man, P., Wing-Chung, M.: A novel four-step search algorithm for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **6**, 313–317 (1996)
8. Reoxiang, L., Bing, Z., Liou, M.L.: A new three-step search algorithm for block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **4**, 438–442 (1994)
9. Nisar, H., Choi, T.S.: An advanced center biased three step search algorithm for motion estimation. In: IEEE International Conference on Multimedia and Expo. ICME2000, Part vol. 1, pp. 95–98, Piscataway, NJ (2000)
10. Liu, L.K., Feig, E.: A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Trans. Circuits Syst. Video Technol. **6**(4), 419–422 (1996)
11. Tham, J.Y., Ranganath, S., Ranganath, M., Kassim, A.A.: A novel unrestricted center-biased diamond search algorithm for block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **8**(4), 369–377 (1998)
12. Cheung, C.H., Po, L.M.: A novel cross-diamond search algorithm for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **12**(12), 1168–1177 (2002)
13. Zhu, C., Lin, X., Chau, L.P.: Hexagon-based search pattern for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **12**, 349–355 (2002)
14. Lai-Man Po, Chi Wang Ting, Ka-Ho-Ng: New enhanced hexagon based search using point—oriented inner search for fast block motion estimation. Int. J. Signal Process. **3**(3), 1304–4478 (2006)
15. Hong, W.G., Oh, T.M.: Sorting-based partial distortion search algorithm for motion estimation. Electron. Lett. **40**, 113–115 (2004)
16. Chau, L.-P., Zhu, C.: A fast octagon—based search algorithm for motion estimation. Signal Process. **83**, 671–675 (2003)
17. 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)
18. Gharavi, H., Mills, M.: Block matching motion estimation algorithms—new results. IEEE Trans. Circuits Syst. **37**, 649–651 (1990)
19. Jar-Ferr, Y., Shu-Sheng, H., Wei-Yuan, L.: Simplified block matching criteria for motion estimation. IEICE Trans. Inf. Syst. **E83-D**, 922–930 (2000)
20. Ahmad, A., Khan, N., Masud, S., Maud, M.A.: Efficient block size selection in H.264 video coding standard. Electron. Lett. **40**, 19–21 (2004)