3
Direction-Oriented Motion Search Algorithm

As discussed in Chapter 2, the motion estimation process consumes significant computations and time in video compression. This chapter aims to develop efficient and effective motion search algorithms for motion estimation. Hence, a fast motion estimation algorithm is proposed to address three main problems. First, how to dynamically switch SR to achieve improved matching performance for blocks present at the object boundaries? Second, how to identify suitable search patterns for exploiting direction-oriented motion trajectory for computational efficiency and which has lower chances of becoming trapped in the local minimum. The third one is to find out an optimal threshold value for partial distortion computations.

The proposed work targets to address the problems mentioned above. The detailed discussion on challenges in the traditional motion search algorithms is presented in the next section. Our three major contributions in this chapter are:

- Dynamic switching between SR with adaptive SR dimension,
- New direction-oriented search patterns, and
- Speed-up mechanism with optimal threshold value based partial distortion computation

In this work, the first problem is addressed by introducing a dynamic switch between SR based on the location of the predicted initial search center. The second problem is addressed by the proposed four number of direction-oriented search patterns: horizontal and vertical wings-diamond search patterns, and two $\pm 45^{\circ}$ inclined hexagon-shaped search patterns. For the third problem, we present a method for the selection of optimal threshold values based on the distortion statistics for different partial distortion calculations. The experiments with varying motion videos: slow, medium, fast, and directional motion content are performed to evaluate the performance of the proposed algorithm against competitive methods.

The remainder of the chapter is organized as follows. Section 3.1 summarizes the MV distribution statistics. The proposed direction-oriented motion search algorithm is described in Section 3.2. Experimental setup, results, and performance comparisons between the proposed algorithm and state-of-the-art methods are presented in Section 3.3. Section 3.4 summarizes the chapter.

3.1 MOTION VECTOR DISTRIBUTION CHARACTERISTICS

The probability distribution (PDF) of MV is popularly used in the selection of appropriate search patterns for fast convergence. For example, in CDHS [Cheung and Po, 2005], 97% of blocks are found to possess cross-center biased MV. To demonstrate the behavior of MV distribution, FS is employed on sixteen video sequences, each consisting of different motion contents. The details of the test video sequences used for the study are given in Table 3.1.

In literature, video sequences are traditionally classified into three categories: slow, medium, and fast based on the motion content. After a thorough analysis of MV distribution of each video

sequence, 'Akiyo', 'Mother-Daughter', 'News', and 'Discussion' are classified as slow-motion content sequences, 'Carphone', 'Foreman', 'Coastguard', and 'Crowd run' are classified as medium-motion content sequences, and 'Stefan', 'Football', 'Cheerleaders', and 'Soccer' are classified as fast-motion content sequences. However, some video sequences are observed to follow dominant motion in only one of the possible directions: horizontal, vertical, or diagonal. For example, 'Shields' video sequence contains significant motion in the horizontal direction. Hence, the proposed study focused on test video sequences with four motion categories: slow, medium, fast, and directional. Among them, 'Harbour', 'Park joy', 'Shields', and 'ReadySteadyGo' are classified as directional-motion content sequences.

The PDF and the cumulative distribution (CDF) of MV corresponding to the slow and medium motion category are shown in Figure 3.1. For slow-motion sequences, it is observed that more than 80% of MV are $(0,0)$ and about 93% of MV lie within Chebyshev distance of 2. Similarly, for medium motion sequences, the major part of the distribution of MV is located around $(0,0)$. However, Figure 3.2 indicates that this is not the case for fast and directional motion sequences. For fast motion sequences, about 33% of MVs are $(0,0)$, but the remaining MVs are scattered in all the directions. On the other hand, for directional motion sequences, the distribution of MV is observed to be scattered mostly in the horizontal direction, for test sequences considered in this work. About 70% of MV for directional motion content lie within a vertical Chebyshev distance of 1 from the center $(0,0)$.

Conventional methods of finding of MV for a candidate block use either default location $(0,0)$ or median location for initial search center. For this, the median value of MVs of adjacent blocks (left, top, and top-right) as shown in Figure 3.3 is computed. Let a MV corresponding to adjacent block B_k is denoted as $mv_k = (x_k, y_k)$. Then, the median value of MVs of adjacent blocks is computed as follows:

$$(x_m, y_m) = median(mv_1, mv_2, mv_3) \tag{3.1}$$

CDF of MV distribution shown in Figs. 3.1 (c), (f) and Figs. 3.2 (c), (f) illustrates the advantage provided by the selection of median (x_m, y_m) as initial search center over traditional approach of choosing $(0,0)$ as search center. It is observed that the median (x_m, y_m) is more useful for test video sequences containing higher motion content. Although the selection of median (x_m, y_m) as initial search center helps in a rapid convergence of the search process, this approach severely hits the matching accuracy for blocks located at the object boundaries, as these blocks may follow different motion trajectory than its neighboring blocks. The proposed method addresses this issue by dynamically selecting a better search region with a search center corresponding to the SAD minimum.

3.2 PROPOSED EFFICIENT DIRECTION-ORIENTED SEARCH

This section introduces the new approach for efficient block matching. It consists of mainly three parts: A) dynamic switch between SR with adaptive SR dimension, B) selecting direction-oriented search patterns, followed by C) introducing efficient partial distortion measure.

3.2.1 Dynamic Switch between SR with Adaptive SR Dimension

In literature, starting point is considered as 1) $(0,0)$ or 2) median (x_m, y_m). It is a well-known fact that the median start point contributes to faster convergence. However, we propose to select the best starting point among the two. The selection of the best starting point is based on distortion (SAD) minimum. It has resulted in the novel idea of switching between two different search regions.

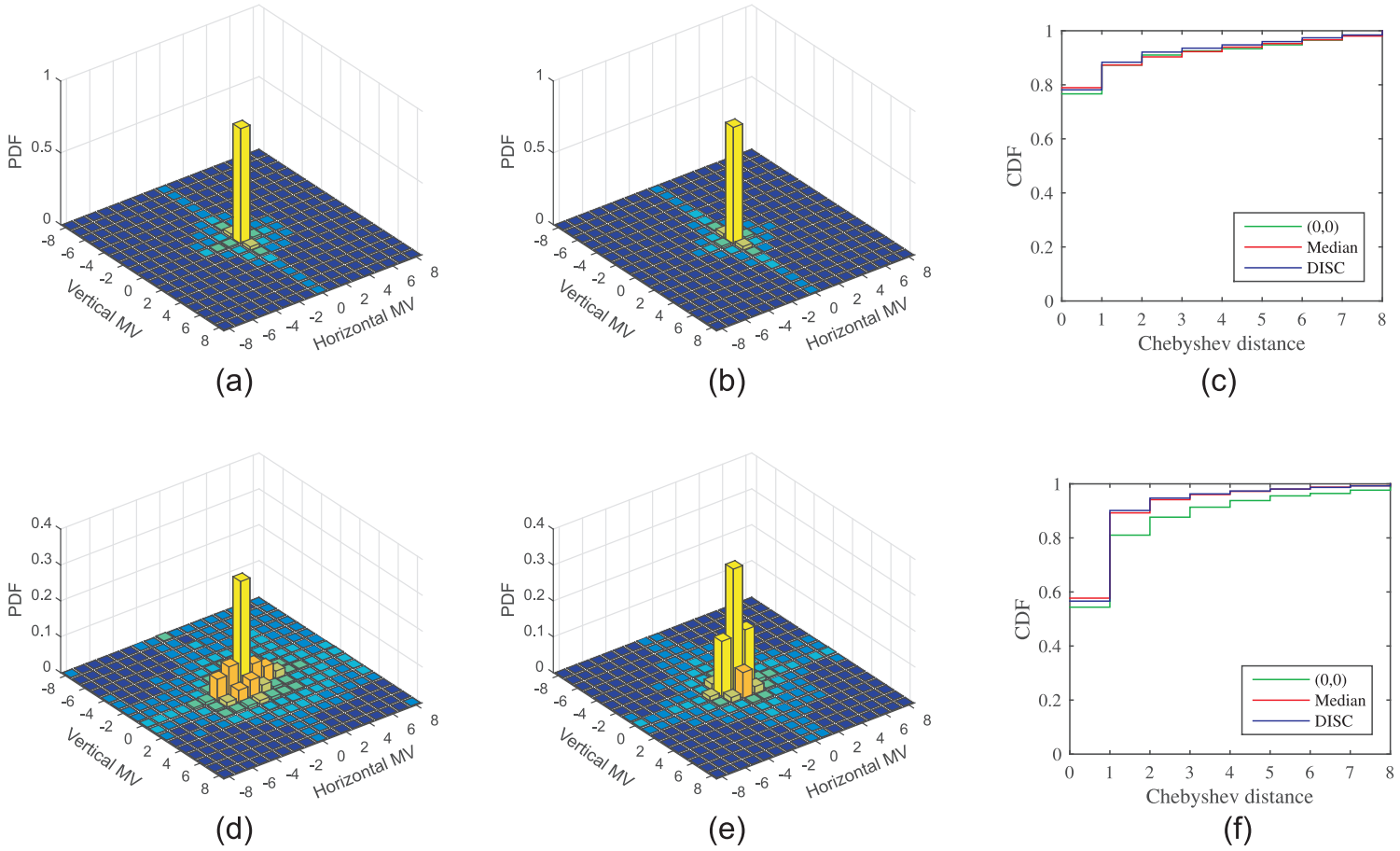

Figure 3.1 : Average MV distribution characteristics for slow and medium motion video sequences: (a) Average MV distribution for slow motion video sequences, (b) Average MV distribution for slow motion video sequences after dynamically selecting SR based on location of DISC, (c) CDF of MV distribution for slow motion video sequences, (d) Average MV distribution for medium motion video sequences, (e) Average MV distribution for medium motion video sequences after dynamically selecting SR based on location of DISC, (f) CDF of MV distribution for medium motion video sequences.

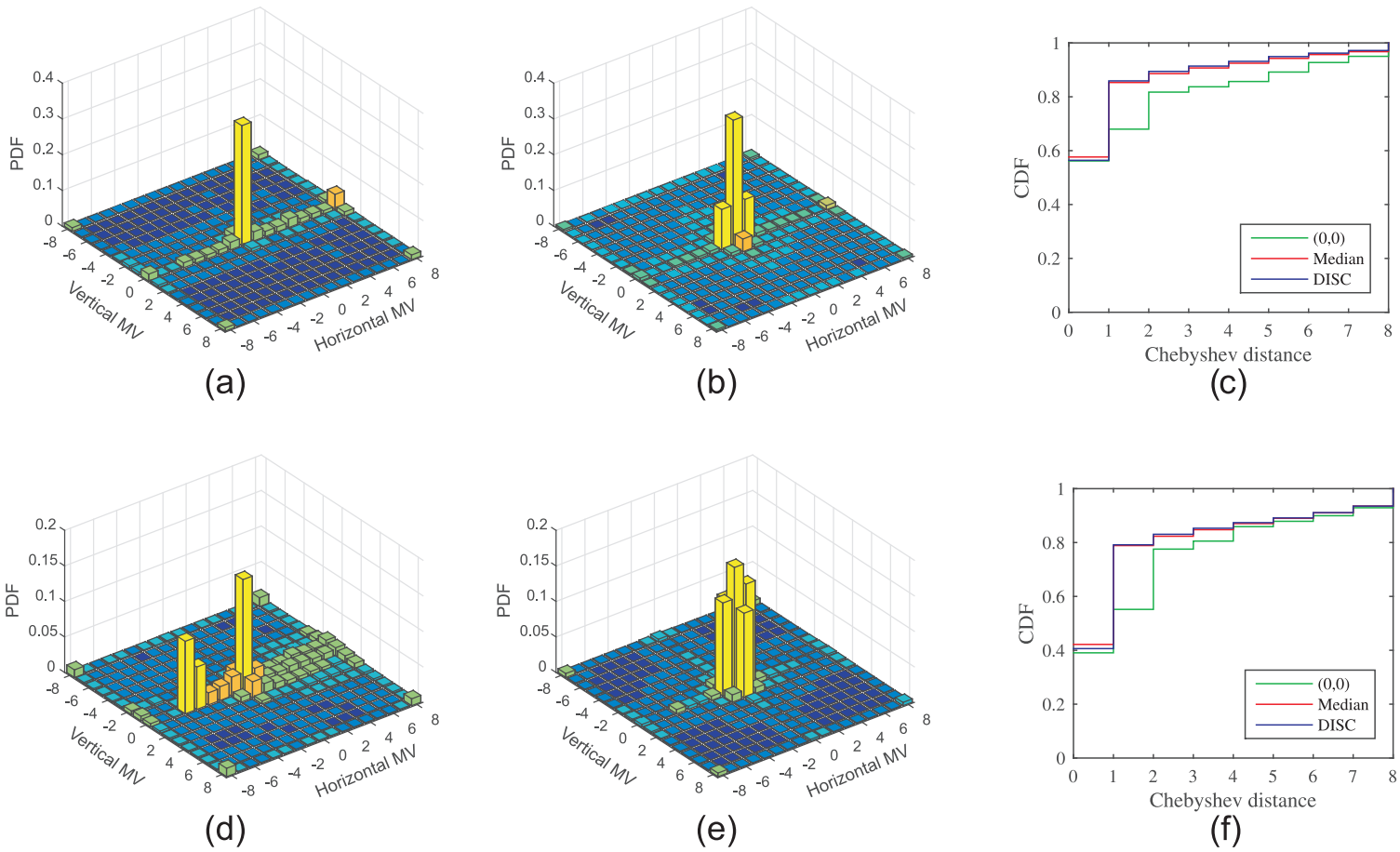

Figure 3.2 : Average MV distribution characteristics for fast and directional motion video sequences: (a) Average MV distribution for fast motion video sequences, (b) Average MV distribution for fast motion video sequences after dynamically selecting SR based on location of DISC, (c) CDF of MV distribution for fast motion video sequences, (d) Average MV distribution for directional motion video sequences, (e) Average MV distribution for directional motion video sequences after dynamically selecting SR based on location of DISC, (f) CDF of MV distribution for directional motion video sequences.

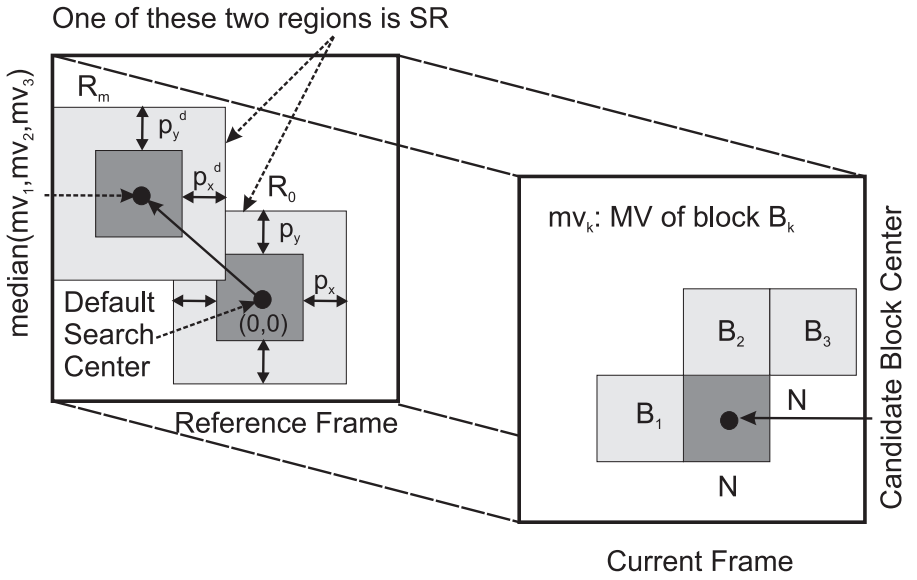

Figure 3.3 : Illustration of proposed block matching algorithm with dynamically switched SR for candidate block corresponding to the minimum SAD location: $(0,0)$ or (x_m, y_m).

In the proposed work the SAD is obtained at two locations: 1) $(0,0)$ and 2) (x_m, y_m) to select different SRs. In Figure 3.3, two different SRs corresponding to SR center locations: 1) $(0,0)$ and 2) (x_m, y_m) are represented as R_0 and R_m. One of these SR centers, where SAD is minimum, is treated as dynamically chosen initial search center (DISC). Then, among two different SRs: R_0 and R_m, SR with DISC being its center is chosen as shown in Figure 3.3. The dynamic selection of SRs in this way ensures following bound on SAD is satisfied.

$$\min_{MV \in R_0 \cup R_m} SAD \leq \min_{MV \in R_0} SAD \tag{3.2}$$

Before moving ahead, let us understand the idea behind considering two different SRs. As FS comprehensively search for optimal motion vectors in a fixed search space, it is expected to provide optimal performance. However, it should be noted that the search space is already fixed using the search start center and search range. However, we have tweaked the search start centers in the proposed approach, and hence it created a new opportunity to explore different search spaces at the same time. The FS mechanism considers search space to be either R_0 or R_m based on the search center. There can be only one fixed search space at a time. For example, if our search space is R_0 for a traditional FS approach, then it could find the best matching block only in R_0. However, we propose to select the best starting point among the R_0 and R_m. The selection of the best starting point is based on distortion (SAD) minimum. It has resulted in the novel idea of switching between two different search regions. It would provide us an advantage over traditional approaches that consider fixed search space. It is also supported by Eq. (3.2). In a nutshell, FS with R_0 would be most likely to provide sub-optimal performance as compared to the proposed search, which selects the best search space among R_0 and R_m.

PDF of MV distribution shown in Figs. 3.1 (b), (e) and Figs. 3.2 (b), (e) depict that dynamically chosen SR using DISC, R_0 or R_m, always outperformed traditionally chosen single SR R_0. Figure 3.1 and Figure 3.2 clearly indicates that most of the MVs are distributed around the newly chosen initial search center, DISC. CDF of MV distribution shown in Figs. 3.1 (c), (f) and Figs. 3.2 (c), (f) illustrates the

comparison between these approaches. This mechanism provided a remarkable increment of 1.92%, 24.23%, 26.81%, and 43.46% in the proportion of MVs falling within Chebyshev distance of 1 for slow, medium, fast, and directional motion content videos, respectively.

In general, SR dimension $p = (p_x, p_y)$ is kept fixed. However, to reduce the total number of search points, it is suggested to keep SR dimension adaptive while maintaining the same quality. For DISC $= (x_m, y_m)$, the predicted MV is highly likely to be located near the DISC location. Hence, a limited dimension search area could also suffice. This reasoning holds good if the candidate block has similar motion characteristics as that of adjacent blocks. However, variability in motion characteristics of adjacent blocks demands a higher dimension search area. For this, the SR dimension is chosen to be adaptive as follows:

$$p^d = (p_x^d, p_y^d) \tag{3.3}$$

where p_x^d and p_y^d denotes maximum adaptive displacement in horizontal and vertical directions, respectively. It is computed as follows:

$$\begin{aligned} p_x^d &= \min(p_x, max(|x_m - x_1|, |x_m - x_2|, |x_m - x_3|)) \\ p_y^d &= \min(p_y, max(|y_m - y_1|, |y_m - y_2|, |y_m - y_3|)) \end{aligned} \tag{3.4}$$

It should be noted that adaptive search range is bounded such that:

$$p_x^d \leq p_x, p_y^d \leq p_y \tag{3.5}$$

The dynamic switch between SR based on SAD value drastically improved the block matching accuracy. Moreover, the proposed adaptive search range selection ensured the fast convergence of the block matching process without compromising on performance. This type of adaptive SR dimension selection provided around 90% reduction in search points for DISC $= (x_m, y_m)$. However, the SR dimension is kept fixed for DISC $= (0, 0)$ considering the chances of deterioration in matching performance with an adaptive approach.

3.2.2 Direction-Oriented Search Patterns

It is reported that most of the fast block matching algorithms converge to SAD minimum rapidly, but they suffer from becoming trapped in the local minimum [Lin *et al.*, 2009]. It is mainly due to the smaller size of search patterns used in the algorithms such as DS [Zhu and Ma, 2000]. To reduce the chances of becoming trapped in the local minimum, we propose to add more appropriate points to traditional patterns. Moreover, the second issue of exploiting directional motion characteristics of a video sequence is addressed by introducing direction-oriented search patterns shown in Figure 3.4. For tracking horizontal and vertical movements, we propose new search patterns by adding wings to the small diamond search pattern (SDSP) [Zhu and Ma, 2000] in horizontal and vertical directions, respectively. These patterns are termed as horizontal wings diamond search pattern (HWDS) and vertical wings diamond search pattern (VWDS), shown in Figs. 3.4 (f), (g) respectively. Similarly, for diagonal movements, $\pm 45^\circ$ inclined hexagon-shaped search patterns (IHS) are shown in Figs. 3.4 (h), (i).

To compute predicted MV, intermediate MV's (IMV) are used, and for this, five SADs are obtained at all the dark circle points of SDSP shown in Figure 3.4 (a). The center of the SDSP is treated

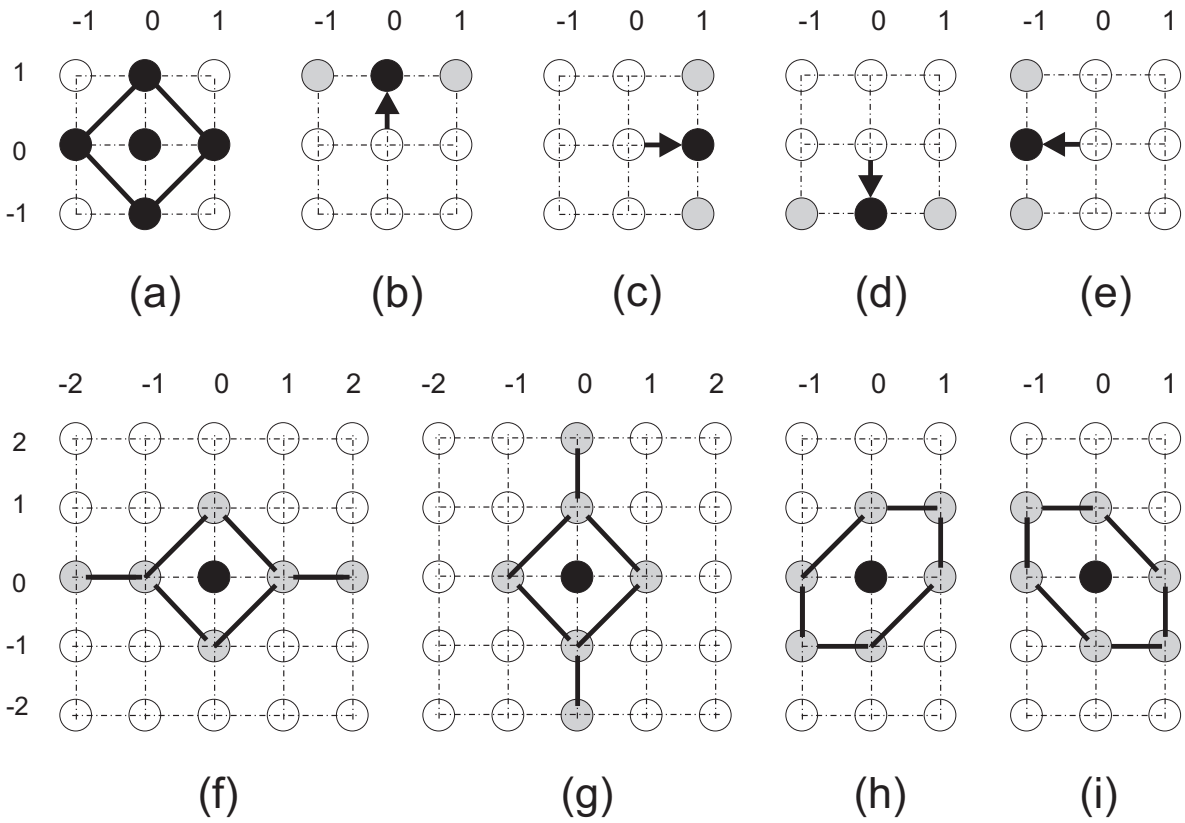

Figure 3.4 : Search patterns: (a) 1^{st} step pattern: SDSP, (b) 2^{nd} step north directional pattern, (c) 2^{nd} step east directional pattern, (d) 2^{nd} step south directional pattern, (e) 2^{nd} step west directional pattern, (f) 3^{rd} step onward horizontal directional pattern: HWDS, (g) 3^{rd} step onward vertical directional pattern: VWDS, (h) 3^{rd} step onward inclined directional pattern: $+45^{\circ}$ IHS, (i) 3^{rd} step onward inclined directional pattern: -45° IHS.

as IMV of the first step. If SAD at the center is the minimum of the five SADs, the search is terminated and predicted MV would be the IMV itself. In subsequent steps, the IMV will be a point in a search pattern, where SAD is a minimum. In the second step, one of the vertices of the SDSP where SAD is minimum is used in further processing for a finding of IMV. A three-point search pattern with this vertex at its center is chosen from four possible patterns, shown in Figs. 3.4 (b-e). Now SADs are obtained at the neighboring points of the vertex also. In this way, three SADs, at the chosen pattern points, are available. The point, where SAD is minimum, of these three SADs, is treated as IMV of the second step.

Since the total number of search steps for the predicted MV may be more than two, a generalized description of the search process is given as follows. For i^{th} search step, for $i > 2$, the new pattern selection is based on the coordinate difference of the IMVs obtained in $(i-1)^{th}$ and $(i-2)^{th}$ search steps. Let coordinates of IMV point at the i^{th} search step be (x_i,y_i). Let coordinate difference of IMVs of the $(i-1)^{th}$ and $(i-2)^{th}$ search steps be denoted as $(\Delta x_i, \Delta y_i)$, such that:

$$(\Delta x_i, \Delta y_i) = (x_{i-1} - x_{i-2}, y_{i-1} - y_{i-2}) \tag{3.6}$$

A point with the coordinate of $(\Delta x_i, \Delta y_i)$ could take any location with Manhattan length ≤ 2, shown in Figure 3.5. These coordinates are shown in a way that depicts one of the patterns in Figs. 3.4

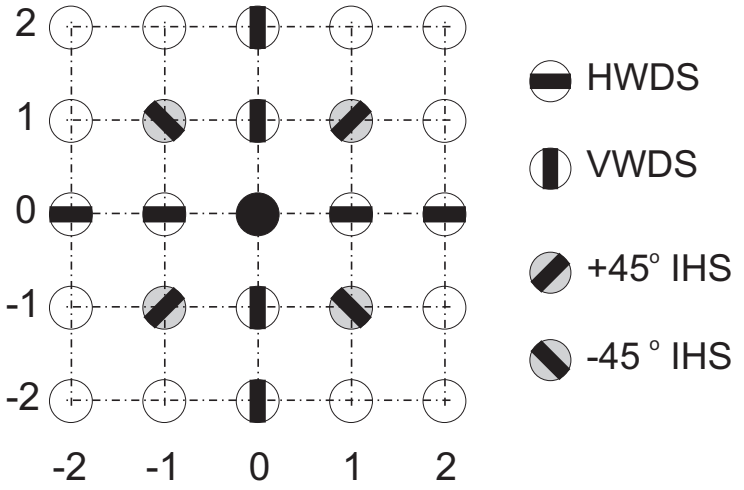

Figure 3.5 : Selection of one of the four direction-oriented search patterns based on the coordinate location $(\Delta x_i, \Delta y_i)$.

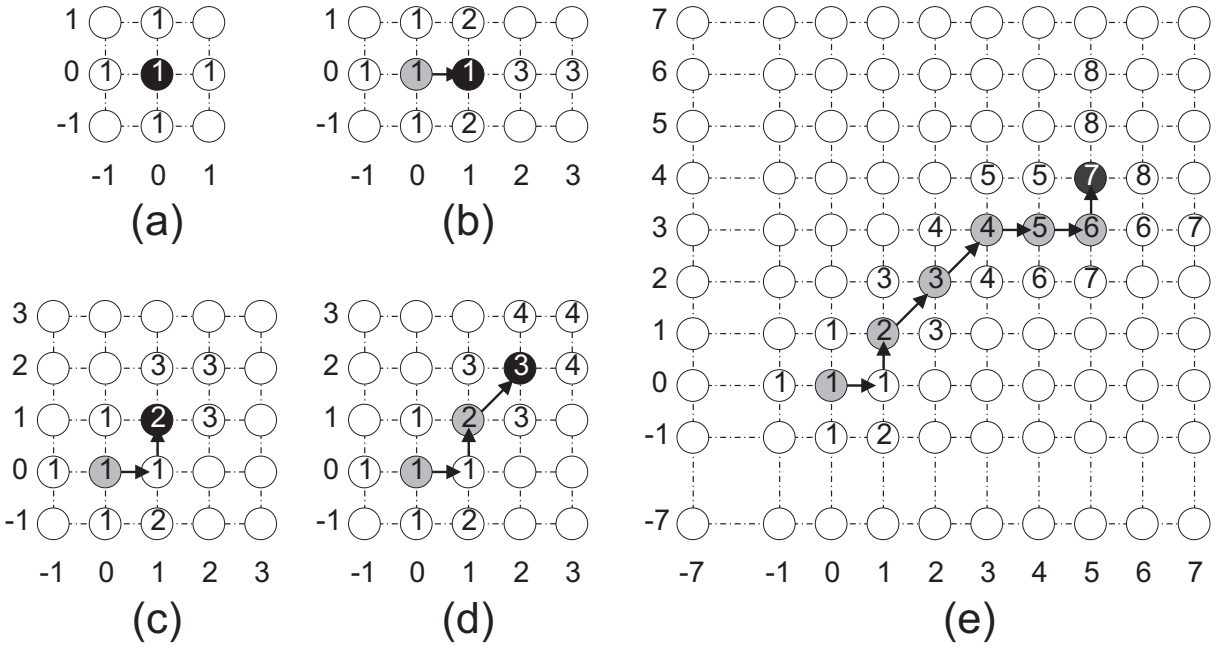

Figure 3.6 : Typical search path examples: (a) 1^{st} step search stop with predicted MV $(0,0)$, (b) 3^{rd} step search stop with predicted MV $(1,0)$, (c) 3^{rd} step search stop with predicted MV $(1,1)$, (d) 4^{th} step search stop with predicted MV $(2,2)$, (e) 8^{th} step search stop with predicted MV $(5,4)$. Note: The checked search points are marked with the corresponding step number. The IMV and predicted MV is represented in the gray and dark circle respectively.

(f-i), to be selected for the IMV. The search for IMV is terminated if $(\Delta x_i, \Delta y_i) = (0,0)$ as further search attempts will bring no change to IMV. Otherwise, the search is repeatedly continued until the selected search pattern reaches the search region boundary.

The path traversed by the IMV of each search step is termed as the search path. Five typical search path examples are shown in Figs. 3.6 (a-e). In each sub-figure, the checked search points are

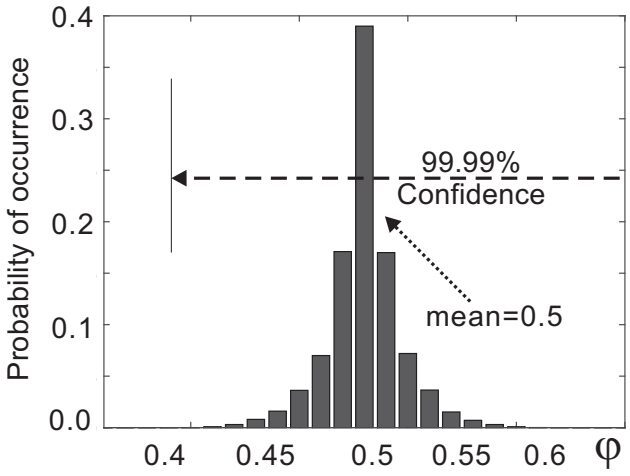

Figure 3.7 : Ratio of SADs obtained with sub-sampling and without sub-sampling and denoted by φ, follows Gaussian distribution.

marked with the corresponding step number. Figure 3.6 (a) shows that search of predicted MV stops in one step, and the MV is $(0,0)$ representing zero (no) motion while, Figure 3.6 (b) shows that search of predicted MV stops in three steps and the MV is $(1,0)$ representing horizontal (directional) motion. However, Figs. 3.6 (c), (d) have MVs representing the diagonal motion. The search stops in eight steps, shown in Figure 3.6 (e) clearly depicts the use of $+45^{\circ}$ IHS in the third and fourth step, HWDS in the fifth and sixth step, and VWDS in seventh and eighth step representing the diagonal, horizontal and vertical motions respectively. These patterns are effective for tracking directional motion content as compared to conventional algorithms [8, 9, 14, 17, 18], mainly because search is carried out in the minimum SAD value direction.

3.2.3 Efficient Partial Distortion Measure

For $1:\beta$ sub-sampling, the threshold value $T_{1:\beta}^{new} = T/\beta$ is popularly used in literature [Seidel *et al.*, 2015]. Hence, for $1:2$ sub-sampling, i.e. $\beta = 2$, the new threshold value $T_{1:2}^{new} = 0.5 \times T$ is considered. However, we observed an erroneous search termination at this scaling factor. Instead, another approach is explored in our work for calculating $T_{1:\beta}^{new}$ based on distortion statistics. It is evident that $T_{1:\beta}^{new}$ depends on the ratio of SADs obtained with sub-sampling and without sub-sampling and denoted by φ and computed as:

$$\varphi = \frac{SAD_{1:\beta}}{SAD_{1:1}} \tag{3.7}$$

It is experimentally observed that, φ follows a Gaussian distribution with mean $\mu = 0.5$, when FS with $1:2$ sub-sampled SAD is employed. The distribution shown in Figure 3.7 depicts the same. A significance level of α=0.0001 is obtained at a scaling factor $s = 0.4$, i.e. $T_{1:2}^{new} = 0.4 \times T$ minimizes erroneous search termination with 99.99% confidence. For similar reasons, in $1:4$ sub-sampling, i.e. $\beta = 4$, the new threshold value $T_{1:4}^{new} = 0.2 \times T$ is chosen instead of $T_{1:4}^{new} = 0.25 \times T$.

Table 3.1 : Test video sequences used in the study.

Index	Sequence Name	Resolution	Total Frames	Frame Rate	Database	Motion Type
1	Akiyo	352×288	300	25	[YUV, 2013]	Slow
2	Mother Daughter	352×288	300	25	[YUV, 2013]	
3	News	352×288	300	25	[YUV, 2013]	
4	Discussion	640×480	300	25	[Vranješ *et al.*, 2013]	
5	Carphone	352×288	382	25	[YUV, 2013]	Medium
6	Foreman	352×288	300	25	[YUV, 2013]	
7	Coastguard	352×288	300	25	[YUV, 2013]	
8	Crowdrun	704×576	250	25	[De Simone *et al.*, 2010]	
9	Stefan	352×288	300	25	[YUV, 2013]	Fast
10	Football	352×288	260	25	[YUV, 2013]	
11	Cheerleaders	640×480	300	25	[Vranješ *et al.*, 2013]	
12	Soccer	704×576	300	30	[De Simone *et al.*, 2010]	
13	Harbour	704×576	300	30	[De Simone *et al.*, 2010]	Directional
14	Park Joy	704×576	250	30	[De Simone *et al.*, 2010]	
15	Shields	768×432	500	50	[Seshadrinathan *et al.*, 2010a]	
16	ReadySteadyGo	1920×1080	600	120	[Group, 2012]	

3.3 EXPERIMENTS AND ANALYSIS

3.3.1 Test Video Sequences

Sixteen test video sequences with varying motion content and of different resolutions are chosen from benchmark databases [YUV, 2013; Vranješ *et al.*, 2013; De Simone *et al.*, 2010; Seshadrinathan *et al.*, 2010a; Group, 2012; Xiph, 2012]. All test sequences are in YUV $4{:}2{:}0$ (uncompressed) color format. Only the luminance (Y) component of each test sequence is used for motion estimation. Table 3.1 shows the details of test sequences used for experimental analysis. In Figure 3.8, each row corresponds to one of the four different motion categories: slow, medium, fast, and directional motion content video sequences. Test video sequences with a frame rate higher than 30 are temporally sub-sampled for fast motion estimation. For this, video sequences: Shields and ReadySteadyGo are temporally sub-sampled by a factor of 2 and 4, respectively, to bring the effective frame rate to about 30. The first 150 frames of each sequence are considered for experimentation.

3.3.2 Evaluation Metrics

Prediction quality: Objective quality metrics such as Peak-Signal-to-Noise-Ratio (PSNR) and Structural Similarity Index (SSIM) are considered for the performance evaluation. SSIM is considered to be consistent with human visual perception than PSNR [Wang *et al.*, 2004]. PSNR can be computed as:

$$PSNR = 10\log_{10}\left(\frac{V_{\max}^2}{MSE}\right) \tag{3.8}$$

where V_{max} is the maximum pixel intensity for the given k-bit resolution such that $V_{max} = 2^k - 1$. For example, a bit-depth of 8 has V_{max} equal to 255. As PSNR correlates poorly with perceptual video quality; it would be interesting to use, the latest video quality metrics such as Video Multi-Method Assessment Fusion (VMAF) [Li *et al.*, 2016] for comparisons.

Search complexity: In general, the efficiency of the selected block matching algorithm (BMA) is determined by an average number of search points traversed in the search process. But sub-sampling for partial SAD computation ensured a lower number of pixels used in each SAD calculation. To incorporate both: the number of search points and the proportion of pixels used for computing partial SAD, we have introduced a new parameter: the average number of checked pixels per candidate block (ANCPB). ANCPB determines the efficiency of selected BMA. The ANCPB not only depends on the

Figure 3.8 : Representative frames of the test video sequences used in the analysis.

average number of search points but also on the proportion of pixels used for computing the partial SAD value for each search point. The ANCPB for $1:\beta$ sub-sampled SAD is computed as:

$$ANCPB_{1:\beta} = \frac{\text{Average Number of Search Points}}{\beta} \tag{3.9}$$

For performance comparisons, speed-up over FS is computed for all fast block matching algorithms. The speed-up is defined as ratio of computational complexity of full search algorithm and fast search algorithm and it is computed as:

$$Speed-up = \frac{ANCPB_{FS}}{ANCPB_{fast\_algorithm}} \tag{3.10}$$

3.3.3 Experimental settings

Search parameters: The candidate block of size 16×16, maximum search displacement parameter $p = \pm 8$ and SAD as a distortion error measure, is considered in our experiments. For slow and

Table 3.2 : Performance comparisons in terms of PSNR, SSIM, and ANCPB for slow motion sequences.

Sequence	Akiyo			Mother Daughter			News			Discussion			Average		
Algorithm	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB
FS	42.85	0.993	289.00	39.55	0.967	289.00	36.75	0.978	289.00	38.85	0.990	289.00	39.50	0.982	289.00
DS	42.83	0.993	13.09	39.45	0.967	14.32	36.50	0.978	13.73	38.81	0.990	14.48	39.40	0.982	13.90
HS	42.71	0.992	11.04	39.26	0.965	11.77	36.21	0.977	11.39	38.71	0.990	11.86	39.22	0.981	11.52
CDHS	42.62	0.992	5.13	39.02	0.964	6.40	36.16	0.976	5.78	38.56	0.989	6.29	39.09	0.980	5.90
ARPS	42.82	0.992	5.35	39.45	0.967	6.82	36.44	0.978	6.12	38.79	0.990	6.63	39.38	0.982	6.23
TZS	42.85	0.993	29.20	39.52	0.967	32.89	36.67	0.978	30.23	38.84	0.990	33.03	39.47	0.982	31.34
EDOS(B)	42.80	0.992	1.69	39.35	0.966	3.22	36.38	0.977	2.81	38.73	0.989	2.59	39.31	0.981	2.58
EDOS(A+B)	42.81	0.992	1.60	39.41	0.966	2.81	36.44	0.977	2.51	38.77	0.990	2.38	39.36	0.981	2.32
$EDOS_{1:2}^{0.6}$	42.81	0.992	0.74	39.36	0.966	1.26	36.41	0.977	1.17	38.75	0.990	1.00	39.33	0.981	1.04
$EDOS_{1:2}^{0.5}$	42.81	0.992	0.80	39.39	0.966	1.40	36.42	0.977	1.25	38.76	0.990	1.19	39.34	0.981	1.16
$EDOS_{1:2}^{0.4}$	42.81	0.992	0.88	39.40	0.966	1.61	36.43	0.977	1.36	38.77	0.990	1.59	39.35	0.981	1.36
$EDOS_{1:4}^{0.3}$	42.77	0.992	0.37	39.31	0.965	0.63	36.27	0.976	0.59	38.72	0.989	0.50	39.26	0.981	0.52
$EDOS_{1:4}^{0.25}$	42.78	0.992	0.40	39.33	0.966	0.70	36.26	0.977	0.62	38.72	0.990	0.59	39.27	0.981	0.58
$EDOS_{1:4}^{0.2}$	42.78	0.992	0.44	39.34	0.966	0.81	36.26	0.977	0.68	38.72	0.990	0.78	39.28	0.981	0.68

Table 3.3 : Performance comparisons in terms of PSNR, SSIM, and ANCPB for medium motion sequences.

Sequence	Carphone			Foreman			Coastguard			Crowdrun			Average		
Algorithm	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB
FS	33.27	0.933	289.00	31.67	0.923	289.00	28.06	0.874	289.00	22.25	0.864	289.00	28.81	0.899	289.00
DS	33.10	0.932	15.94	31.47	0.920	16.94	27.96	0.868	17.91	21.95	0.851	19.31	28.62	0.893	17.53
HS	32.67	0.927	12.60	31.15	0.912	13.28	27.15	0.843	12.25	21.85	0.846	14.64	28.20	0.882	13.19
CDHS	32.64	0.925	8.17	30.89	0.909	9.06	27.26	0.841	9.37	21.29	0.815	12.02	28.02	0.872	9.65
ARPS	33.03	0.931	8.46	31.40	0.920	9.19	28.05	0.873	9.02	22.03	0.855	11.09	28.63	0.895	9.44
TZS	33.22	0.933	32.02	31.61	0.923	31.71	28.06	0.874	28.93	22.20	0.863	31.58	28.77	0.898	31.06
EDOS(B)	32.99	0.930	7.69	31.32	0.918	9.53	27.89	0.865	12.27	21.79	0.843	14.35	28.50	0.889	10.96
EDOS(A+B)	33.05	0.931	6.06	31.43	0.921	6.90	28.05	0.874	7.34	22.09	0.858	9.11	28.65	0.896	7.35
$EDOS_{1:2}^{0.6}$	33.00	0.930	2.74	31.39	0.918	3.25	28.04	0.873	3.68	22.06	0.857	4.55	28.62	0.895	3.56
$EDOS_{1:2}^{0.5}$	33.02	0.930	3.02	31.41	0.920	3.45	28.04	0.873	3.69	22.06	0.857	4.56	28.63	0.895	3.68
$EDOS_{1:2}^{0.4}$	33.03	0.930	3.41	31.43	0.921	3.64	28.04	0.873	3.69	22.06	0.857	4.56	28.64	0.896	3.83
$EDOS_{1:4}^{0.3}$	32.59	0.926	1.34	31.30	0.917	1.63	27.96	0.871	1.87	21.91	0.851	2.29	28.44	0.891	1.78
$EDOS_{1:4}^{0.25}$	32.60	0.927	1.49	31.33	0.919	1.72	27.96	0.871	1.87	21.91	0.851	2.29	28.45	0.892	1.84
$EDOS_{1:4}^{0.2}$	32.61	0.927	1.70	31.35	0.920	1.83	27.96	0.871	1.88	21.91	0.851	2.29	28.46	0.892	1.92

medium motion video sequences, average SAD values lie in the range of $600 \sim 1200$; hence a threshold value of $T = 512$ is conservatively considered. The $1:2$ and $1:4$ sub-sampling is employed for partial SAD calculations with $T_{1:2}^{new} = 0.4 \times T = 205\ (rounded)$ and $T_{1:4}^{new} = 0.2 \times T = 102\ (rounded)$ respectively.

Test setup for rate-distortion comparison: The test sequences listed in Table 5.1, are also used for rate-distortion (RD) comparisons between different block matching algorithms. In this pursuit, all the block matching algorithms, including the proposed algorithm, are integrated into simple video encoder and simulated under QP values of 22, 27, 32, and 37. For comparing the average difference between RD curves, the Bjontegaard's method created [Bjontegaard, 2001] is used to calculate the average Bit-rate and PSNR.

All the block matching algorithms are implemented to best of our knowledge and understanding in MATLAB 2015a running on 64-bit Windows 7 platform with Intel Xeon(R) CPU E5-2650 v2 @ 2.60 GHz with 32.0 GB RAM.

Table 3.4 : Performance comparisons in terms of PSNR, SSIM, and ANCPB for fast motion sequences.

Sequence	Stefan			Football			Cheerleaders			Soccer			Average		
Algorithm	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB
FS	23.86	0.865	289.00	23.73	0.712	289.00	21.25	0.788	289.00	23.93	0.808	289.00	23.19	0.793	289.00
DS	23.23	0.835	18.39	22.98	0.671	23.81	20.61	0.773	18.24	22.89	0.762	22.27	22.43	0.760	20.68
HS	22.77	0.813	13.27	22.98	0.663	16.28	20.71	0.774	13.67	22.83	0.745	14.90	22.32	0.749	14.53
CDHS	22.67	0.812	11.11	22.51	0.644	16.91	20.38	0.764	10.94	22.55	0.745	16.01	22.03	0.741	13.74
ARPS	23.66	0.857	9.04	23.33	0.692	14.37	20.98	0.781	11.40	23.63	0.796	12.52	22.90	0.781	11.83
TZS	23.83	0.864	26.59	23.49	0.705	25.30	21.03	0.784	30.75	23.74	0.801	22.28	23.02	0.788	26.23
EDOS(B)	22.91	0.822	11.49	22.93	0.662	20.26	20.89	0.775	13.29	23.34	0.762	19.44	22.52	0.755	16.12
EDOS(A+B)	23.60	0.865	7.31	23.76	0.730	14.23	21.45	0.792	11.75	26.28	0.877	11.78	23.77	0.816	11.27
$EDOS_{1:2}^{0.6}$	23.56	0.863	3.59	23.71	0.720	7.01	21.41	0.791	5.67	26.18	0.872	5.83	23.72	0.811	5.52
$EDOS_{1:2}^{0.5}$	23.56	0.863	3.64	23.73	0.720	7.02	21.41	0.791	5.79	26.22	0.873	5.84	23.73	0.812	5.57
$EDOS_{1:2}^{0.4}$	23.57	0.864	3.71	23.74	0.721	7.25	21.41	0.791	5.94	26.24	0.874	5.86	23.74	0.812	5.69
$EDOS_{1:4}^{0.3}$	23.31	0.853	1.80	23.51	0.710	3.38	21.19	0.783	2.72	25.89	0.857	2.88	23.48	0.801	2.69
$EDOS_{1:4}^{0.25}$	23.32	0.853	1.83	23.51	0.710	3.39	21.19	0.783	2.77	25.92	0.859	2.90	23.49	0.801	2.72
$EDOS_{1:4}^{0.2}$	23.33	0.854	1.87	23.51	0.710	3.40	21.19	0.783	2.85	25.95	0.860	2.91	23.49	0.802	2.76

Table 3.5 : Performance comparisons in terms of PSNR, SSIM, and ANCPB for directional motion sequences.

Sequence	Harbour			Parkjoy			Shields			ReadySteadyGo			Average		
Algorithm	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB	PSNR	SSIM	ANCPB
FS	27.55	0.928	289.00	20.62	0.832	289.00	27.80	0.950	289.00	19.57	0.705	289.00	23.89	0.854	289.00
DS	27.42	0.926	16.03	18.98	0.680	23.43	25.98	0.904	22.06	18.93	0.661	24.43	22.83	0.793	21.49
HS	27.03	0.919	12.86	19.05	0.689	16.80	25.25	0.878	15.46	19.02	0.667	16.77	22.59	0.788	15.47
CDHS	27.04	0.917	8.61	18.48	0.640	16.60	24.79	0.872	16.39	18.70	0.645	17.94	22.25	0.769	14.88
ARPS	27.36	0.925	8.94	20.30	0.808	11.43	27.62	0.947	9.46	19.43	0.693	15.71	23.67	0.843	11.38
TZS	27.51	0.928	32.00	20.57	0.830	18.98	27.79	0.950	23.58	19.29	0.685	25.75	23.79	0.848	25.08
EDOS(B)	27.32	0.924	10.54	18.51	0.637	18.77	24.96	0.870	18.13	19.29	0.677	20.61	22.52	0.777	17.01
EDOS(A+B)	27.38	0.926	7.90	20.47	0.828	9.39	27.62	0.947	7.08	21.15	0.767	16.13	24.15	0.867	10.12
$EDOS_{1:2}^{0.6}$	27.37	0.926	3.92	20.35	0.822	4.61	27.56	0.946	3.53	21.11	0.764	7.75	24.10	0.864	4.95
$EDOS_{1:2}^{0.5}$	27.37	0.926	3.95	20.36	0.822	4.62	27.57	0.946	3.55	21.11	0.765	7.79	24.10	0.865	4.98
$EDOS_{1:2}^{0.4}$	27.37	0.926	3.97	20.36	0.822	4.63	27.57	0.946	3.55	21.12	0.765	7.84	24.10	0.865	5.00
$EDOS_{1:4}^{0.3}$	27.28	0.924	1.97	20.07	0.806	2.30	27.42	0.943	1.80	20.91	0.753	3.74	23.92	0.856	2.45
$EDOS_{1:4}^{0.25}$	27.28	0.924	1.99	20.07	0.806	2.31	27.44	0.943	1.80	20.92	0.754	3.76	23.93	0.857	2.46
$EDOS_{1:4}^{0.2}$	27.28	0.924	2.00	20.07	0.806	2.31	27.44	0.943	1.81	20.93	0.755	3.78	23.93	0.857	2.47

3.3.4 Results and Comparisons for the Proposed and Existing Methods

The performance of the proposed Efficient Direction-Oriented Search (EDOS) method is compared against six popular block matching algorithms: FS, DS [Zhu and Ma, 2000], HS [Zhu *et al.*, 2002], CDHS [Cheung and Po, 2005], ARPS [Nie and Ma, 2002] and TZS [JCT-VC, 2013]. Overview of these state-of-the-art algorithms is summarized in Table 2.1. The EDOS algorithm consists of mainly three parts: A) dynamic switch between SR with adaptive SR dimension, B) selecting direction-oriented search patterns, followed by C) efficient partial distortion measure. In this study, EDOS(B), and EDOS(A+B) is evaluated independently to analyze the effect of each part. Moreover, multiple variants of EDOS(A+B+C) with different sub-sampling parameters ($1:\beta = \{1:2, 1:4\}$) and respective threshold values are considered. For this, six sub-sampled variants of EDOS: $EDOS_{1:\beta}^{s} = \{EDOS_{1:2}^{0.6}, EDOS_{1:2}^{0.5}, EDOS_{1:2}^{0.4}, EDOS_{1:4}^{0.3}, EDOS_{1:4}^{0.25}, EDOS_{1:4}^{0.2}\}$, where β is the sub-sampling factor and s represents scaling factor corresponding to the selected sub-sampling factor, such that threshold value $T_{1:\beta}^{s} = s \times T$.

The PSNR and SSIM of the video sequences reconstructed by the different block matching algorithms are listed in Table 3.2. It is reported that FS provides the best matching quality for the predefined search region at the cost of the highest ANCPB values. Experimental evaluation of the proposed method reported negligible decrements in PSNR for slow and medium motion videos as

compared to FS. On the other hand, an average increment in PSNR of about 0.58 dB and 0.26 dB is observed for fast and directional motion videos, respectively. All fast block matching algorithms [Zhu and Ma, 2000; Zhu *et al.*, 2002; Cheung and Po, 2005; Nie and Ma, 2002; JCT-VC, 2013] provided similar effectiveness for slow-motion videos. DS, HS, and CDHS severely fail to provide good block matching quality for fast and directional motion videos, while the proposed method is observed to be superior for these motion categories. However, the performance of TZS is observed to be superior to DS, HS, CDHS, and ARPS. It is apparent from Table 3.2 that the PSNR is in the order FS $>$ EDOS(A+B) $>$ $\text{EDOS}_{1:2}^{0.4}$ $>$ $\text{EDOS}_{1:2}^{0.5}$ $>$ $\text{EDOS}_{1:2}^{0.6}$ $>$ $\text{EDOS}_{1:4}^{0.2}$ $>$ $\text{EDOS}_{1:4}^{0.25}$ $>$ $\text{EDOS}_{1:4}^{0.3}$ $>$ TZS $>$ ARPS $>$ EDOS(B) $>$ DS $>$ HS $>$ CDHS for high motion content video sequences. Hence, an overall PSNR performance for the proposed method is evaluated over FS. The PSNR performance shown in Figure 3.9 (a) clearly indicates that EDOS(A+B) outperformed other competitive algorithms. A similar trend as of PSNR results is observed for SSIM values as well. The increment of 8.8% in SSIM is observed for directional video sequence 'ReadySteadyGo', largely due to the part A of the proposed EDOS method.

FS is the computationally complex method and requires $(2p{+}1)^2$ search points for the single MV prediction. It is evident from Table 3.2 that the ANCPB is in the order FS $>$ TZS $>$ DS $>$ HS $>$ EDOS(B) $>$ CDHS $>$ ARPS $>$ EDOS(A+B) $>$ $\text{EDOS}_{1:2}^{0.4}$ $>$ $\text{EDOS}_{1:2}^{0.5}$ $>$ $\text{EDOS}_{1:2}^{0.6}$ $>$ $\text{EDOS}_{1:4}^{0.2}$ $>$ $\text{EDOS}_{1:4}^{0.25}$ $>$ $\text{EDOS}_{1:4}^{0.3}$. An overall speed-up for the fast BMA is evaluated over FS. Figure 3.9 (b) indicates that EDOS(A+B) provided speed-up of about 125, 45, 25, and 35 times for slow, medium, fast, and directional motion video sequences respectively. EDOS(A+B) and EDOS(B) provided significant speed-up without any degradation in PSNR. It should be noted that both EDOS(A+B) and EDOS(B) provided similar performance for slow and medium motion sequences. However, EDOS(A+B) outperformed EDOS(B) for the remaining test sequences due to the presence of the fast and directional motion content in them. EDOS(A+B) could directly start the search process from the highly probable vicinity of true MV and hence provided better results.

Moreover for fair comparison, sub-sampling ($1:\beta = \{1:2, 1:4\}$) is also applied to six popular BMA with $s = \{0.5, 0.25\}$, respectively resulting into twelve more combinations. Due to space limitation, the results for sub-sampled versions of six popular BMA are not tabulated in Table 3.2. However, comparisons in terms of curves are shown in Figure 3.9. The performance of sub-sampled variants of EDOS is separately compared against respective sub-sampled versions of popular BMA. In our experiments, we have considered three different threshold values for each partial SAD computation. For example, for $1:2$ sub-sampled partial SAD computation $s = \{0.6, 0.5, 0.4\}$ is considered resulting into three variants of EDOS: $\{\text{EDOS}_{1:2}^{0.6}$, $\text{EDOS}_{1:2}^{0.5}$, $\text{EDOS}_{1:2}^{0.4}\}$. The PSNR and speed-up performance for $1:2$ and $1:4$ sub-sampled algorithms are shown in Figs. 3.9 (c), (d) and Figs. 3.9 (e), (f) respectively. The superiority of EDOS over other BMA is also observed after employing partial SAD computations. Our idea of choosing $s = 0.4$ for $1:2$ sub-sampled SAD computation provided slightly better PSNR results as compared to other threshold values. However, this is achieved at the cost of slight compromise in speed-up. Threshold value with $s = 0.4$ avoided early erroneous termination in the search process as compared to $s = 0.6$ and hence resulted in better PSNR values. The variation in speed-up for different threshold values is higher for slow-motion sequences. However, for fast and directional motion content, the speed-up values are indistinguishable. Hence, $\text{EDOS}_{1:2}^{0.4}$ is preferred over $\text{EDOS}_{1:2}^{0.6}$ and $\text{EDOS}_{1:2}^{0.5}$ due to better PSNR results. A similar analysis could be extended for $1:4$ algorithms.

Next, to study the effect of different sub-sampled versions of EDOS, all the variants of EDOS are analyzed together in Figs. 3.10 (a), (b). The PSNR for EDOS(A+B) is highest among different variants of EDOS, whereas EDOS(B) performed worst. It is also observed that degradation in PSNR increases with an increase in sub-sampling from $1:2$ to $1:4$. However, speed-up is doubled with the increase in sub-sampling from $1:2$ to $1:4$. Hence, our algorithm could be used for different high-speed applications if slight PSNR degradation is acceptable.

Overall, FS, TZS, EDOS(A+B) provided superior results than other competitive methods.

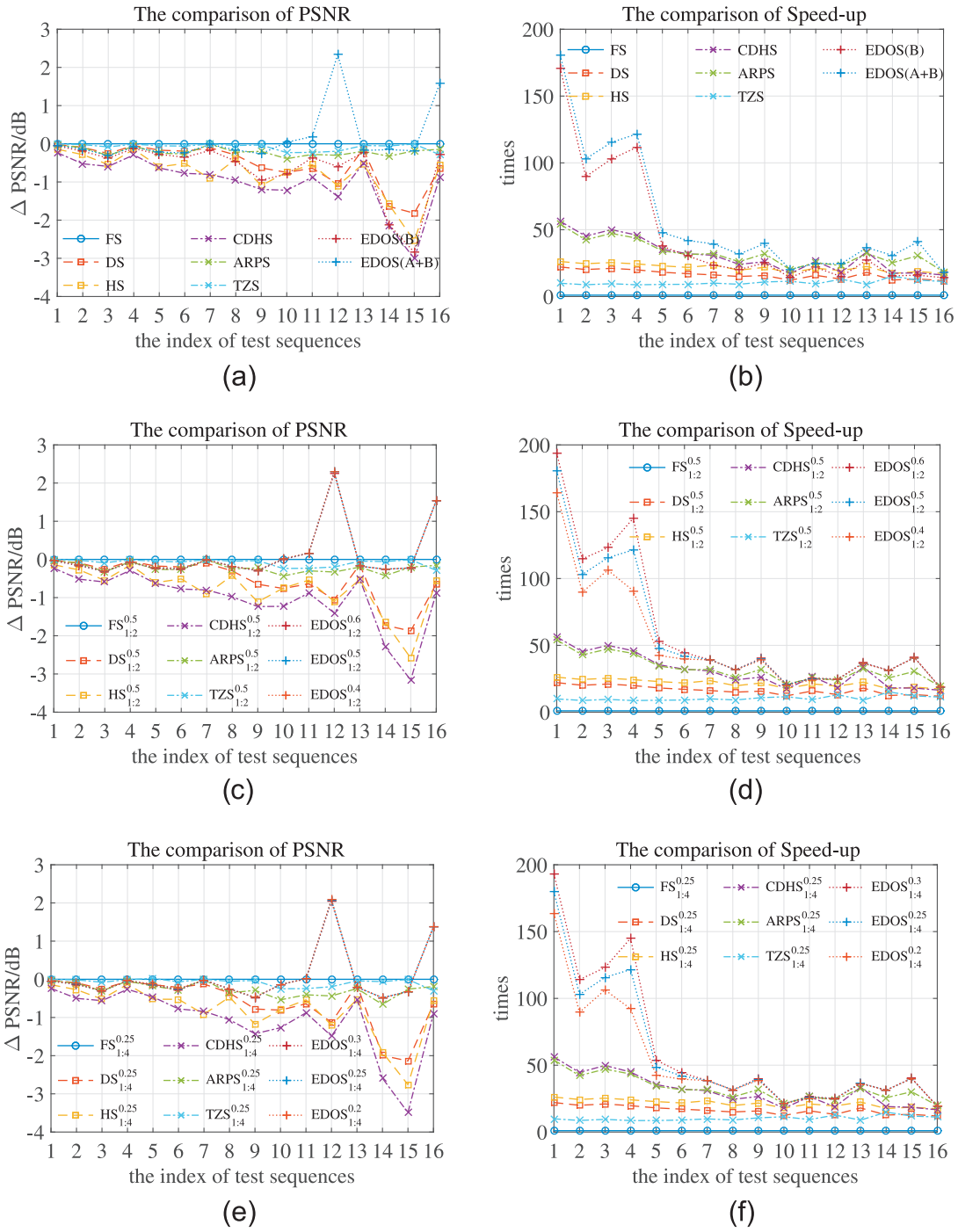

Figure 3.9 : Performance comparison in terms of PSNR and speed-up for proposed method against different block matching algorithms: (a) PSNR comparison for different algorithms without sub-sampling, (b) Speed-up comparison for different algorithms without sub-sampling, (c) PSNR comparison for different algorithms with 1 : 2 sub-sampling, (d) Speed-up comparison for different algorithms with 1 : 2 sub-sampling, (e) PSNR comparison for different algorithms with 1 : 4 sub-sampling, (f) Speed-up comparison for different algorithms with 1 : 4 sub-sampling.

However, for directional motion sequences, the proposed method obtained better PSNR and SSIM values. ANCPB values clearly dictate the superiority of all the variants of EDOS. However, it is unfair

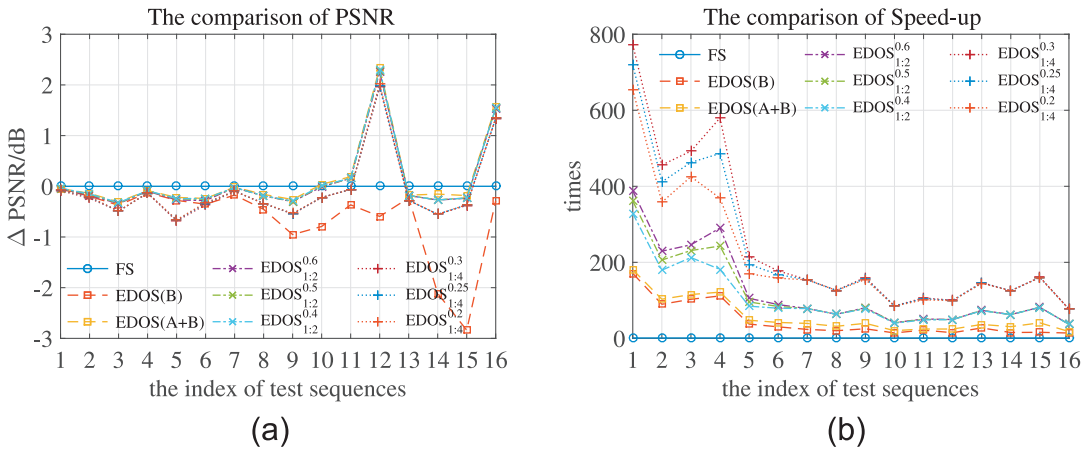

Figure 3.10 : (a) PSNR comparison for different variants of proposed method against FS, (b) Speed-up comparison for different variants of proposed method against FS.

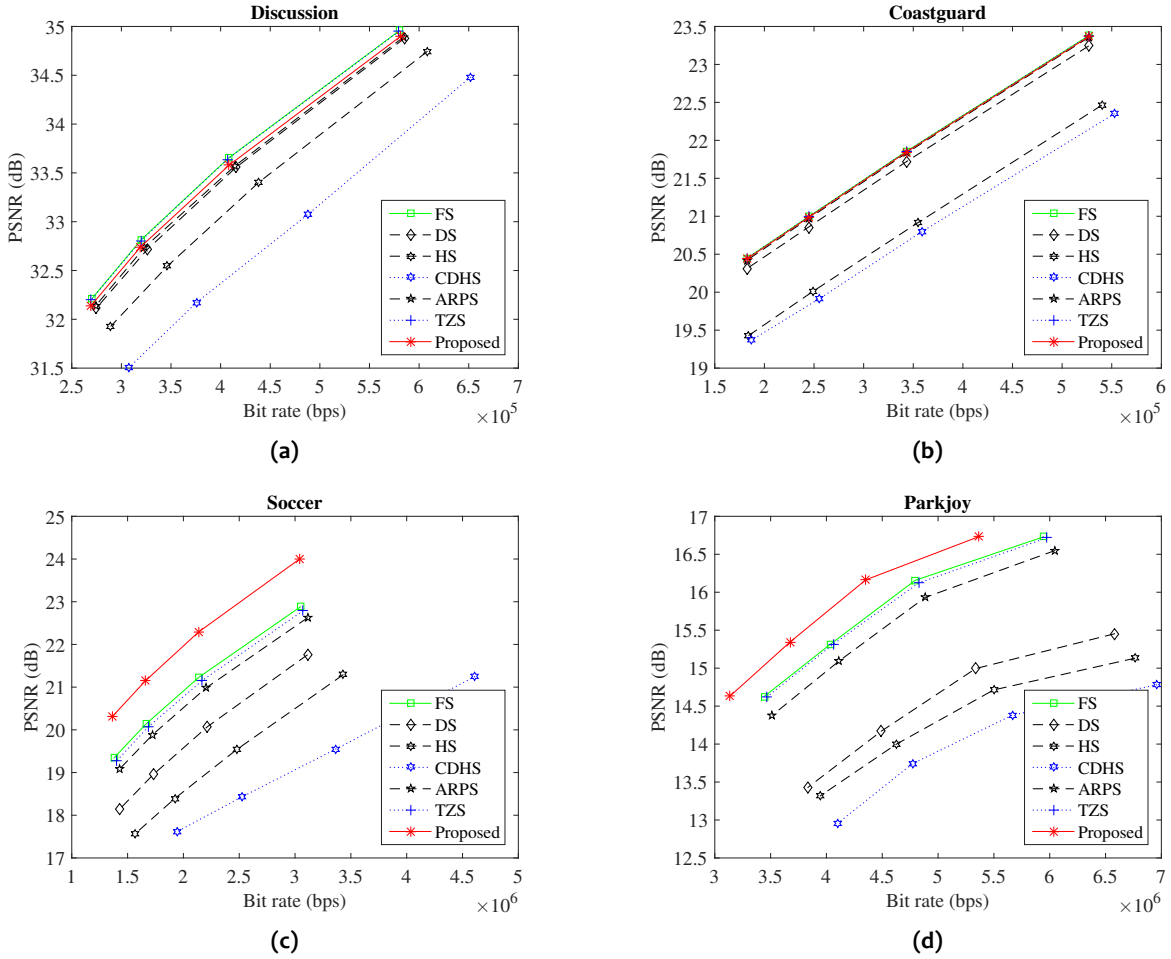

Figure 3.11 : Rate-distortion performance for video sequences (a) Discussion (slow motion), (b) Coastguard (medium motion), (c) Soccer (fast motion), and (d) Parkjoy (directional motion) under different QP values of 22, 27, 32, and 37.

to compare partial distortion measure incorporated search algorithm with the full distortion measure algorithm. For this, the speed-up of sub-sampled variants of EDOS is compared against sub-sampled versions of popular BMA. Figure 3.9 clearly depicts that all the variants of the proposed method outperformed other fast block matching algorithms. It is observed that the benefits of the proposed algorithm are significant at the lower-sized SR since dynamic switching between SRs at high SR dimension could result only in a minor change in SAD minimum values. Hence, the results for $p = \pm 8$ are discussed here.

Figure 3.11 presents the RD performance of the proposed and competitive algorithms for video sequences: Discussion (representing slow motion), Coastguard (representing medium motion), Soccer (representing fast motion), and Parkjoy (representing directional motion). The RD performance is computed under different QP values of 22, 27, 32, and 37. It is evident from Figure 3.11 that, increase in PSNR is proportional to an increase in bit-rate. For the slow-motion video sequence, the FS algorithm provided the best RD performance, followed by the proposed algorithm. On the other hand, the RD performance for medium, fast, and directional motion sequences indicate the superiority of the proposed algorithm over existing algorithms. One may observe that our algorithm outperformed not only to state-of-the-art TZS but also to the FS algorithm. The gains for fast and directional motion sequences are higher than those of medium motion sequences due to the adaptability of our direction oriented search patterns. This indicates that the proposed algorithm is well suited for fast and directional motion sequences.

3.3.5 Bit Requirement for MV in the Proposed Work

The bit requirement for MV depends on the MV distribution. An average information content termed as entropy, is measured using probability distribution as follows:

$$E(MV) = -\sum_{i} P_i(MV) \log(P_i(MV)) \tag{3.11}$$

where $P_i(MV)$ is the probability distribution of MV. The MV distribution shown in Figure 3.1 and Figure 3.2 indicates that entropy for slow motion videos is lower than the fast motion videos. Moreover, bound on SAD mentioned in (3.2) guarantees that entropy for proposed dynamically chosen SR is always lower than the fixed SR since uncertainty in the MV distribution shown in Figure 3.1 and Figure 3.2 decreases by dynamically switching SRs.

In the proposed method, MV for the candidate block is represented as relative displacement of the best-matched block from DISC rather than from the traditionally used default search center $(0,0)$. Since the decoder must be informed about the SR center, default, or median, one-bit information per block will be sent as overhead. Although this overhead seems to be low (for a block size of 16×16, overhead is $1/256\, bpp$), however, at low bandwidth requirements, reduction in this overhead is desired. For this, one bit overhead corresponding to each candidate block is sequentially arranged in the raster scan order, and then it is compressed by employing delta encoding [Schindler, 1970]. Delta encoding is the process of storing data in the form of differences between sequential data. This process reduced the overhead bit requirement from 1 bit/block to about 0.6 bits/block, depending on the correlation between the DISCs of the adjacent blocks.

Moreover, the bit requirement for encoding each MV depends on the dimension of SR. For improved MV prediction accuracy, particularly for fast and directional motion sequences, it is desired to have SR of higher dimension. However, the introduction of the dynamic switch between SR based on DISC location helped in no requirement of higher SR dimension, for the desired accuracy. Subsequently, the bit requirement for encoding each MV is also reduced. Hence, the overall bit requirement is further reduced.

3.4 SUMMARY

In this chapter, an efficient motion search algorithm is presented. The algorithm is developed based on the MV distribution characteristics of video sequences with varying motion content. The MV distribution provided a strong case for our proposal of direction-oriented search patterns. These patterns performed well for varying motion sequences as compared to the existing methods. Not only this, our method used the dynamic switch between the search region based on the SAD minimum location. This has helped us in a faster and more effective search convergence. Moreover, the proposed adaptive search range selection ensured the further improved convergence of the block matching process without compromising on performance. The computational complexity is further improved by a partial distortion scheme. We presented a novel threshold selection mechanism for effective partial SAD computations. The experiments were performed on a total of sixteen video sequences containing four sequences, each belonging to slow, medium, fast, and directional motion content. The experimental results indicate the efficacy of the proposed scheme is compared to the state-of-the-art methods. Our scheme achieved more than two times speed-up over the fastest state-of-the-art algorithm without any compromise in the matching accuracy. Not only this, our method even outperformed the FS method in terms of matching accuracy for fast and directional motion video sequences. This improvement in matching quality over FS along with reduced computational cost mainly comes from the proposed dynamic switch between SR based on the location of DISC, adaptive SR dimension selection, adaptability of direction-oriented search patterns, and optimal threshold value selection for partial SAD computations. With this method, the first objective of the Thesis, to develop an efficient and effective motion search algorithm is achieved. The proposed EDOS algorithm is very suitable for a wide range of applications, such as high-speed and high-quality video conferencing.

The video conferencing and surveillance videos are very important application areas. An efficient motion search algorithms presented in this chapter need to be fine-tuned for the specific type of video sequences such as surveillance videos. It is a well-known fact that surveillance videos contain a significantly large proportion of static blocks. These static blocks, generally, do not carry very important information. Hence, there is a scope to develop efficient and effective motion search mechanisms for surveillance videos. In this pursuit, the motion estimation scheme for surveillance videos is explored in the next chapter.

…