Block Based Motion Estimation Algorithms: Analysis
KIRAN KUMAR VEMULA* and S.NEERAJA. **

* Department of Electronics and Communication Engineering, Malla Reddy College of Engineering and Technology, Hyderabad, Telangana, India
** Department of Electrical, Electronics and Communications Engineering, GITAM (Deemed to be University) Visakhapatnam, Andhra Pradesh, India.

Abstract—several video coding standards and techniques have been introduced for multimedia applications, particularly h.26x series for video processing. These standards employ motion estimation process in order to reduce the amount of data that is required to store or transmit the video. Motion estimation process is an inextricable part of the video coding as it removes the temporal redundancy between successive frames of video sequences. This paper is about these motion estimation algorithms, their search procedures, complexity, advantages, and limitations. A survey of motion estimation algorithms including full search algorithm, many fast search and fast full search block based algorithms has been presented. An evaluation of up to date motion estimation algorithms, based on a number of empirical results on several test video sequences, is presented as well.

Index Terms— Motion estimation, video coding, minimum block distortion measure, temporal / spatial redundancy.

1. INTRODUCTION

At present, online videos play a significant role in everyday life and the video technology has become the future of content marketing. The basic task of video coding is to reduce the huge amount of raw data in video sequence by removing spatial and temporal redundancies in video data. Motion estimation technique plays an important role in video coding process by removing temporal redundancy of video signal. The simple and efficient motion estimation technique is block based motion estimation (BBME) technique, which has been adopted in many video coding standards such as h.26x series and MPEGx series [1] – [6].

In real time video processing, the Full-search (FS) algorithm demands enormous computations. The huge computational cost of FS algorithm has laid the foundations for broad and deep research in motion estimation. The research has given many fast block matching algorithms. These algorithms can roughly be categorized as fast search [7]–[50] and fast full-search [51]–[59] block matching algorithms.

In this paper, an overview of selected algorithms in the last forty years and a comprehensive comparison of some well known algorithms in terms of computational complexity and error distortion are presented. The rest of the paper is organized as follows. In section 2, the brief analysis of fast search and fast full-search block based motion estimation algorithms are presented. Section 3 gives the comparison of some well known algorithms. Finally, the conclusions are presented in section 4.

2. BLOCK BASED MOTION ESTIMATION ALGORITHMS

The key goal of block based motion estimation algorithms is to find out the magnitude and direction of motion (motion vector) between a macroblock of current frame and best matched candidate block of the reference frame. The most commonly used matching criterion which measures the error distortion between the macroblock of current frame and candidate blocks in reference frame is sum of absolute difference (SAD). The SAD between an $M \times N$ size macroblock with top-left corner at $(p, q)$ and an $M \times N$ size candidate block with top-left corner at $(p + x, q + y)$ is defined in the eq (1).

$$SAD(x,y)=\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |(p+i, q+j) - R (p+x+i, q+y+j)|$$

where $I(.,.)$ and $R(.,.)$ denote current frame and reference frame pixel values. The co-ordinates of motion vector $x$ and $y$ are defined in the eq (2).

$$(x,y)=\arg\min_{(\hat{x}, \hat{y}) \in R} SAD(\hat{x}, \hat{y})$$

where $R=\{(\hat{x}, \hat{y}) \mid -s \leq \hat{x}, \hat{y} \leq d\}$ and $d$ represents the search range. It is obvious from eq (2) that the SAD criterion involves $(M \times N) - 1$ addition operations, $M \times N$ absolute operations and $M \times N$ subtraction operations i.e. one SAD computation requires $3 \times M \times N$ operations approximately.

2.1 Fast Search Block Based Motion Estimation Algorithms

In order to reduce the huge computational cost of FS algorithm, many fast search block based motion estimation algorithms [7]–[50] have been presented at the cost of slight reduction in error distortion given by peak signal-to-noise.
These algorithms may be classified into the following categories: reduction in number of search points [7]–[27], predictive motion estimation [28]–[34], adaptive search pattern switching strategy [35]–[38], multi-resolution motion estimation [39]–[45] and fractional-pixel interpolation [46]–[50]. Present fast search block based motion estimation algorithms belong to any one of them or utilize a combination of the above categories.

In general, the fast search block matching algorithms which belong to reduction in number of search points category [7]–[27] are mainly developed with an assumption that the error between a macroblock and a candidate block increases monotonically as the search point moves away from the optimal search point. In the early 1980s, some fast search block based motion estimation algorithms such as the three-step search (TSS) [7], two-dimensional logarithmic search (TDL) [8], the conjugate directional search (CDS) and its simplified version one-at-a-time search (OTS) [9] etc were proposed.

In TSS algorithm, the search procedure employs rectangular shaped search pattern which consists of nine search points including the centre at each step. Initially, the step size is taken as ceil(s/2) and is reduced by a factor two in the subsequent steps, where s is search range. The search stops when step size is reduced to 1. Fig. 1 shows an example of TSS search procedure to find a motion vector at (3, -2). The total number of steps and the total number of checking points are given by log2(s + 1) and 1 + 8[log2(s + 1)] respectively.

Fig.1. An example of a search procedure of TSS algorithm for finding motion vector (3, -2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

NTSS algorithm, proposed by Renxiang Li et al, performs better than TSS in terms of motion prediction quality and computational complexity while retaining the regularity and simplicity of the TSS algorithm. NTSS algorithm is developed mainly with an assumption that the motion vector distribution of most real world video sequences is centre biased. Therefore, besides the original search points of TSS, NTSS checks eight additional search points around the search centre at the first step (total 17) as shown fig. 2. Furthermore, the NTSS quickly identifies stationary and quasi-stationary blocks by applying a half way stop technique. In the first step, the minimum BDM point may occurs at the search window centre, at any one of the eight search points around the search centre or at any one of the remaining eight search points. In the first case, the block is considered as stationary block and the search stops. In the second case, the block is considered as quasi stationary block and the search stops after checking eight search points around the minimum BDM. In the final case (if the block is neither stationary nor quasi stationary), the search follows complete TSS procedure.

In [13], a four step search (4SS) algorithm has been proposed for motion estimation [13]. This algorithm includes a half-way stop technique and center-biased motion vector distribution characteristic similar to NTSS. However, the number of block matches of 4SS in the worst case is 27 when the maximum search range is ±7. With the maximum search range of ±7, the 4SS employs two different search patterns with 5×5 and 3×3 square window sizes. For first three search steps, if the minimum BDM search point is positioned at centre the search goes directly to fourth search step. An example of search procedure to find a motion vector at (6, 4) is shown in fig. 3.

Fig.2. An example of a search procedure of NTSS for finding motion vector (2, -2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

Fig.3. An example of a search procedure of 4SS for finding motion vector (6, 4). Each search point is indicated by its search step number and red colored point is the minimum BDM point.
One-at-a-time search (OTS) [9] is a 1-D gradient descent search algorithm. At first, OTS searches along the horizontal search direction until the minimum BDM value lies between two higher BDM values. Then, the search direction changes to vertical direction until the minimum BDM value is found in vertical direction. The OTS search path to locate motion vector (3, 3) is shown in fig. 4.

Several OTS based motion estimation algorithms such as block-based gradient descent search (BBGDS) [14] and directional gradient descent search (DGDS) [15] algorithms have been developed. The BBGDS is a 2-D gradient descent search motion estimation algorithm which searches for the minimum BDM block along the block-based gradient descent direction. At each search step, it applies a square search pattern which consists of nine search points. The eight search points surround the search center independently performs motion estimation in all the possible eight directions from the search center. The search continues until the minimum BDM search point is positioned at the search centre. An example of BBGDS search path to locate a motion vector at (2, -2) is shown in fig. 5.

The DGDS independently applies OTS strategy in eight directions of the search center to find eight directional minimum search points. Among these eight directional minimum search points, the minimum one becomes the search centre for the next search step. At any search step if the least among eight directional minimum search points is search centre, search stops with search centre as the motion vector. The DGDS search path to locate motion vector (5, 2) is shown in fig. 6.

The diamond search (DS) algorithm [16]-[17] locates a small area of global minimum by applying large diamond search pattern (LDSP) and then traces the global minimum in the located small area by applying a compact small diamond search pattern (SDSP). An example of search procedure to find a motion vector at (3, -2) is shown in Fig. 7. DS starts search by checking 9 search points of LDSP positioned at search window center. A new SDSP or LDSP is centered at minimum BDM point depending on whether the minimum BDM point is search center or not. The search continues until the new SDSP is centered and the minimum BDM point of SDSP will be the final motion vector. The hexagonal search (HS) algorithm with circle approximated search pattern is proposed in [18]. The search procedure of HS is same as that of DS except that the HS performs a coarse search by using a large hexagon search pattern which is close enough to a circle. An example of HS search path to locate a motion vector at (3, -2) is shown in fig. 8.

The modifications of HS [19] - [21] are developed for reducing computational cost against HS algorithm. These
algorithms essentially focus on methods to improve the inner search procedure of HS. An enhanced hexagonal search (EHS) algorithm [19] reduces the search points by employing the six-side-based fast inner search method. EHS algorithm calculates the group-sum distortion to predict a part of inner search that has to be examined. In [20], an enhanced hexagonal search using point-oriented inner search (EHS-POIS) [20] apply mean internal distance to calculate the normalized group distortions of the large hexagon. Then it checks only two inner search points which are associated to minimum normalized group distortions. An enhanced hexagonal search using direction-oriented inner search (EHS-DOIS) [21] forms pseudo-points prediction pattern from the large hexagon. EHS-DOIS calculates the group distortions of these pseudo-points to select one inner search point.

Fig.7. An example of a search procedure of DS for finding motion vector (3, -2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

Fig.8. An example of a search procedure of HS for finding motion vector (3, -2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

The algorithms belong to predictive motion estimation category [28]–[34] reduce the computational cost considerably by using the temporal and/or spatial correlation among motion vectors. In [31], motion vector field adaptive search technique (MVFAST) efficiently uses adjacent blocks motion information for performing motion estimation effectively. Before starting search at each macroblock, MVFAST calculates the city block lengths of the adjacent motion vectors. This city block length classifies the motion content of current macroblock as high, medium or slow motion. According to motion activity, the search strategy and search centre of current macroblock are determined. Furthermore, a halfway-stop technique is included in MVFAST such that it terminates the search early by checking (0, 0) predictor.

The search performance of MVFAST is further improved in predictive motion vector field adaptive search technique (PMVFAST) [32] with median predictor and collocated block’s motion vector. PMVFAST employs adaptively early search termination technique, unlike MVFAST where a fixed early search termination technique is used. Enhanced predictive zonal search (EPZS) [34] improves the search performance of PMVFAST by using additional higher probable predictors, and with improved threshold calculations. The algorithms belong to search patterns switching category [35]–[38] employ an adaptive switching strategy i.e. the algorithms dynamically applies various search patterns according to the motion activity. Consequently, the number of search locations is reduced drastically.

An adaptive search patterns switching algorithm was proposed in [38]. This algorithm predicts the motion activity of a block and then uses an appropriate search pattern for performing motion estimation. For small motions, center-biased search patterns such as NTSS, DS and BBGDS are used. The non center-biased search patterns such as TSS and 4SS are used for large motions. The motion content of a block is determined by an error descent rate (EDR). This EDR is calculated from block distortions of search window center and its four neighboring search points. This EDR is defined as $EDR = D_B/D_A$ where $D_A$ represents distortion of the block at center of the search window and $D_B$ represents minimum distortion of the four neighboring blocks of the search window center.

The algorithms belong to multiresolution techniques [39]–[45] represent the reference and current frames by pyramidal structure with various levels. Each level of this representation is a reduced resolution representation of the lower level and is obtained by sub sampling and spatial low-pass filtering the lower level. The motion field estimated at the present coarser resolution level is interpolated to form the initial solution for the motion field at the next finer resolution level as this initial solution is more likely to be near to the global minimum point. Therefore, the search at each resolution level is restricted to a smaller search range than the actual search range at the finest resolution level. Consequently, total computational cost is less than the computational cost demanded in the finest resolution directly. The algorithms belong to fractional-pixel motion estimation (FPME) techniques [46]–[50] achieve further reduction in bit rate i.e., improvement in video quality by applying fractional-pixel interpolation (FPI) algorithms.

2.2 Fast Full-Search Block Based Motion Estimation Algorithms

The fast full-search algorithms minimize the computational
complexity of the motion estimation process while preserving the same PSNR performance of full-search algorithm. Many fast full-search algorithms have been proposed in last four decades. Some eminent algorithms are: successive elimination technique based algorithms [51]–[58], partial distortion elimination based algorithms [75]–[82], winner-update approach based algorithms [69]–[70], and projection techniques based algorithms [71]–[74]. The most popular of these algorithms is the successive elimination algorithm (SEA) [51].

SEA finds the optimal motion vectors like full-search algorithm, but with less computational cost. The SEA rejects the search points which may not be best possible search points before computing full distortion measure for those search points. SEA skips these impossible search points by examining if the current minimum SAD (SADmin) is less than partial distortion measure.

In [52], block sum pyramid algorithm (BSPA) skips the non best candidate blocks by calculating partial errors hierarchically at every candidate block before computing the rigorous full distortion. In [53], multilevel successive elimination algorithm (MSEA) rejects more number of candidate blocks than those of SEA by using additional boundary levels. MSEA obtains these boundary levels by partitioning blocks into four equal sized sub blocks continually until a 2×2 sub block is arrived at. MSEA has shown search speed improvement against SEA by applying these boundary levels sequentially to skip some highly impossible search points which could not be rejected by the SEA boundary.

In MSEA, very large gaps exist between two contiguous boundary levels. Because of such large gaps, the effectiveness of MSEA is undermined. In [54], a fine granularity successive elimination (FGSE) is proposed to make up for this inefficiency of MSEA. FGSE algorithm reduces the gaps between two contiguous boundary levels by increasing the number of boundary levels. So, highly impossible search points are filtered out earlier in FGSE algorithm than in MSEA.

In [55], an adaptive MSEA (AdaMSEA) divides the search area based on homogeneity of the macroblock. In order to increase the possibility of skipping impossible search points in the early stage, the blocks with large variances are partitioned into sub blocks first. Winner update algorithm with integral image (WUI) is proposed in [59]. This algorithm replaces the hierarchical pyramid structure of the matching block by an integral image. This integral image facilitates the evaluation of partial block sum norms dynamically and therefore, WUI reduces the computational complexity of motion estimation.

3. RESULTS

This section presents the simulation results pertaining to the motion prediction quality and computational complexity of various up to date and famous motion estimation algorithms such as DS, CDS, DGDS, EHS-DOIS, ARPS-2, DASp, SEA, MSEA, AdaMSEA and WUI. Ten test video sequences with different motion contents and different video formats (HD, CIF, and QCIF) have been used to analyze the performance of these algorithms. Ten test videos contain various motion contents and have different resolutions. Kirsten-Sara and Akiyo test videos contain low-motion content i.e., maximum blocks are stationary blocks. Suzie, Mobile, and Flower are the test videos which consist of medium motions with stationary and quasi-stationary blocks. Mobile is a typical test video in which the local and global motions are complex. Rocket launch, Cricket and Foreman test videos have large motions. Rhinos and Robot boat test videos consist of complex motions with fast camera zooming and panning.

The search ranges ±63 and ±15 are used for HD test video sequences (Rocket launch and Kirsten-Sara) and the remaining (QCIF and CIF) video sequences respectively. Block size set to 16 x 16. In the comparison of various algorithms, PSNR is used as a measure for motion prediction quality and average number of operations per block measures the computational complexity. The average numbers of operations per block in each algorithm are summarized in table 1. The degree of motion prediction quality of every algorithm with respect to full search algorithm is shown in table 2. It is very clear from these tables that the fast search algorithms (DS, CDS, DGDS, C, ARPS-2, and DASp) reduce the computational complexity significantly but degrade the PSNR performance when compared to full search algorithm. Whereas, the fast full search algorithms (SEA, MSEA, AdaMSEA and WUI) obtain same PSNR of full search but with high computational complexity.

From table 1, it is obvious that DASp demands less number of operations when compared to other algorithms. ARPS-2 is better than DS, CDS, EHS-DOIS and DGDS in terms of number of operations. With respect to video sequences (Akiyo and Kirsten-Sara) that have small motion content, all the algorithms including DASp and ARPS-2 have approximately same number of operations. However, DASp and ARPS-2 require less number of operations irrespective of motion activity in video sequences.

It is clear from table 2 that the DGDS obtains better average PSNRs than those of DS, CDS, DASp, ARPS-2, and EHS-DOIS in all the video sequences. On an average, DGDS obtains 0.304dB better PSNR than that of CDS. However, CDS requires less number of operations when compared to that of DGDS. It is very clear from table 1 that EHS-DOIS finds motion vectors with less computational cost when compared to that of DGDS and CDS. However, EHS-DOIS gives least PSNR performance among all the algorithms (refer table 2). On the whole, in terms of average number of operations per block as the indicator for computational complexity, DASp is certainly the best ever. Simultaneously, with reference to PSNR as an indication for quality of video, the DASp is also apparently better than the DS, CDS, EHS-DOIS and ARPS-2 algorithms and comparable to the DGDS. Among fast full-search algorithms (SEA, MSEA, AdaMSEA and WUI), WUI has faster search performance.
4. CONCLUSION

In last four decades, multimedia research involves in development of efficient block matching algorithms to decrease the computational cost of motion estimation. These algorithms have been categorized into fast search and fast full search algorithms. This paper has presented basic search procedures of well known fast search and fast full search algorithms. A comprehensive analysis of famous and state-of-the-art algorithms in terms of their computational costs and motion prediction qualities is presented.

REFERENCES

1. G. Cote, B. Erol, M. Gallant, and F. Kosseintini, “H.263+: Video Coding at Low Bit Rates,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, pp. 849–66, November 1998.
2. ITU-T Rec. H.261, “Video Coding for Low Bit Rate Communication,” v1, Nov. 1995; v2, Jan. 1998; v3, Nov. 2000.
3. ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), “Advanced Video Coding for Generic Audiovisual Services,” v1, May, 2003; v2, Jan. 2004; v3 (with FRext), Sept. 2004; v4, July 2005.
4. J.-B. Lee and H. Kalva, The VC-1 and H.264 Video Compression Standards for Broadband Video Services, Springer Science+Business Media, New York, 2008.
5. H. G. Musmann, P. Pirsch, and H. J. Grallert, “Advances in picture coding,” Proc. IEEE, vol. 73, pp. 523–548, April 1985.
6. F. Dufaux and F. Moscheni, “Motion estimation techniques for digital TV: A review and a new contribution,” Proc. IEEE, vol. 83, pp. 858–876, June 1995.
7. T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, Motion compensated interframe coding for video conferencing,” in Proc. Nat. Telecommun. Conf., 1981, pp. C9.6.1–C9.6.5.
8. J.R. Jain and A.K. Jain, “Displacement measurement and its application in interframe image-coding” IEEE T. Commun. 29, 1799–1808 (1981).
9. R. Srinivasan and K. R. Rao, “Predictive coding based on efficient motion estimation,” IEEE T. Commun. 33, 888–896 (1985).
10. L.-W. Lee, J.-F. Wang, J.-Y. Lee, and J.-D. Shie, “Dynamic search window adjustment and interlaced search for block-matching algorithm,” IEEE Trans. Circuits and Systems for Video Tech., vol. 3, pp. 85–87, Feb. 1993.
11. S. C. Kwatra, C-M Lin and W. A. Whyte, “An adaptive algorithm formotion compensated color image coding,” IEEE Trans. Consum. Electron., vol. COM-35, pp. 747–754, July 1987.
12. R. Li, B. Zeng, and M.L. Liou, “A New Three-step Search Algorithm for Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 4, no. 4, pp. 438–442, Aug. 1994.
13. L.M. Po and W.C. Ma, “A Novel Four-step Search Algorithm for Fast Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 313–317, 1996.
14. L.K. Liu and E. Feig, “A Block-based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 4, pp. 419–422, 1996.
15. Lai-Man Po, Ka-Ho Ng, Kwok-Wai Cheung, Ka-Man Wong, Yusaf Md. Salah Uddin, and Chi-Wang Ting, “Novel Directional Gradient Descent Searches for Fast Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 8, pp. 1189–1195, Aug. 2009.
33. "Optimization Model Version 1.0," ISO/IEC JTC1/SC29/WG11 MPEG2000/N3224, Noordwijk, Netherlands, March 2000.

34. M. Tourapis, “Enhanced predictive zonal search for single and multiple frame motion estimation,” in Proc. SPIE Visual Commun. Image Process., San Jose, CA, pp. 1069–1079, 2002.

35. Shih-Yu Huang; Chuan-Yu Cho; Jia-Shung Wang, "Adaptive fast block-matching algorithm by switching search patterns for sequences with wide-range motion content," Circuits and Systems for Video Technology, IEEE Transactions on, vol.15, no.11, pp.1375,1384, Nov. 2005.

36. Ka-Ho Ng; Lai-Man Po; Ka-Man Wong, "Search Patterns Switching for Motion Estimation using Rate of Error Descent," Multimedia and Expo, 2007 IEEE International Conference on, vol., no., pp.1583,1586, 2-5 July 2007.

37. Jang-Jer Tsai; Hseuh-Ming Hang, "On Adaptive Pattern Selection for Block Motion Estimation Algorithms," Acoustics, Speech and Signal Processing, 2007. ICASSP 2007 IEEE International Conference on, vol.1, no., pp.I-1173-I-1176, 15-20 April 2007.

38. Ka-Ho Ng; Lai-Man Po; Ka-Man Wong; Chi-Wang Ting; Kwok-Wai Cheung, "A Search Patterns Switching Algorithm for Block Motion Estimation," Circuits and Systems for Video Technology, IEEE Transactions on, vol.19, no.5, pp.753,759, May 2009.

39. Zan, Jinwen; Ahmad, M.O.; Swamy, M.N.S., "A multiresolution motion estimation technique with indexing," Circuits and Systems for Video Technology, IEEE Transactions on, vol.16, no.2, pp.157,165, Feb. 2006.

40. Byung Cheol Song; Kang-Wook Chun, "Multi-resolution block matching algorithm and its VLSI architecture for fast motion estimation in an MPEG-2 video encoder," Circuits and Systems for Video Technology, IEEE Transactions on, vol.14, no.9, pp.1119,1137, Sept. 2004.

41. HailBing Yin; Jia, Huizhu; Honggang Qi; Ji, Xianghu; Xie, Xiaodong; Wen Gao, "A Hardware-Efficient Multi-Resolution Block Matching Algorithm and its VLSI Architecture for High Definition MPEG-Like Video Encoders," Circuits and Systems for Video Technology, IEEE Transactions on, vol.20, no.9, pp.1248,1254, Sept. 2010.

42. Varray, F.; Liebott, H., "Multi-resolution transverse oscillation in ultrasound imaging for motion estimation," Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on, vol.60, no.7, pp.1333,1342, July 2013.

43. Stuckler, J., Behnke, S.: Multi-resolution surfel maps for efficient dense 3D modeling and tracking. Journal of Visual Communication and Image Representation 25(1), 137-147 (2014).

44. Nieuwenhuisen, Matthias, and Sven Behnke. “Hierarchical planning with 3D local multi-resolution obstacle avoidance for micro aerial vehicles.” Proceedings of the Joint Int. Symposium on Robotics (ISR) and the German Conference on Robotics (ROBOTIK). 2014.

45. Drechsel, D., Stuckler, J., Behnke, S.: Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner. In: Robotics and Automation (ICRA), IEEE International Conference on (2014).

46. T. Wedi, “Adaptive interpolation filters and high-resolution displacements for video coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 4, pp. 484–491, Apr. 2006.

47. L. Shen, Z. Zhang, Z. Liu, and W. Zhang, “An adaptive and fast fractional pixel search algorithm in H.264,” Signal Process., vol. 87, no. 11, pp. 2629–2639, 2007.

48. Y. Vatis and J. Ostermann, “Adaptive interpolation filter for H.264/AVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 2, pp. 179–192, Feb. 2009.

49. Y. Lin and Y. C. Wang, “Improved parabolic prediction-based fractional search for H.264/AVC video coding,” Image Process. IET, vol. 3, no. 5, pp. 261–271, Oct. 2009.

50. S. Dikbas, T. Arici, and Y. Altunbasak, “Fast motion estimation with interpolation-free sub-sample accuracy,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 7, pp. 1047–1051, Jul. 2010.

51. W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” IEEE Trans. Image Processing, vol. 4, pp. 105–107, Jan. 1995.

52. C. Lee and L. Chen, “A fast motion estimation algorithm based on the block sum pyramid,” IEEE Trans. Image Process., vol. 6, no. 11, pp. 1587–1591, Nov. 1997.

53. X. Q. Gao, C. J. Duanmu, and C. R. Zou, “A Multilevel Successive Elimination Algorithm for block matching motion estimation,” IEEE Trans. Image Processing, vol. 9, pp. 501–504, Mar. 2000.

54. C. Zhu, W. S. Qi, and W. Ser, “Predictive Fine Granularity Successive Elimination for fast optimal block matching motion estimation,” IEEE Trans. Image Processing, vol. 14, no. 2 pp. 213-221, Feb.2005.

55. Shao-Wei Liu, Shou-Der Wei, and Shang-Hong Lai, “Fast Optimal Motion Estimation Based on Gradient-Based Adaptive Multilevel Successive Elimination,” IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 2, pp. 156–160, Feb. 2008.

56. C. Lee and L. Chen, “A fast motion estimation algorithm based on the block sum pyramid,” IEEE Trans. Image Process., vol. 6, no. 11, pp. 1587–1591, Nov. 1997.

57. Jong-Nam Kim; Dae-Kap Kang; Sung-Cheal Byun; II-Lo Lee; Byung-Ha Ahn, “A fast full-search motion estimation algorithm using sequential rejection of candidates from hierarchical decision structure,” Broadcasting, IEEE Transactions on, vol.48, no.1, pp.43,46, Mar 2002.

58. Yu-Wen Huang; Shao-Yi Chien; Bing-Yu Hsieh; Liang-Gee Chen, “Global elimination algorithm and architecture design for fast block matching motion estimation,” Circuits and Systems for Video Technology, IEEE Transactions on, vol.14, no.6, pp.898,907, June 2004.

59. Jik-Han Jung, Hwal-Suk Lee, Je Hee Lee, and Dong-Jo Park “A Novel Template Matching Scheme for Fast full-search Boosted by an Integral Image,” IEEE Signal Processing Letters, Vol. 17, No. 1, January 2010.
Table 1. The average numbers of operations per block in each algorithm.

| Video sequence | FS or Fast full-search algorithms | Fast search motion estimation algorithms | Fast full-search algorithms |
|----------------|-----------------------------------|------------------------------------------|----------------------------|
|                | DS      | CDS     | DGDS    | EHS-DOIS | ARPS-2 | DSAP | SEA | MSEA | Ada | WUI |
|Foreman         | 601520  | 13203   | 12398   | 14334    | 8178   | 7288 | 6532| 175080| 29301 | 25612 | 25301 |
|Mobile          | 668516  | 8253    | 7877    | 8919     | 5948   | 5181 | 4174| 270890| 30927 | 24153 | 24039 |
|Rhinos          | 668516  | 29327   | 23440   | 26515    | 12990  | 11662| 10378| 329652| 65321 | 50602 | 49305 |
|Robot boat      | 668516  | 26494   | 22016   | 25413    | 12447  | 11067| 11943| 363925| 72109 | 66944 | 66001 |
|Suzie           | 601520  | 9627    | 7993    | 8587     | 5768   | 4805 | 4496| 113280| 14317 | 12209 | 12152 |
|Akiyo           | 601520  | 6327    | 6314    | 6054     | 5155   | 4305 | 3551| 24608 | 6877  | 6205  | 6109 |
|Cricket         | 668516  | 13911   | 12734   | 12599    | 8344   | 7508 | 6379| 106799| 28641 | 22943 | 21513 |
|Flower          | 668516  | 10072   | 9323    | 9968     | 6729   | 6194 | 5322| 104960| 18303 | 16112 | 16001 |
|Kirsten-Sara    | 1081330 | 7981    | 7410    | 7537     | 5839   | 5201 | 4218| 216852| 63928 | 54883 | 53697 |
|Rocket launch   | 1081330 | 13364   | 12149   | 12964    | 8278   | 7403 | 6752| 284621| 82504 | 73864 | 73359 |

Table 2. The degree of motion prediction quality of every algorithm with respect to full search algorithm.

| Video sequence    | FS or Fast full-search algorithms | Fast search motion estimation algorithms | Fast full-search algorithms |
|-------------------|-----------------------------------|------------------------------------------|----------------------------|
|                   | DS      | CDS     | DGDS    | EHS-DOIS | ARPS-2 | DSAP | SEA | MSEA | Ada | WUI |
|Foreman            | 28.89   | 28.15   | 28.03   | 28.28    | 26.70  | 28.22| 28.05|
|Mobile             | 24.29   | 23.52   | 23.85   | 23.87    | 22.71  | 23.63| 23.83|
|Rhinos             | 30.23   | 27.62   | 27.81   | 28.40    | 27.66  | 27.34| 28.35|
|Robot boat         | 30.62   | 29.21   | 29.10   | 29.54    | 28.83  | 29.32| 29.05|
|Suzie              | 35.90   | 35.02   | 35.10   | 35.25    | 33.87  | 35.11| 35.13|
|Akiyo              | 44.16   | 44.16   | 44.16   | 44.16    | 43.25  | 44.16| 44.16|
|Cricket            | 35.95   | 33.66   | 33.95   | 34.99    | 33.19  | 33.94| 34.67|
|Flower             | 33.69   | 33.02   | 33.19   | 33.35    | 31.47  | 33.28| 33.17|
|Kirsten-Sara       | 44.74   | 44.18   | 44.21   | 44.39    | 42.45  | 44.10| 44.17|
|Rocket launch      | 38.95   | 37.53   | 37.69   | 37.90    | 36.20  | 37.66| 37.89|