Block Matching Algorithms for the Estimation of Motion in Image Sequences: Analysis

K. Srinivas Rao[a,*] **and A. V. Paramkusam**[b,**]

[a] *Department of CSE, MLR Institute of Technology, Hyderabad, 500043 India*
[b] *Department of ECE, Lendi Institute of Engineering and Technology, Vizianagaram, 535005 India*
** e-mail: ksreenu2k@gmail.com*
*** e-mail: adapa74@gmail.com*

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 for reducing 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.

Keywords: motion estimation, video coding, minimum block distortion measure, temporal/spatial redundancy

DOI: 10.1134/S1054661822010072

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 [4, 7, 10, 11, 23, 33]. 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 [1–3, 5, 6, 9, 12–22, 24, 27, 29, 30, 32, 34–52, 54–57, 59] and fast full-search [8, 25, 26, 28, 31, 53, 58] 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 fame 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)

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

where $I(.,.)$ and $R(.,.)$ denote current frame and reference frame pixel values. The coordinates of motion vector x and y are defined in Eq. (2)

$$(x, y) = \arg\min_{(\hat{x},\hat{y})\in R} SAD(\hat{x},\hat{y}), \quad (2)$$

where $R = \{(\hat{x},\ \hat{y})| - s \le \hat{x},\ \hat{y} \le d\}$ and d represents the search range. It is obvious from Eq. (2) that the *SAD*

Received November 24, 2020; revised April 29, 2021; accepted September 15, 2021

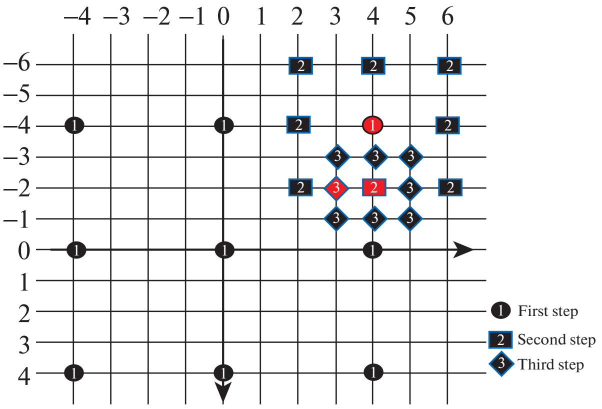

Fig. 1. An example of a search procedure of TSS algorithm for finding motion vector (3, –2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

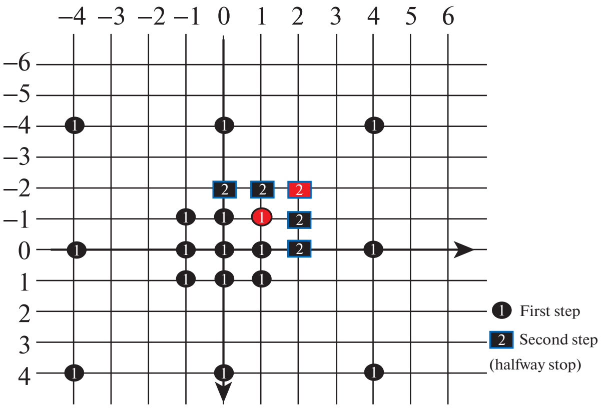

Fig. 2. An example of a search procedure of NTSS for finding motion vector (2, –2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

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 [1–3, 5, 6, 9, 12–14, 17–22, 24, 27, 29, 30, 32, 34–52, 54–57, 59] have been presented at the cost of slight reduction in error distortion given by peak signal-to-noise ratio (PSNR).These algorithms may be classified into the following categories: reduction in number of search points [1–3, 12, 17–22, 24, 27, 30, 36, 38, 40, 44, 55–57, 59], predictive motion estimation [14, 32, 39, 45–47, 52], adaptive search pattern switching strategy [9, 13, 34, 35], multiresolution motion estimation [6, 37, 42, 43, 48, 51, 54] and fractional-pixel interpolation [5, 29, 41, 49, 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 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) [17], two-dimensional logarithmic search (TDL) [12], the conjugate directional search (CDS) and its simplified version one-at-a-time search (OTS) [40], etc., were proposed. In TSS algorithm, the search procedure employs rectangular shaped search pattern which consists of nine search points including the center 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. Figure 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 $\log_2(s + 1)$ and $1 + 8[\log_2(s + 1)]$, respectively. NTSS algorithm [27], 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 center biased. Therefore, besides the original search points of TSS, NTSS checks eight additional search points around the search center 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 occur at the search window center, at any one of the eight search points around the search center 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 quasi-stationary 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 [20], a four-step search (4SS) algorithm has been proposed for motion estimation [20]. 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

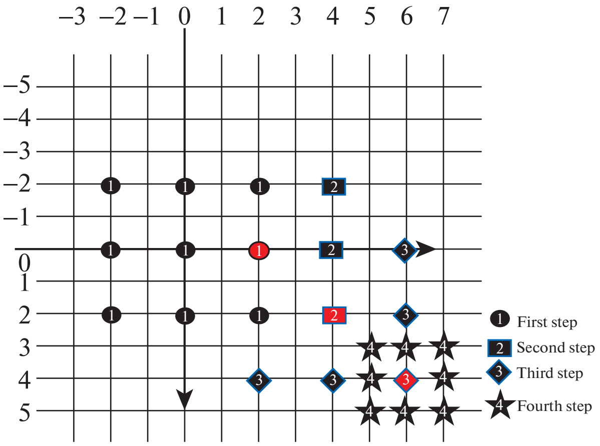

Fig. 3. An example of a search procedure of 4SS for finding motion vector (6, 4). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

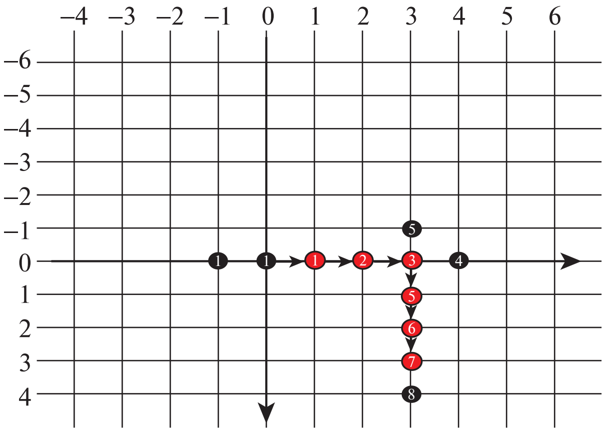

Fig. 4. An example of a search procedure of OTS for finding motion vector (3, 3). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

search steps, if the minimum BDM search point is positioned at center, 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.

One-at-a-time search (OTS) [40] is a 1D 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) [30] and directional gradient descent search (DGDS) [21] algorithms have been developed.

The BBGDS is a 2D 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 center. 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 center for the next search step. At any search step, if the least among eight directional minimum search points is search center, search stops with search center as the motion vector. The DGDS search path to locate motion vector (5, 2) is shown in Fig. 6.

The diamond search (DS) algorithm [44, 55] 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

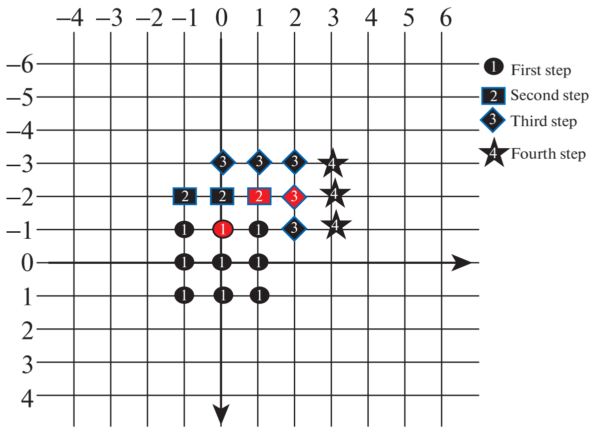

Fig. 5. An example of a search procedure of BBGDS for finding motion vector (2, –2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

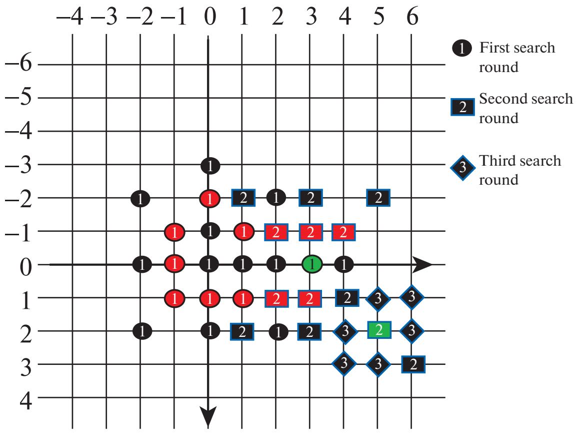

Fig. 6. An example of a search procedure of DGDS for finding motion vector (5, 2), each search point is indicated by its search round number, red colored points are the directional minimum search points and green colored point is the least of directional minimum search points.

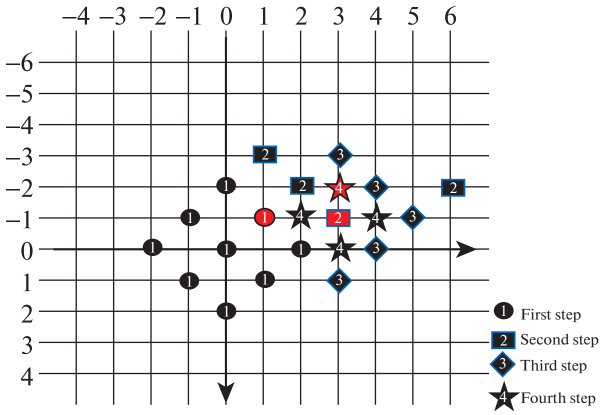

Fig. 7. An example of a search procedure of DS for finding motion vector (3, –2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

proposed in [56]. 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 [22, 57, 59] 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 [57] 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 [22], an enhanced hexagonal search using point-oriented inner search (EHS-POIS) [22] 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) [59] 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.

In the adaptive rood pattern search (ARPS) algorithm [36], it is modified algorithm adaptive rood pattern search (ARPS-2) [38] and the directional asymmetric search with prediction scheme (DASp) [18], a prediction scheme has been employed to better track large motions. These algorithms reduce the computational complexity of search process with prediction and best match prejudgment schemes. The ARPS predicts current block's motion vector with the motion vector of left adjacent block. This algorithm uses an adaptive rood pattern at initial search stage and then applies a unit-size rood pattern repeatedly to find the final motion vector. The ARPS has shown two to three times of search speed-up while maintaining fairly close PSNR when compared to DS. The ARPS-2 employs the median prediction to find the predicted motion vector, and, then, an adaptive rood pattern is positioned on this predicted motion vector. This results in a great reduction on computational cost of ARPS-2 over ARPS. The matching error information and the center-biased motion vector distribution characteristic have been effectively utilized in DASp algorithm for reducing the computational cost greatly. At first, DASp check eight adjacent search points of the search center in eight directions to estimate the most probable search direction in whose vicinity the optimal motion vector is present. Then, it uses one of the proposed directional search patterns to find the final motion vector.

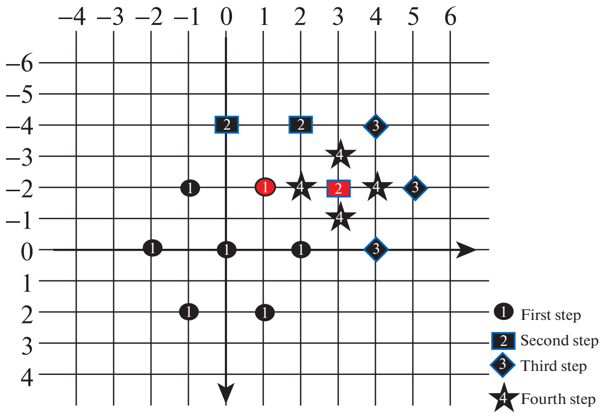

Fig. 8. An example of a search procedure of HS for finding motion vector (3, –2). Each search point is indicated by its search step number and red colored point is the minimum BDM point.

The algorithms belong to predictive motion estimation category [14, 32, 39, 45–47, 52] reduce the computational cost considerably by using the temporal and/or spatial correlation among motion vectors. In [39], 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 center 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) [45] 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) [47] improves the search performance of PMVFAST by using additional higher probable predictors, and with improved threshold calculations.

The algorithms belonging to search patterns switching category [9, 13, 34, 35] employ an adaptive switching strategy, i.e., the algorithms dynamically apply 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 [35]. 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 $\mathrm{EDR} = \mathrm{D_B}/\mathrm{D_A}$, where $\mathrm{D_A}$ represents distortion of the block at center of the search window and $\mathrm{D_B}$ represents minimum distortion of the four neighboring blocks of the search window center.

The algorithms belonging to multiresolution techniques [6, 37, 42, 43, 48, 51, 54] 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 subsampling and spatial low-pass filtering of 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 belonging to fractional-pixel motion estimation (FPME) techniques [5, 29, 41, 49, 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 [8, 15, 16, 25, 26, 28, 31, 53, 58]. The most popular of these algorithms is the successive elimination algorithm (SEA) [28]. 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 the 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 ($\mathrm{SAD_{min}}$) is less than partial distortion measure

In [25], 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 [8], multilevel successive elimination algorithm (MSEA) rejects a greater 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 subblocks continually until a 2 × 2 subblock 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 [58], 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 [31], 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 subblocks first. Winner-update algorithm

Table 1. The average numbers of operations per block in each algorithm

Video sequence	FS	Fast search motion estimation algorithms						Fast full-search algorithms			
		DS	CDS	DGDS	EHS-DOIS	ARPS-2	DASp	SEA	MSEA	Ada MSEA	WUI
Foreman	601 520	13 203	12 398	14 334	8178	7288	6532	175 080	29 301	25 612	25 301
Mobile	668 516	8253	7877	8919	5948	5181	4174	270 890	30 927	24 153	24 039
Rhinos	668 516	29 327	23 440	26 515	12 990	11 662	10 378	329 652	65 321	50 602	49 305
Robot boat	668 516	26 494	22 016	25 413	12 447	11 067	11 943	363 925	72 109	66 944	66 001
Suzie	601 520	9627	7993	8587	5768	4805	4496	113 280	14 317	12 209	12 152
Akiyo	601 520	6327	6314	6054	5155	4305	3551	24 608	6877	6205	6109
Cricket	668 516	13 911	12 734	12 599	8344	7508	6379	106 799	28 641	22 943	21 513
Flower	668 516	10 072	9323	9968	6729	6194	5322	104 960	18 303	16 112	16 001
Kirsten-Sara	10 813 330	7981	7410	7537	5839	5201	4218	216 852	63 928	54 883	53 697
Rocket launch	10 813 330	13 364	12 149	12 964	8278	7403	6752	284 621	82 504	73 864	73 359

with integral image (WUI) is proposed in [15]. 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 × 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 (ANOB) 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 degrades 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 a smaller 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 demand a smaller number of operations. However, DASp and ARPS-2 require a smaller 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 average, DGDS obtains 0.304 dB better PSNR than that of CDS. However, CDS requires a smaller 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 indi-

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					
		DS	CDS	DGDS	EHS-DOIS	ARPS-2	DASp
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

cator 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.

To comprehend the comparative studies shown in Tables 1 and 2 more vividly, ANOB and PSNR of all the algorithms are plotted in Figs. 9 and 10. Figures 9a–9j plot a frame by frame comparison of ANOB for all the algorithms applied to the ten test video sequences. Figures 10a–10j plot a frame by frame comparison of PSNR for all the algorithms applied to the ten test video sequences. In Figs. 9a–9j, the results of fast full-search algorithms have not been shown to avoid congestion between graphic lines of fast search algorithms. This is because the fast full-search algorithms require a huge computation when compared to fast search algorithms. Since the PSNR values of FS and the fast full-search algorithms are the same, the graph of FS in Figs. 10a–10j can be considered as graphs of the fast full-search algorithms. Figures 9a–9j clearly manifest that the DASp algorithm requires fewer operations compared to other algorithms in each frame. From these figures, it is also very clear that the ARPS-2 competes with DASp and performs better when compared to other algorithms in each frame.

It is clear from Figs. 10a–10j that all algorithms, except EHS-DOIS, can obtain a PSNR that is close to the PSNR that the FS algorithm can obtain in each frame. In most frames of all video sequences, DGDS shows better PSNR values when compared with other algorithms. In video sequences with small motion content such as Akiyo and Kirsten-Sara, all algorithms, except EHS-DOIS, show same performance as shown in Figs. 10f and 10j, respectively. So, we can observe that the graphs of all algorithms, except EHS-DOIS, are overlapping each other in these figures.

4. CONCLUSIONS

In last four decades, multimedia research involves in development of efficient block matching algorithms to decrease the computational cost of motion estimation. This paper has presented basic search procedures of well-known fast search and fast full-search algorithms. The integral image concept of the WUI algorithm makes WUI algorithm the fastest search algorithm of all fast full-search algorithms. On average, the WUI algorithm achieves a 96.51, 82.21, 15.97, and 1.74% speed-improvement rate over FS, SEA, MSEA, and AdaMSEA, respectively. On average, the DASp achieves a 99.17, 51.45, 46.36, 49.84, 21.79, and 11.36% speed-improvement rate over FS, DS, CDS, DGDS, EHS-DOIS, and ARPS-2, respectively. Computationally, EHS-DOIS has shown an excellence. DGDS has proven to be the best in terms of quality. However, the DASp has proven its efficiency in both computational cost and quality over other fast search algorithms. In summary, in terms of ANOB as the indicator for search speed, the fast search algorithms are certainly the best over fast full-search algorithms. Whereas, in terms of PSNR as the sign for quality, the fast full-search algorithms are clearly a bit better than the fast search algorithms. The EHS-DOIS and DASp have shown their computational efficiency by reducing as many search points as possible.

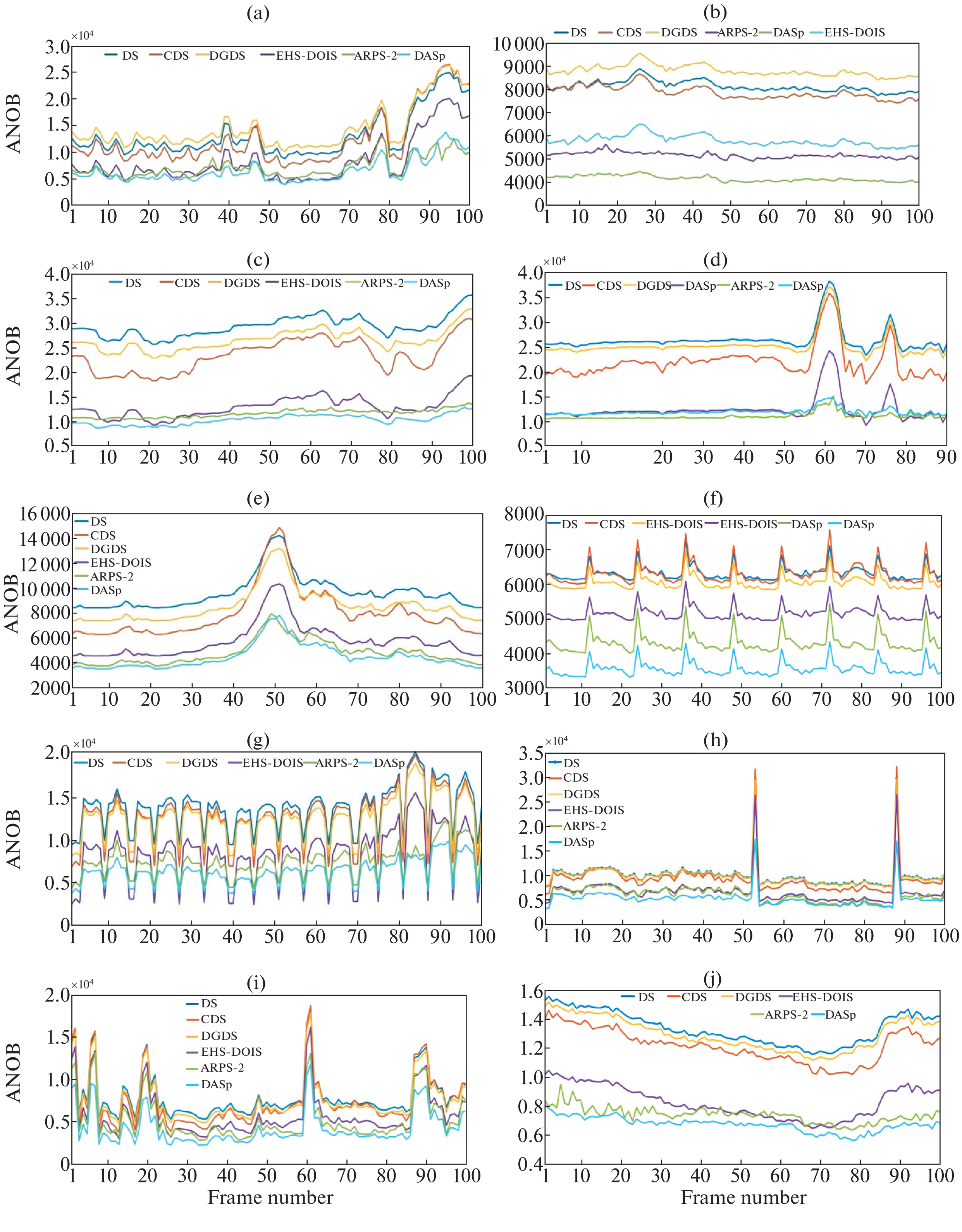

Fig. 9. The computational cost comparison of all the algorithms in terms of the average numbers of operations per block (ANOB) for various video sequences: (a) Foreman, (b) Mobile, (c) Rhinos, (d) Robot boat, (e) Suzie, (f) Akiyo, (g) Cricket, (h) Flower, (i) Kirsten-Sara, and (j) Rocket launch.

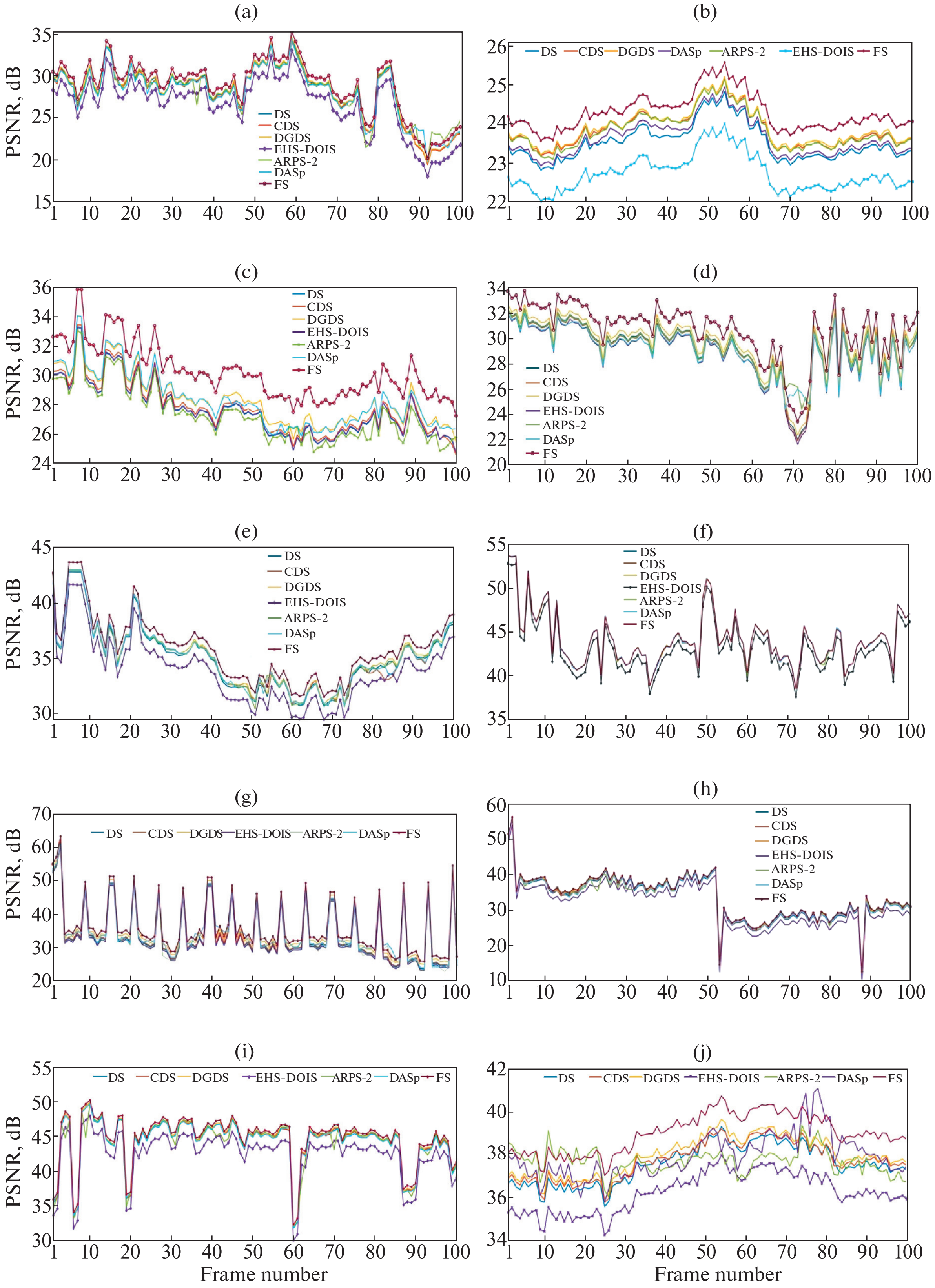

Fig. 10. The motion prediction quality comparison of all the algorithms in terms of the peak signal-to-noise ratio (PSNR) for various video sequences: (a) Foreman, (b) Mobile, (c) Rhinos, (d) Robot boat, (e) Suzie, (f) Akiyo, (g) Cricket, (h) Flower, (i) Kirsten-Sara, and (j) Rocket launch.

COMPLIANCE WITH ETHICAL STANDARDS

This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the *PRIA* Editorial Board decides not to accept it for publication.

Conflict of Interest

The authors declare that they have no conflicts of interest.

REFERENCES

1. Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," J. Visual Commun. Image Representation **17**, 264–290 (2006). https://doi.org/10.1016/j.jvcir.2004.12.002
2. C.-H. Cheung and L.-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol. **12**, 1168–1177 (2002). https://doi.org/10.1109/TCSVT.2002.806815
3. C.-H. Cheung and L.-M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," IEEE Trans. Multimedia **7**, 16–22 (2005). https://doi.org/10.1109/TMM.2004.840609
4. G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," IEEE Trans. Circuits Syst. Video Technol. **8**, 849–866 (1998). https://doi.org/10.1109/76.735381
5. S. Dikbas, T. Arici, and Y. Altunbasak, "Fast motion estimation with interpolation-free sub-sample accuracy," IEEE Trans. Circuits Syst. Video Technol. **20**, 1047–1051 (2010). https://doi.org/10.1109/TCSVT.2010.2051283
6. D. Droeschel, J. Stückler, S. Behnke, "Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner," in *IEEE Int. Conf. on Robotics and Automation (ICRA), Hong Kong, 2014* (IEEE, 2014), pp. 5221–5226. https://doi.org/10.1109/ICRA.2014.6907626
7. F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution," Proc. IEEE **83**, 858–876 (1995). https://doi.org/10.1109/5.387089
8. X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A Multilevel Successive Elimination Algorithm for block matching motion estimation," IEEE Trans. Image Process. **9**, 501–504 (2000). https://doi.org/10.1109/83.826786
9. S.-Y. Huang, C.-Y. Cho, and J.-S. Wang, "Adaptive fast block-matching algorithm by switching search patterns for sequences with wide-range motion content," IEEE Trans. Circuits Syst. Video Technol. **15**, 1373–1384 (2005). https://doi.org/10.1109/TCSVT.2005.856931
10. ITU-T Rec. H.263, "Video coding for low bit rate communication," v1 (1995); v2 (1998); v3 (2000).
11. ITU-T Rec. H.264 and ISO/IEC 14496–10 (MPEG4-AVC), "Advanced video coding for generic audiovisual services," v1 (2003); v2 (2004); v3 (with FRExt) (2004); v4 (2005).
12. J. Jain and A. Jain, "Displacement measurement and its application in interframe image–coding," IEEE T. Commun. **29**, 1799–1808 (1981). https://doi.org/10.1109/TCOM.1981.1094950
13. J.-J. Tsai and H.-M. Hang, "On adaptive pattern selection for block motion estimation algorithms," in *IEEE Int. Conf. on Acoustics, Speech Signal Processing–ICASSP'07, Honolulu, 2007* (IEEE, 2007), pp. 1173–1176. https://doi.org/10.1109/ICASSP.2007.366122
14. J.-Bin Xu, L.-M. Po, and C.-K. Cheung, "Adaptive motion tracking block matching algorithms for video coding", IEEE Trans. Circuits Syst. Video Technol. **9**, 1025–1029 (1999). https://doi.org/10.1109/76.795056
15. J.-H. Jung, H.-S. Lee, J. H. Lee, and D.-J. Park, "A novel template matching scheme for fast full-search boosted by an integral image," IEEE Signal Process. Lett. **17**, (2010). https://doi.org/10.1109/LSP.2009.2032452
16. J.-N. Kim, D.-K. Kang, S.-C. Byun, I.-L. Lee, and B.-H. Ahn, "A fast full-search motion estimation algorithm using sequential rejection of candidates from hierarchical decision structure," IEEE Trans. Broadcasting **48**, 43–46 (2002). https://doi.org/10.1109/11.992854
17. J T. Koga, K. linuma, 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.
18. C.-M. Kuo, Y.-H. Kuan, C.-H. Hsieh, and Y.-H. Lee, "A novel prediction-based directional asymmetric search algorithm for fast block-matching motion estimation," IEEE Trans. Circuits Syst. Video Technol. **19**, 893–899 (2009).
19. S. C. Kwatra, C-M Lin, and W. A. Whyte, "An adaptive algorithm for motion compensated color image coding," IEEE Trans. Commun. **35**, 747–754 (1987). https://doi.org/10.1109/TCOM.1987.1096840
20. L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol. **6**, 313–317 (1996). https://doi.org/10.1109/76.499840
21. L.-M. Po, K.-H. Ng, K.-W. Cheung, K.-M. Wong, Y. Md. Salah Uddin, and C.-W. Ting, "Novel directional gradient descent searches for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol. **19**, 1189–1195 (2009). https://doi.org/10.1109/TCSVT.2009.2020320
22. L.-M. Po, C.-W. Ting, K.-M. Wong, and K.-H. Ng, "Novel point oriented inner searches for fast block motion estimation," IEEE Trans. Multimedia **9**, 9–15 (2007). https://doi.org/10.1109/TMM.2006.886330
23. J.-B. Lee and H. Kalva, *The VC–1 and H.264 Video Compression Standards for Broadband Video Services*, Multimedia Systems and Applications (Springer, Boston, 2008). https://doi.org/10.1007/978-0-387-71043-3
24. 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 Syst. Video Technol. **3**, 85–87 (1993). https://doi.org/10.1109/76.180692
25. C.-H. Lee and L.-H. Chen, "A fast motion estimation algorithm based on the block sum pyramid," IEEE

Trans. Image Process. **6**, 1587–1591 (1997). https://doi.org/10.1109/83.641419

26. R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol. **4**, 438–442 (1994). https://doi.org/10.1109/76.313138
27. W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. Image Process. **4**, 105–107 (1995). https://doi.org/10.1109/83.350809
28. Y. Lin and Y. C. Wang, "Improved parabolic prediction-based fractional search for H.264/AVC video coding," Image Process. IET **3**, 261–271 (2009). https://doi.org/10.1049/iet-ipr.2008.0192
29. 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. **6**, 419–422 (1996). https://doi.org/10.1109/76.510936
30. S.-W. Liu, S.-D. Wei, and S.-H. Lai, "Fast optimal motion estimation based on gradient-based adaptive multilevel successive elimination," IEEE Trans. Circuits Syst. Video Technol. **18**, 156–160, (2008). https://doi.org/10.1109/TCSVT.2007.913973
31. L. Luo, C. Zou, X. Gao, and Z. He, "A new prediction search algorithm for block motion estimation in video coding," IEEE Trans. Consum. Electron. **43**, 56–61, (1997). https://doi.org/10.1109/30.580385
32. H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," Proc. IEEE **73**, 523–548, (1985). https://doi.org/10.1109/PROC.1985.13183
33. K.-H. Ng, L.-M. Po, and K.-M. Wong, "Search patterns switching for motion estimation using rate of error descent," in *IEEE Int. Conf. on Multimedia and Expo, Beijing, 2007* (IEEE, 2007), pp. 1583–1586. https://doi.org/10.1109/ICME.2007.4284967
34. K.-H. Ng, L.-M. Po, K.-M. Wong, C.-W. Ting, and K.-W. Cheung, "A search patterns switching algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol. **19**, 753–759 (2009). https://doi.org/10.1109/TCSVT.2009.2017414
35. Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block matching motion estimation," IEEE Trans. Image Process. **11**, 1442–1449 (2002). https://doi.org/10.1109/TIP.2002.806251
36. M. Nieuwenhuisen, S. Behnke, "Hierarchical planning with 3D local multiresolution obstacle avoidance for micro aerial vehicles," in *ISR/Robotik 2014; 41st Int. Symp. on Robotics, Munich, 2014* (VDE, 2014), pp. 1–7.
37. K. K. Ma and G. Qiu, "An improved adaptive rood pattern search for fast block-matching motion estimation in JVT/H.26L," in *Proc. of the 2003 Int. Symp. on Circuits and Systems, ISCAS'03, Bangkok, 2003* (IEEE, 2003), vol. 2, pp. 25–28. https://doi.org/10.1109/ISCAS.2003.1206072
38. P. I. Hosur and K. K. Ma, "Motion vector field adaptive fast motion estimation," in Second Int. Conf. on Information, Communications, and Signal Processing (ICICS'99), Singapore, *1999.*
39. R. Srinivasan and K.R. Rao, "Predictive coding based on efficient motion estimation," IEEE T. Commun. **33**, 888–896 (1985). https://doi.org/10.1109/TCOM.1985.1096398
40. L. Shen, Z. Zhang, Z. Liu, and W. Zhang, "An adaptive and fast fractional pixel search algorithm in H.264," Signal Process. **87**, 2629–2639 (2007). https://doi.org/10.1016/j.sigpro.2007.04.013
41. B. C. Song and K.-W. Chun, "Multi-resolution block matching algorithm and its VLSI architecture for fast motion estimation in an MPEG-2 video encoder," IEEE Trans. Circuits Syst. Video Technol. **14**, 1119–1137 (2004). https://doi.org/10.1109/TCSVT.2004.833161
42. J. Stuckler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3D modeling and tracking," J. Visual Commun. Image Representation **25**, 137–147 (2014). https://doi.org/10.1016/j.jvcir.2013.02.008
43. J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol. **8**, 369–377 (1998). https://doi.org/10.1109/76.709403
44. A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST) enhancing block based motion estimation," Proc. SPIE **4310**, 883–892 (2001). https://doi.org/10.1117/12.411871
45. A. M. Tourapis, "Optimization model version 1.0," in *ISO/IEC JTC1/SC29/WG11 MPEG2000/N3324* (Noordwijkerhout, Netherlands, 2000).
46. A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," Proc. SPIE **4671**, 1069–1079 (2002). https://doi.org/10.1117/12.453031
47. F. Varray and H. Liebgott, "Multi-resolution transverse oscillation in ultrasound imaging for motion estimation," IEEE Trans. Ultrason., Ferroelectr., Freq. Control **60**, 1333–1342 (2013). https://doi.org/10.1109/TUFFC.2013.2707
48. Y. Vatis and J. Ostermann, "Adaptive interpolation filter for H.264/AVC," IEEE Trans. Circuits Syst. Video Technol. **19**, 179–192 (2009). https://doi.org/10.1109/TCSVT.2008.2009259
49. T. Wedi, "Adaptive interpolation filters and high-resolution displacements for video coding," IEEE Trans. Circuits Syst. Video Technol. **16**, 484–491 (2006). https://doi.org/10.1109/TCSVT.2006.870856
50. H. Yin, H. Jia, H. Qi, X. Ji, X. Xie, and W. Gao, "A hardware-efficient multi-resolution block matching algorithm and its VLSI architecture for high definition MPEG-like video encoders," IEEE Trans. Circuits Syst. Video Technol. **20**, 1242–1254 (2010). https://doi.org/10.1109/TCSVT.2010.2058476
51. Y.-H. Ko, H.-S. Kang, and S.-W. Lee, "Adaptive search range motion estimation using neighboring motion vector differences," IEEE Trans. Consum. Electron. **57**, 726–730 (2011). https://doi.org/10.1109/TCE.2011.5955214
52. Y.-W. Huang, S.-Y. Chien, B.-Y. Hsieh, and L.-G. Chen, "Global elimination algorithm and architecture design for fast block matching motion estimation," IEEE Trans. Circuits Syst. Video Technol. **14**, 898–907

(2004).
https://doi.org/10.1109/TCSVT.2004.828321

53. J. Zan, M. O. Ahmad, M. N. S. Swamy, “A multiresolution motion estimation technique with indexing,” IEEE Trans. Circuits Syst. Video Technol. **16**, 157–165 (2006).
https://doi.org/10.1109/TCSVT.2005.857304

54. S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” IEEE Trans. Image Process. **9**, 287–290 (2000).
https://doi.org/10.1109/83.821744

55. C. Zhu, X. Lin, and L.-P. Chau, “Hexagon-based search pattern for fast block motion estimation” IEEE Trans. Circuits Syst. Video Technol. **12**, 349–355 (2002).
https://doi.org/10.1109/TCSVT.2002.1003474

56. C. Zhu, X. Lin, L. Chau, and L.-M. Po, “Enhanced hexagonal search for fast block motion estimation,” IEEE Trans. Circuits Syst. Video Technol. **14**, 1210–1214 (2004).
https://doi.org/10.1109/TCSVT.2004.833166

57. C. Zhu, W.-S. Qi, and W. Ser, “Predictive fine granularity successive elimination for fast optimal block matching motion estimation,” IEEE Trans. Image Process. **14**, 213–221 (2005).
https://doi.org/10.1109/TIP.2004.840702

58. B.-J. Zou, C. Shi, C.-H. Xu, and S. Chen, “Enhanced hexagonal-based search using direction-oriented inner search for motion estimation,” IEEE Trans. Circuits Syst. Video Technol. **20**, 156–160 (2010).
https://doi.org/10.1109/TCSVT.2009.2031461

Dr. K. Srinivas Rao is the Professor of Computer Science and Engineering Department at MLR Institute of Technology, Hyderabad, India. He obtained his PhD in Computer Science and Engineering from Anna University, Tamilnadu, India. He received his BTech and MTech degrees from Osmania University, Hyderabad. He is having more than 20 yr of teaching and research experience. His current researches are in the fields of data mining, image processing, and big data analytics.

Dr. A. V. Paramkusam is a Professor of Electronics and Communication Engineering at Lendi Institute of Engineering and he obtained his PhD in Electronics and Communication Engineering from the JNTU Hyderabad, India in 2015. He received BE and ME degrees in Electronics and communication from Andhra University, Visakhapatnam, India, in 1996 and 2004, respectively. His research interests include image compression, video coding, and video water marking. He is a life member of ISTE. He has outstanding contribution with 36 publications in the national, IEEE International Conferences, and reputed international journals (Springer and IET). He is reviewer for SCI and Scopus indexed journals.