4

Background-Foreground Boundary aware Motion Search Algorithm for Surveillance Videos

As discussed in Chapter 3, the motion estimation process consumes significant computations and time in video compression. In the previous chapter, we have already developed efficient and effective motion search algorithms for motion estimation. However, these algorithms need to be fine-tuned for a specific type of video sequences. Surveillance video plays a vital role in today's security and navigation applications. The need for fine-tuned motion search algorithms to address computational efficiency challenges in surveillance videos is need of an hour. In this pursuit, the motion estimation scheme for surveillance videos is explored in this chapter. Traditionally, candidate blocks in the surveillance video are classified into two classes, namely, background and foreground. The background block belongs to the static regions in the surveillance video, whereas the foreground block contains moving regions. Hence, class-specific search strategies are further explored for efficient and effective motion search in surveillance videos.

The proposed work targets to address the problems mentioned above. The detailed discussion on problem analysis is presented in the next section. This chapter presents a novel background foreground boundary aware block matching algorithm. Our three major contributions in this chapter are:

- We propose a novel framework that can classify block into three categories based on the block characteristics.
- Under the framework, we introduce new search strategies for each category. The search strategies are specially tailored to exploit the characteristics of surveillance videos.
- We also introduce a novel speed-up mechanism for different categories. Our speed-up mechanism also helps in improving block matching accuracy.

In our work, firstly, the blocks are categorized into three classes, namely, background (BG), foreground (FG), and boundary (BD) blocks. The block classification process is based on block distortion values, and hence it is easy to implement. Secondly, the motion search is performed by employing different search strategies for each class. The zero-motion vector-based search is applied for background blocks. On the other hand, to exploit fast and directional motion characteristics of the boundary and foreground blocks, the eight rotating uni-wing diamond search patterns are proposed. The direction oriented search patterns help in faster motion search convergence. Thirdly, the speed-up is achieved through the novel region-based sub-sampled structure. The speed-up not only reduced the computational complexity but also improved block matching accuracy. An illustration of the traditional two-class block classification approach and our idea of three class block classification for surveillance video coding is shown in Figure 4.1.

The rest of the chapter is organized as follows: Section 4.1 discusses problems in existing block matching works for surveillance video coding applications. The proposed algorithm is described in Section 4.2. Experimental settings, results, and performance comparisons are presented in Section 4.3. Section 4.4 summarizes the chapter.

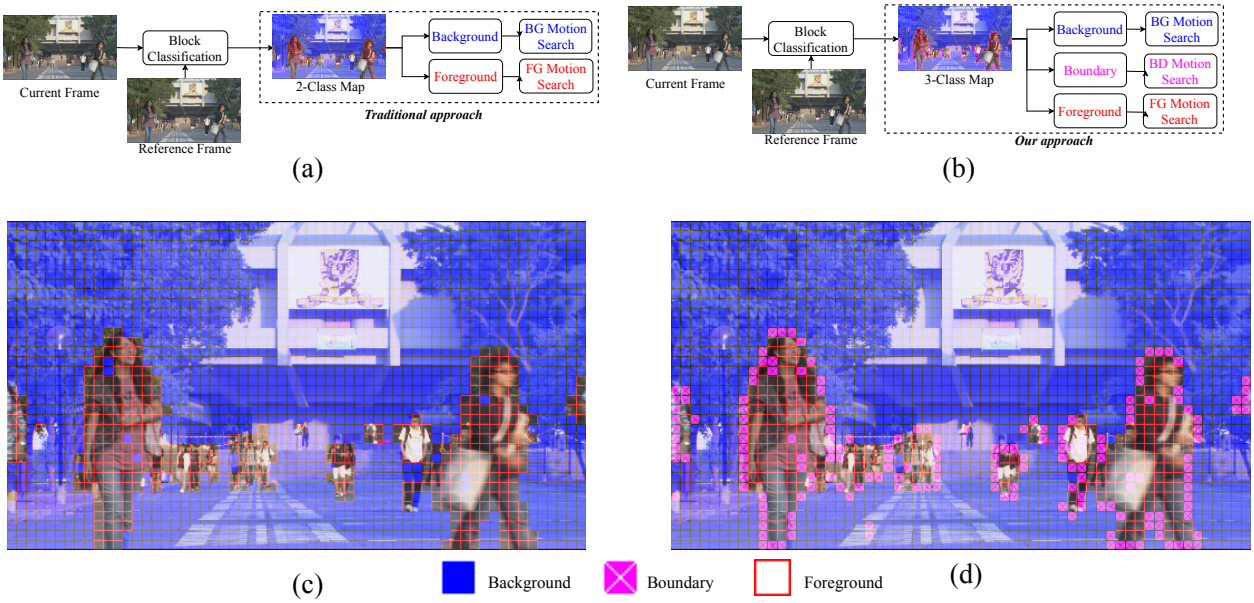

Figure 4.1 : (a) Illustration of traditional block classification approach with two classes: background and foreground, (b) proposed approach with three classes: background, foreground, and boundary, (c) two classes example (square sequence), (d) three classes example (square sequence). (Best viewed in color)

4.1 PROBLEM ANALYSIS

In this section, the characteristics of surveillance videos are explored, followed by problem analysis in existing works.

4.1.1 Characteristics of Surveillance Videos

The key characteristic of surveillance video lies in the high proportion of static blocks corresponding to the BG region. A large number of static blocks in the BG region provides a two-fold advantage: (1) better matching accuracy, and (2) faster motion search convergence. The static nature of BG regions has two direct implications: (1) the motion vectors (MV) for BG are concentrated near zero, and (2) the distortion values are smaller compared to FG regions.

The experiments were carried on twelve different surveillance test sequences taken from standard datasets [Gao *et al.*, 2014; Lab, 2019, 2011; Seshadrinathan *et al.*, 2010b]. The representative frames of each sequence are shown in Figure 4.11. The details are mentioned in Table 4.1. The MV distribution for BG and FG regions shown in Figure 4.2 (a) clearly depicts that over 95% of MVs for BG are within unit area distance from the center due to its static nature. On the other hand, the MVs for the FG region are dynamically distributed since FG always contain moving objects. The MV distribution for fast motion FG region indicates that about 30% of MVs lie within a unit area from zero-MV, but the rest of the MVs are present in all the directions. On the other hand, for the directional motion FG region, the MVs are mostly situated in a particular direction. About 70% of MVs are observed to lie along one specific distance path.

The better matching accuracy for static BG regions resulted in lower distortion values as compared to FG regions. The MAD distribution for BG and FG regions is shown in Figure 4.2 (c). The bi-modal MAD distribution for BG and FG regions can be further used to improve classification accuracy.

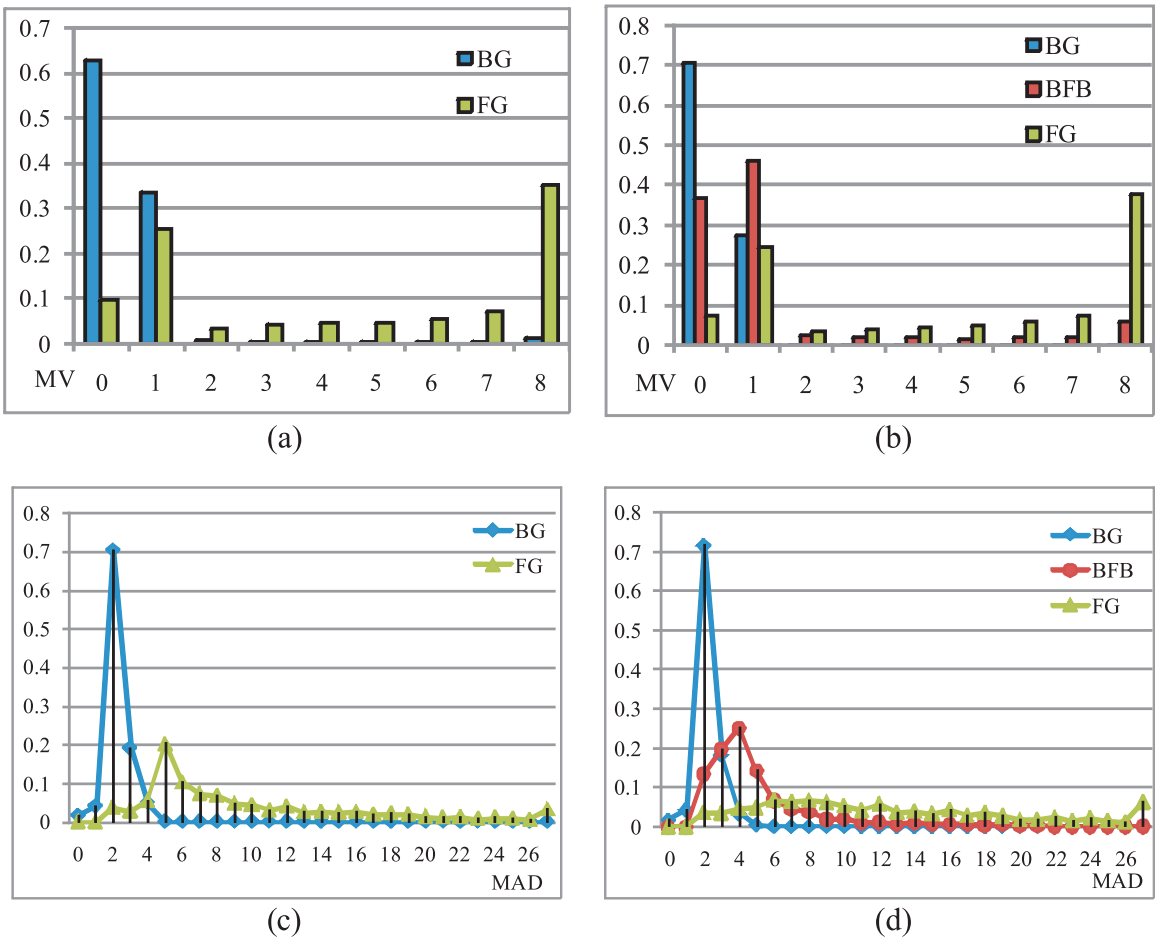

Figure 4.2 : Statistics of MV distribution: (a) traditional BG-FG approach, (b) Our BG-FG-boundary approach, statistics of average MAD distribution: (c) traditional BG-FG approach, (d) Our BG-FG-boundary approach.

It is evident from MV distribution that zero-MV biased search could be employed for BG regions to reduce the number of search points. On the other hand, the existing search methods could be used for FG regions resulting in optimum motion estimation.

4.1.2 Problems in the Existing Works

We investigated three major problems in the existing works. First, the blocks may be wrongly classified as BG or FG due to bi-level classification in traditional approaches [Zhao *et al.*, 2014]. The block classification could be correct for smaller block sizes like 4×4, whereas question lies for the block sizes larger than 8×8. There exists a considerable proportion of blocks on object boundary, which contains both BG and FG regions partially (called as a boundary (BD) blocks hereafter). The block matching process for these blocks is not straightforward. Performing the BG-based search for BD blocks could result in poor matching accuracy due to the presence of the FG region. On the other hand, performing FG-based search could still result in poor matching accuracy due to the presence of the BG region. Hence, there is a need to investigate newer search strategies for these boundary blocks.

It should also be noted that the video frames are divided into non-overlapping blocks in the existing works. In the non-overlapping scheme, some of the internal regions of the moving objects could

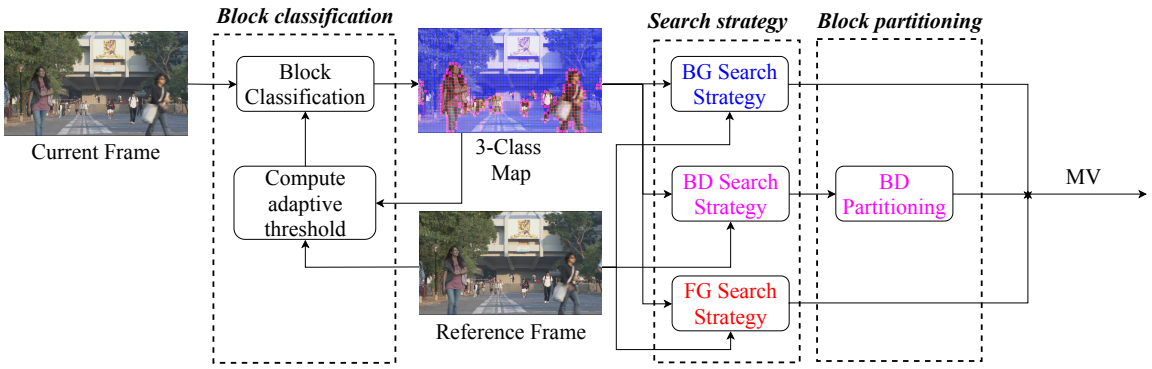

Figure 4.3 : Framework of the proposed background foreground boundary aware motion search scheme.

go undetected if the object has a size larger than the current block size, and if the change in pixel values is relatively small. Hence, a low distortion FG region may be wrongly classified as a static BG region. Thus, there is a need to employ overlapping block classification mechanism for better classification accuracy.

Second, the existing fast block matching algorithms suffer from a high probability of becoming trapped in the local distortion minimum [Lin *et al.*, 2009]. The primary reasons for this are (1) lower number of search points in the pattern, and (2) the non-sparsity in the location of search points in selected patterns. These problems make the DS [Zhu and Ma, 2000] algorithm ineffective. Although TZS [JCT-VC, 2013] and BFDS [Zhao *et al.*, 2014] overcome the challenge of being trapped in a local minimum, they still require a considerable number of search points. Conventional MV estimation methods use either zero $(0,0)$ location or median location as an initial search center for block matching process [Lin *et al.*, 2016a]. The median value is computed using MVs of adjacent blocks located at the left, top, and top-right position. It is observed that the median start location is more useful for FG blocks containing higher motion content. Although this approach speeds-up the matching process, the matching accuracy is severely compromised for blocks with object boundaries. Since these blocks may follow a different motion path than its neighboring blocks. The DS and TZS have widely adopted search algorithms in practice, and BFDS is a variation of the TZS algorithm designed for surveillance sequences. However, none of them could exploit the unique motion characteristics of FG blocks in the surveillance sequences. Hence, there is a need to innovate newer direction oriented search patterns for the faster and accurate matching process.

Third, the computational complexity involved in the search process is still high. The sub-sampling in the spatial domain substantially speeds up the computation of distortion measures. Since the number of pixels used for distortion computation is reduced [Yang *et al.*, 2010]. In literature, $1:2$ and $1:4$ sub-sampled distortion measure is widely adopted. However, to achieve better classification accuracy and improve block partitioning for boundary blocks present in the surveillance videos, a novel region-based sub-sampling is desired.

4.2 PROPOSED BACKGROUND FOREGROUND BOUNDARY AWARE SEARCH

A background foreground boundary aware motion search method is proposed to address the problems present in the existing works.

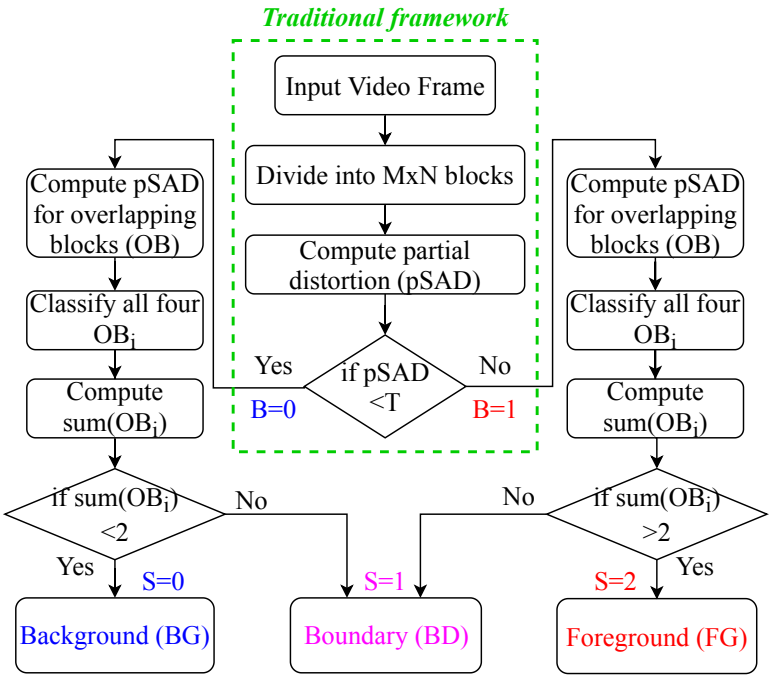

Figure 4.4 : Block classification framework.

4.2.1 Overview of the Proposed Background Foreground Boundary aware Search Scheme

The framework of the proposed method is shown in Figure 4.3. The proposed framework contains three parts: (1) three class block classification, (2) class-based search strategy, and (3) block partitioning mechanism. First, the current frame is divided into non-overlapping blocks of size $M \times N$. Then, each block is classified into one of the three classes: (1) background (BG), (2) foreground (FG), and (3) boundary (BD) based on the self-adaptive threshold. The foreground is the region with moving objects. On the other hand, the background is the static region in nature without any moving objects. In our work, a simple approach for the block classification, which is solely based block distortion measure is presented. The current frame, reference frame, and the generated class map is used for motion estimation. Second, different search strategies are employed for each class to exploit the inherent motion characteristics in the surveillance videos. Third, the block partitioning mechanism is incorporated for BD blocks for efficient matching. The details of each component of the proposed scheme are provided in the next sections.

4.2.2 Block Classification

The block classification framework for our work is shown in Figure 4.4. The current frame is first divided into non-overlapping blocks of size $M \times N$ and partial SAD (pSAD) is computed for each block. The partial SAD is a popularly used efficient measure of similarity between two frame-blocks: candidate block (C) and reference block (R) and it is computed as:

$$pSAD(x,y,m_x,m_y) = \sum_{i \in h_x} \sum_{j \in h_y} \left| \begin{matrix} C_{(x+i,y+j)} - \\ R_{(m_x+x+i,m_y+y+j)} \end{matrix} \right| \tag{4.1}$$

In (4.1), (x,y) represents position of the candidate block and (m_x,m_y) denotes relative displacement from the candidate block location. On the other hand, (h_x,h_y) denotes position of

highlighted pixels (in $M \times N$ sized block) used in partial SAD computation. The traditional SAD uses all the pixels present in the candidate block for distortion measure computation. On the other hand, partial SAD uses a subset of pixels, which are highlighted explicitly for computational advantage. Then based on the partial SAD value, the block is classified into two intermediate classes: (1) intermediate-BG, and (2) intermediate-FG, which can be expressed as:

$$B = \begin{cases} 0, & pSAD < T \\ 1, & otherwise. \end{cases} \tag{4.2}$$

where T is the SAD threshold used for classification. It is understood that the fixed thresholds will mostly provide judgment error in block classification. Hence, the block classification threshold needs to be adaptive in nature. To this end, [Liu and Jia, 2013] considers average distortion value for BG blocks belonging to the reference frame as an adaptive threshold. However, this mechanism fails to adjust the threshold in slight initial misclassification. To avoid this scenario, we propose to use a better adaptive threshold than used in [Liu and Jia, 2013].

$$T = \overline{SAD}_{pfbg} = \beta \times \frac{\sum_{(x,y) \in pfbg} SAD_{(x,y)}}{n_{pfbg}} \tag{4.3}$$

where β is multiplying factor, $pfbg$ is the background region in the previous frame and n_{pfbg} is the total number of BG blocks in the previous frame. The detailed discussion on selection of β is carried out in the experimental settings section.

The non-overlapping scheme used in traditional block classification framework has the disadvantage that some of the internal regions of the moving objects could go undetected if the object has a size larger than the current block size, and if the change in pixel values is relatively small. To address this, we propose to use four overlapping blocks (OB) centered at the center of the current block. The illustration of the current block and corresponding overlapping blocks is shown in Figure 4.8 (b). The overlapping blocks are also classified based on the threshold of T.

$$OB_i = \begin{cases} 0, & pSAD < T \\ 1, & otherwise. \end{cases} \tag{4.4}$$

The different overlapped-block classification scenarios are shown in Figure 4.5. The block is classified into three different classes, namely: (1) BG block, (2) FG block, and (3) BD block based on the number of overlapped blocks belong to the respective intermediate class. The current block has a total of four possible overlapped blocks, as shown in Figure 4.5. Let us consider if the current block is classified as BG block, and all the four overlapped blocks are also classified as BG block then the current block is highly likely to be BG block and vice-versa. With this understanding, a current block shown in Figure 4.5 (a) and (f) can be classified as BG and FG blocks, respectively. However, it is not necessary that all the overlapping blocks be classified as the same class. In a situation where the current block is located at the object boundary, some of the overlapped blocks may be classified as BG blocks, while remaining overlapped blocks are classified as FG blocks. In this situation, we can consider the number of overlapped blocks matching the class of the current block to update the current block classification. To address this issue, we propose to classify the current block as boundary block if and only if at least 50% of the overlapped blocks belong to the different class than the class of the current block. The class of the current block is unchanged for situations illustrated in Figure 4.5 (b) and (g). On the other hand, the current block is classified as a BD block for situations illustrated in Figure 4.5 (c, d, e) and (h, i, j).

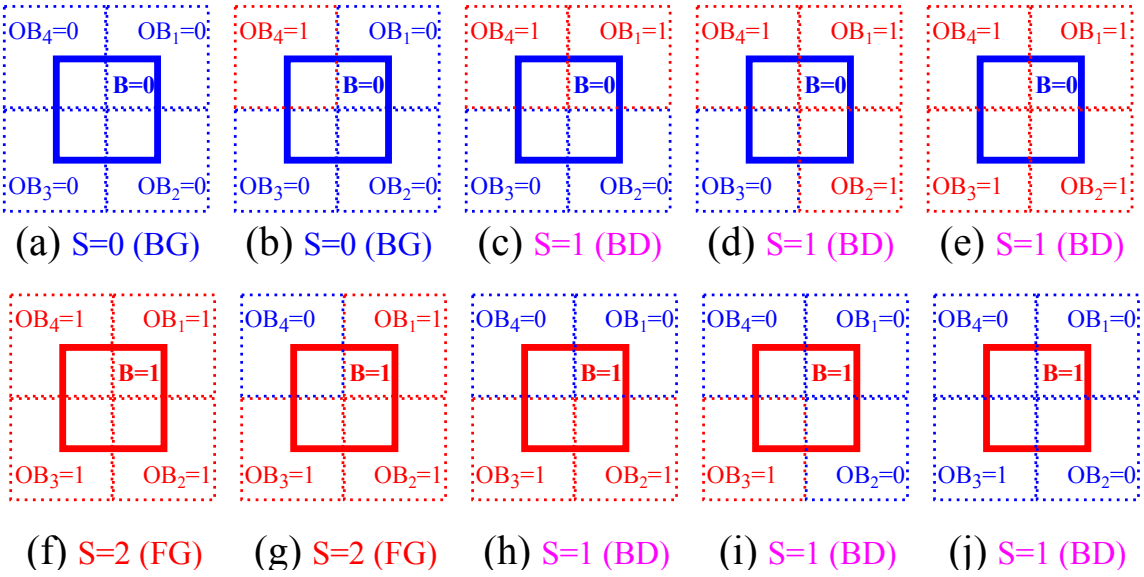

Figure 4.5 : Different block overlap scenarios for block classification.

$$S = \begin{cases} 0, & \sum_{i=1}^{4} OB_i < 2 \; and \; B = 0 \\ 2, & \sum_{i=1}^{4} OB_i > 2 \; and \; B = 1 \\ 1, & otherwise. \end{cases} \tag{4.5}$$

This classification, with the inclusion of overlapping blocks, has not only resulted in better three-level saliency detection but also reduced the chances of block misclassification. The addition of boundary class, along with traditional background and foreground classes, provides multiple benefits. The advantages are listed as follows: (1) improved motion vector prediction since boundary blocks cannot be trusted for predicting possible motion direction, (2) improved matching accuracy due to multiple classes, and (3) fast motion search convergence for each category.

4.2.3 Block-class-based Search Strategy

Different search strategies are employed for each class in the surveillance video coding framework based on their MV distribution characteristics. The MV distribution corresponding to the FS algorithm is used for better understanding and analysis.

4.2.3.1 Zero-biased Search for BG Blocks

The MV distribution is shown in Figure 4.2 (b) that clearly indicates that the BG blocks are static in nature and seldom move. Hence, no block matching is performed for these blocks, and the MV is set to zero. It should be noted that the use of better search patterns would only increase computational requirements without any significant improvement in matching accuracy for these blocks. The higher chances of MV process to be trapped in local optimum makes fast search patterns ineffective for BG blocks. The fixed zero-MV strategy for BG blocks has achieved better trade-off compared to other fast block matching methods. The BFDS [Zhao *et al.*, 2014] and TZS [JCT-VC, 2013] needs 1 and at least 24 points, respectively. It should be noted that although our method is searching one zero-MV point, our computational complexity is 75% better than the traditional zero-MV approach due to the utilization of

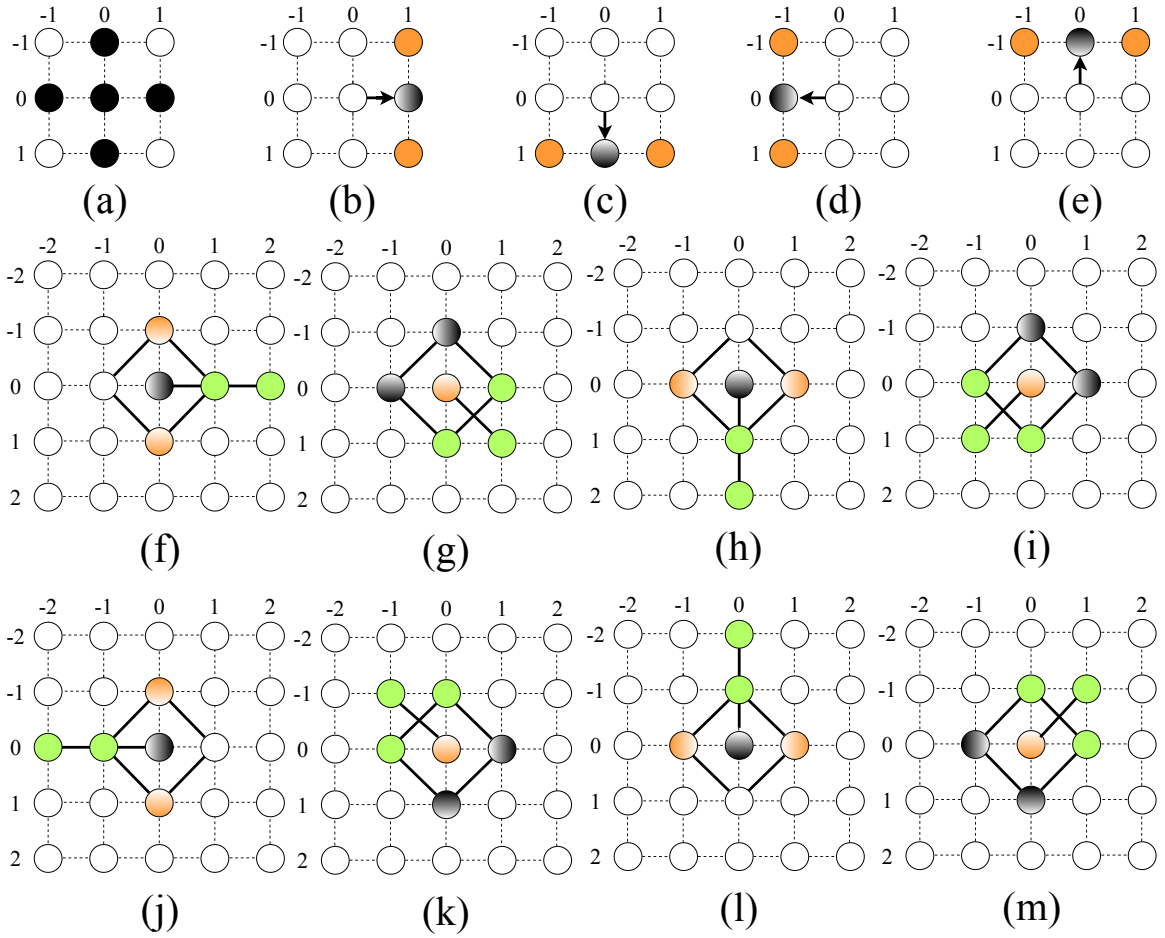

Figure 4.6 : Search patterns: (a) 1^{st} step pattern: SDSP, (b) 2^{nd} step east directional pattern, (c) 2^{nd} step south directional pattern, (d) 2^{nd} step west directional pattern, (e) 2^{nd} step north directional pattern, (f-m) 3^{rd} step onward search patterns: (f) 0° directional pattern: RUWDS, (g) -45°, (h) -90°, (i) -135°, (j) 180°, (k) 135°, (l) 90°, and (m) 45°.

$1:4$ sub-sampling during the block classification process. The proposed sub-sampling structure is shown in Figure 4.8 (a). Detailed discussion on the sub-sampling structure is carried in the section related to the speed-up mechanism.

4.2.3.2 Direction-Oriented Search for FG Blocks

The MV distribution is shown in Figure 4.2 (b) for a fast motion FG region that indicates that about 30% of MVs lie within a unit area from zero-MV, but the rest of the MVs are present in all the directions. On the other hand, for the directional motion content FG region, the MVs are mostly located in a particular direction. About 70% of MVs for these content lie in a particular direction. More appropriate points are included in traditional search patterns to avoid the problem of being trapped in a local minimum. Further, to exploit the directional motion characteristics of FG blocks, this work introduces novel directional search patterns shown in Figure 4.6. For efficient directional motion tracking, the novel search patterns are created by adding uni-wing to the small diamond search pattern (SDSP) [Zhu and Ma, 2000] in respective rotating directions. These patterns are termed as rotating uni-wing diamond search pattern (RUWDS). Eight possible combinations of uni-wing patterns: $0^{\circ}, \pm 45^{\circ}, \pm 90^{\circ}, \pm 135^{\circ}, 180^{\circ}$ are shown in Figure 4.6 (f-m). These patterns can track horizontal, vertical

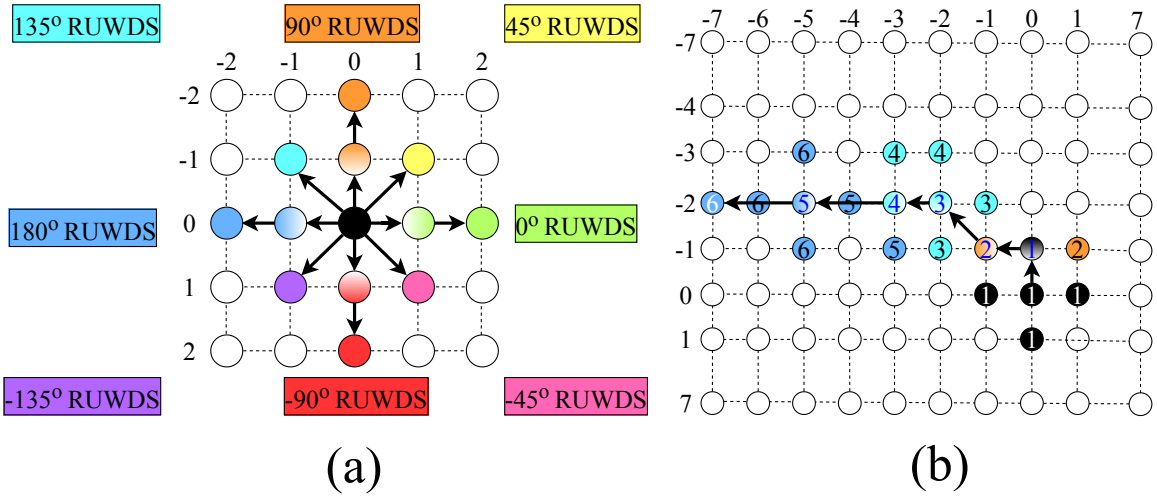

Figure 4.7 : (a) Illustration of coordinate location $(\Delta x_i, \Delta y_i)$. One of the eight direction-oriented search patterns is chosen based on $(\Delta x_i, \Delta y_i)$, (b) Typical search path example. The corresponding step number is marked in all the traversed search points.

as well as diagonal movements very effectively. These patterns can provide faster convergence due to the inclusion of extra points in the direction of motion.

The MV obtained during intermediate steps of search process is referred as intermediate-MV (IMV), whereas the final MV obtained after completion of the search process is referred as final-MV (FMV). The search process is described in following steps:

Step 1: The search process starts from search center: either default zero-MV or median-MV and uses the SDSP shown in Figure 4.6 (a). The block distortion values for the five points of SDSP (highlighted in black color) are obtained and minimum distortion location is treated as IMV of the first step. The search process is terminated if the minimum distortion location lies at the search center.

Step 2: One of the three-point search patterns shown in Figure 4.6 (b-e) is chosen such that IMV obtained during the first step, is the new search center. The minimum distortion location is treated as IMV of the second step.

Step 3: The new search pattern selection for the i^{th} step is based on the coordinate difference of the IMVs obtained in previous two search steps: $(i-1)^{th}$ IMV and $(i-2)^{th}$ IMV. 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:

$$(\Delta x_i, \Delta y_i) = (x_{i-1} - x_{i-2}, y_{i-1} - y_{i-2}) \tag{4.6}$$

The search pattern for the subsequent steps are chosen based on the current direction of motion identified by $(\Delta x_i, \Delta y_i)$. The illustration of coordinate $(\Delta x_i, \Delta y_i)$ mapping is shown in Figure 4.7 (a). This mapping helps in understanding the pattern to be used during next search step. The search process is terminated if $(\Delta x_i, \Delta y_i) = (0,0)$. Otherwise, the process is continued with Step 3 till the pattern touches search region boundary.

The path traversed by each search step for the block matching is referred to as the search path. The typical search path example is shown in Figure 4.7 (b). The search process uses SDSP in 1^{st} step,

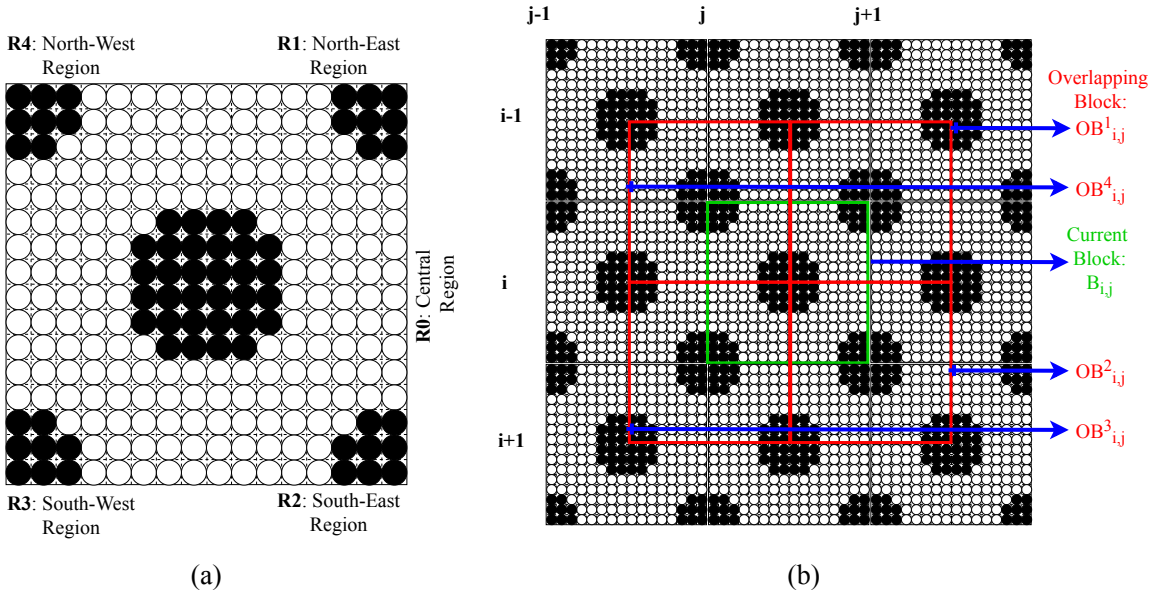

Figure 4.8 : (a) 16×16 block (pixel grid) with region-based $1:4$ sub-sampling. Highlighted pixels of five different regions are used for partial distortion computation, (b) Illustration of 16×16 current block (highlighted in green) and corresponding four overlap blocks (highlighted in red).

two vertex point north-directional search pattern in 2^{nd} step, $135°$ RUWDS pattern in 3^{rd} and 4^{th} steps, followed by $180°$ RUWDS pattern in the last two steps, before the search is terminated since the pattern touched the search region boundary. It should be noted that our patterns can jump two-pixel locations at once in horizontal and vertical directions. Hence, these patterns converge faster without increasing the number of points searched.

4.2.3.3 BD Block Partitioning and Center-biased Direction-Oriented Search for BD Blocks

The MV distribution is shown in Figure 4.2 (b) for boundary blocks that indicate that more than 70% of MVs lie within a unit area from zero-MV, but the rest of the MVs are present in all the directions. Although BD blocks have static nature similar to BG blocks, they also tend to have MVs located at all the directions. This is mainly because of the fact that these blocks contain both BG and FG regions simultaneously. This means that the partial region belonging to BG would have zero-MV, and the remaining region would have object motion based larger MV. This has resulted in matching ambiguity in the motion search process. The zero-biased search strategy could result in more mediocre matching accuracy, whereas a direction-oriented search strategy could also provide sub-optimal performance due to the presence of the BG region.

To address this problem, we partitioned the BD blocks into five non-overlapping sub-regions, as shown in Figure 4.8 (a). The sub-regions are intentionally spaced apart for a better understanding of BG and FG portions. The BD partitioning process is illustrated in Figure 4.9. The sub-regions: R0, R1, R2, R3, and R4 are classified into two categories based on the distortion values. The MAD distribution is shown in Figure 4.2 (d), that indicates that the MADs for BG is quite lower and concentrated near zero, whereas MADs for FG is uniformly distributed and large in general. Based on this understanding, the MADs for the five sub-regions are computed, and the lowest distortion sub-region is assumed to belong to the BG area. In contrast, the highest distortion sub-region is assumed to belong to the FG area. The remaining

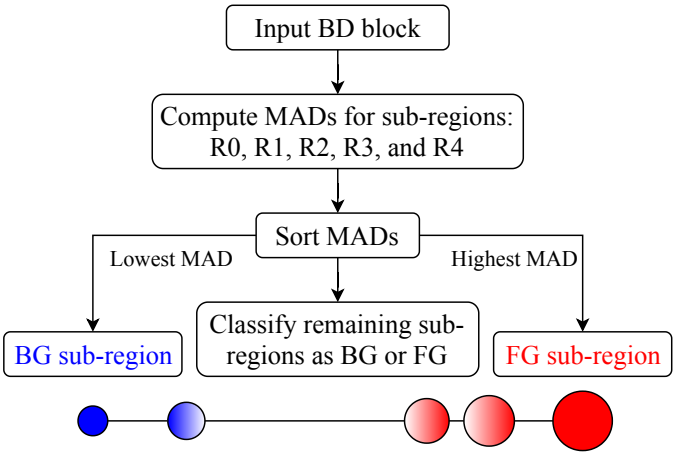

Figure 4.9 : BD block partitioning process. The size of circle is proportional to the MAD values. BG and FG sub-regions are highlighted in blue and red color respectively.

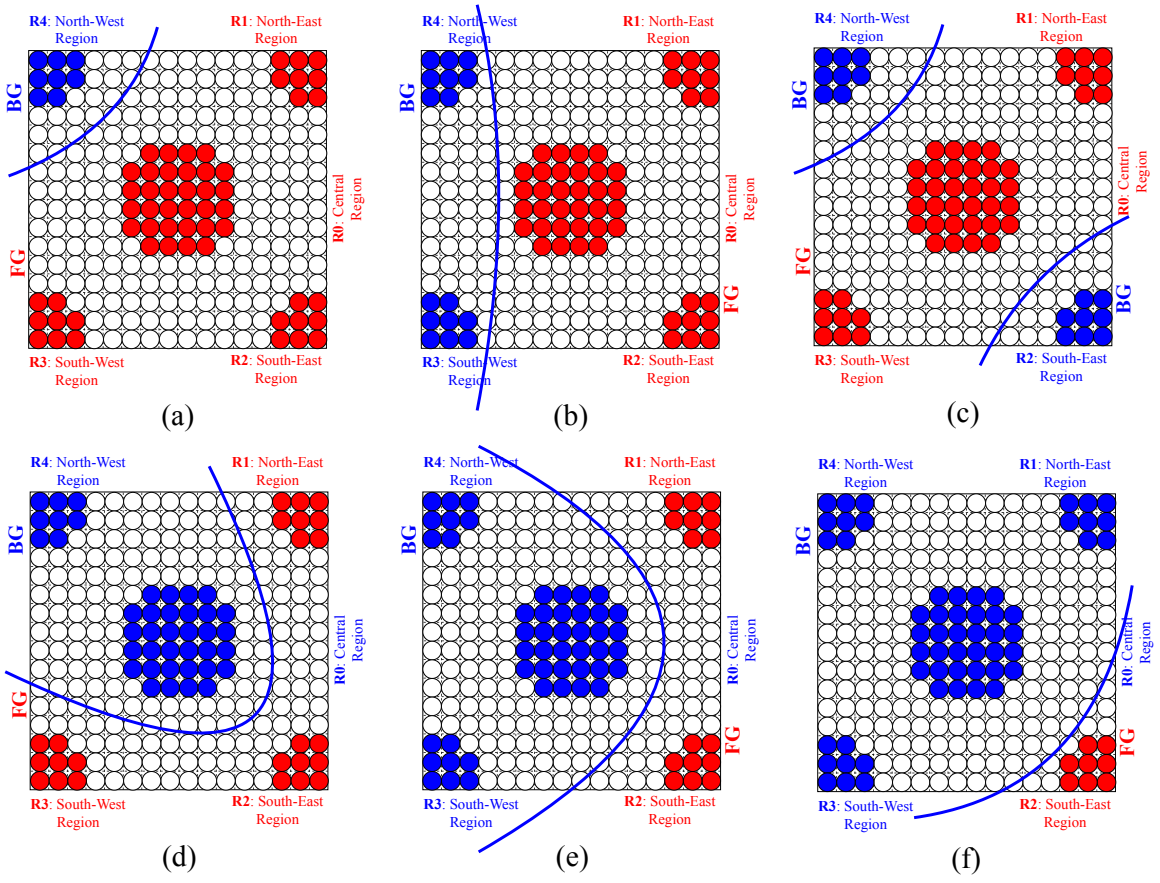

Figure 4.10 : Different BD block partitioning scenarios: (a) single region R4, (b) two adjacent regions R4 and R3, (c) opposite regions R4 and R2, (d) R4 and central region R0, (e) R4, R3 and R0, (f) R4, R1, R3 and R0. BG and FG regions are highlighted in blue and red color respectively.

three sub-regions are classified based on the linear threshold. Different BD partitioned scenarios are illustrated in Figure 4.10. The five sub-regions could result in a total of 32 different partitioned scenarios. However, our assumption of at least one sub-region belongs to FG and BG each, results in the following two cases to be non-existent: (1) all sub-regions are BG, and (2) all sub-regions are FG.

Once the sub-regions are classified, we propose to use only FG sub-regions for the block matching process. This mechanism not only improves the matching accuracy but also increases the search speed due to the exclusion of BG regions from the current BD block. Although FG sub-regions in the BD block are considered for block matching process, it should be noted that remaining BG sub-regions in the BD block are considered to be static in nature, and more specifically, they are treated as zero-MV sub-regions. Hence, region segmentation for the BD block provided not only better matching accuracy for FG sub-regions but also for BG sub-regions. In a nutshell, if the BG region is the main part of the BD block, the matching accuracy will not be compromised. The matching accuracy could only be further improved as compared to the traditional block matching approaches.

It should also be noted that traditionally considered median-MV as an initial search center approach would be counter-productive for these blocks since the motion characteristics at object boundaries are quite different and challenging to predict. Hence we propose to use a zero-biased direction oriented search for BD blocks. The search patterns proposed for FG blocks are utilized on partitioned BD sub-regions for better matching accuracy.

4.2.4 Block Partitioning Mechanism for Speed-up

The sub-sampling in the spatial domain for distortion measure computation provides additional speed-up. Since the number of pixels used for distortion computation is reduced [Yang *et al.*, 2010]. In literature, $1:2$ and $1:4$ sub-sampled distortion measure is widely adopted. It resulted in 50% and 75% savings in the number of pixels used for each distortion measure computation, respectively. However, to achieve better classification accuracy and improve block partitioning for boundary blocks present in the surveillance videos, region-based sub-sampling is desired. To address this, we present region-based $1:4$ sub-sampling. The illustration of 16×16 block (pixel grid) with $1:4$ sub-sampling is shown in Figure 4.8 (a). The sub-sampled grid contains a total of five sub-regions. One central sub-region referred to as R0 and four corner sub-regions, namely north-east region, south-east region, south-west region, and north-west region referred to as R1, R2, R3, and R4, respectively. The advantages of our region-based partial distortion measure as three-fold: (1) it provides desired computational saving without compromise in matching accuracy, (2) it would be used to improve block classification accuracy further, and (3) the region-based sub-sampling is symmetric, and hence it is easily scalable for different block sizes.

4.3 EXPERIMENTS AND ANALYSIS

4.3.1 Datasets

Experiments were carried on different surveillance test sequences listed in Table 4.1. Twelve test sequences with varying foreground motion content, different background proportion, and of different spatial frame sizes are chosen from benchmark databases [Gao *et al.*, 2014; Lab, 2019; Seshadrinathan *et al.*, 2010b; Lab, 2011]. Among them, eight sequences are taken from PKU-SVD-A dataset [Gao *et al.*, 2014], two sequences are taken from standard CIF sequences [Lab, 2019], and a sequence is taken from each LIVE dataset [Seshadrinathan *et al.*, 2010b] and IVPL dataset [Lab, 2011]. The background is mostly static throughout the length of the test sequence. The sequences are categorized into four different types: {A, B, C, and D}, based on motion content and proportion of motion area. The description of each sequence is as follows: Campus- Students and car moving in the campus, Classover- Students walking and cycling after class is over, Hall monitor- Person moving and picking briefcase, Ice- Crowd scene: people doing ice skating, Office- Office working scene with a person entering, Square- Students

Table 4.1 : Surveillance test sequences used in the study.

Sr. No.	Sequence name	Spatial resolution	Total frames	Frame rate	Frame starts	Dataset	Motion type	Motion area	Sequence type
1	Campus	720×576	3001	30	2701	PKU	Slow	Small	A
2	Classover	720×576	3001	30	351	PKU			
3	Hall monitor	352×288	300	30	21	NCTU			
4	Ice	352×288	240	30	21	NCTU	Slow	Large	B
5	Office	720×576	3001	30	1351	PKU			
6	Square	1920×1088	250	25	11	IVPL			
7	Bank	720×576	3001	30	201	PKU	Fast	Small	C
8	Crossroad	720×576	3001	30	1501	PKU			
9	Intersection	1600×1200	1001	30	101	PKU	Fast	Large	D
10	Mainroad	1600×1200	1001	30	501	PKU			
11	Overbridge	720×576	3001	30	1001	PKU			
12	Pedestrian area	768×432	250	25	81	LIVE			

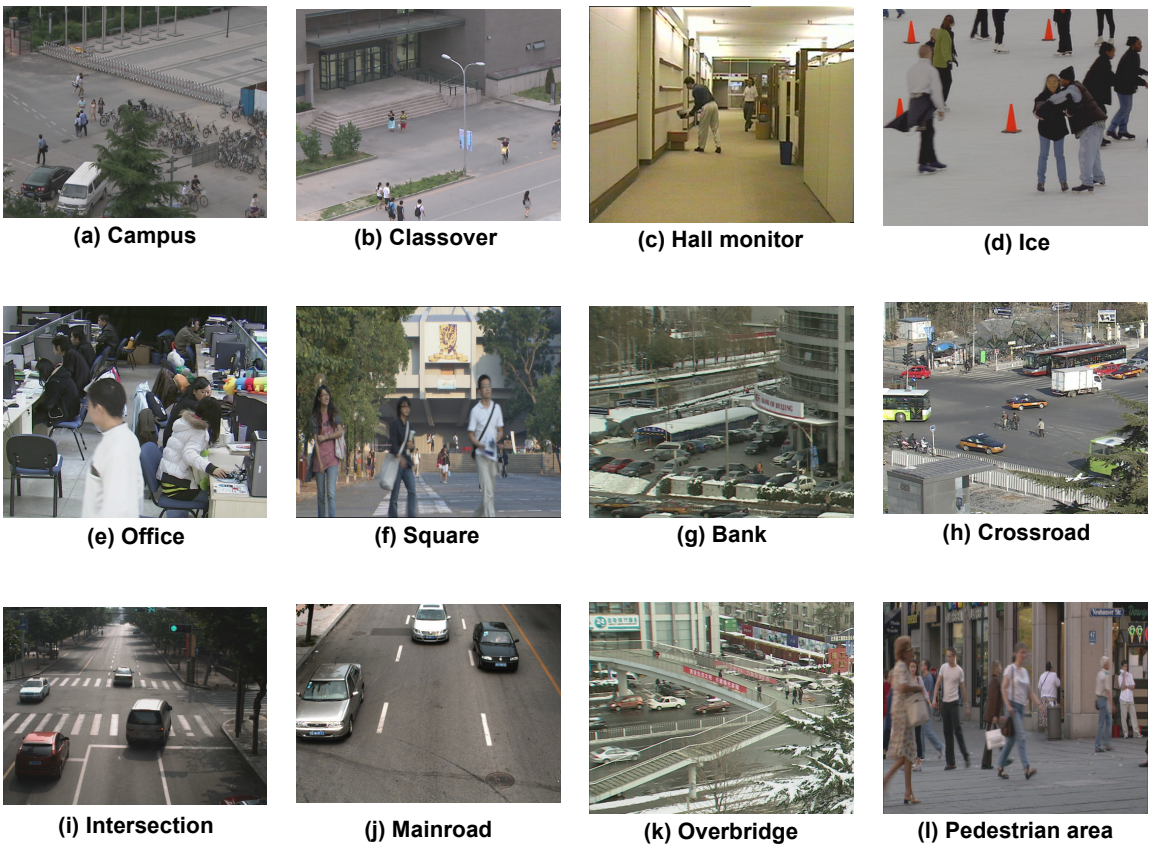

Figure 4.11 : Representative frames of surveillance sequences used in the analysis.

moving in front of the building, Bank- Vehicles moving in the perpendicular direction, Crossroad- Buses moving at a crossroad, Intersection- Vehicles moving in speed at the intersection, Mainroad- Fast moving cars seen from closer top-view, Overbridge- Moving vehicles and people walking over the bridge, and Pedestrian area- Passerby encountering on an open square. All test sequences are in YUV 4:2:0 (uncompressed) color format, and only the luminance (Y) component of each test sequence with the

8-bit resolution is used for block matching experiments. Total 100 frames of each sequence starting from the frame start mentioned in Table 4.1 are considered for experiments.

4.3.2 Evaluation Metrics

The peak signal-to-noise ratio (PSNR) is a widely adopted objective quality metric to evaluate the performance of block matching methods. Although bit-rate provides a better understanding of compression efficiency, the standard block matching algorithms only focus on matching accuracy. Better matching accuracy could lead to better compression performance. Moreover, our analysis is focused only on the motion estimation component of the surveillance coding engine. Hence, in this chapter, bit-rate, PSNR, and search complexity are considered during the experimental evaluation.

Prediction quality: The traditional PSNR inherently provides equal weight to each pixel during mean-squared-error (MSE) computation. It lacks to provide different importance to different areas in the sequence based on their relevance. However, for surveillance sequences, it is highly desired that the foreground area is given more weight based on its higher relevance over background areas. Evaluation metric Weighted PSNR (WPSNR) based on Weighted MSE (WMSE) is used in [Zhang *et al.*, 2012]. WPSNR provided a better understanding of the surveillance sequence coding performance. Hence, WPSNR is considered for objective performance evaluation.

$$WPSNR = 10\log_{10}\left(\frac{V_{\max}^2}{WMSE}\right) \tag{4.7}$$

$$WMSE = \sum_{s=0}^{S-1} \frac{\omega_s}{|C_s|} \sum_{(m,n)\in C_s} MSE(m,n) \tag{4.8}$$

where V_{max} is the maximum pixel intensity for the given bit resolution, S is the number of classes, ω_s is the weight of class s, C_s is the set of blocks belonging to the class s, $|C_s|$ is the number blocks belonging to the class s, and $MSE(m,n)$ is the mean-squared-error of $(m,n)^{th}$ block.

Search complexity: The search complexity in the motion search algorithm not only depends on the total number of search points traversed in the search path but also on the number of computations performed at each search point. Hence the parameter is proposed to include both: (1) average number of search points and (2) proportion of pixels used for partial distortion measure. To this end, the average number of checked pixels per candidate block (ANCPB) is computed as:

$$ANCPB_\alpha = \alpha \times \text{Average number of search points} \tag{4.9}$$

where α represents the proportion of pixels used in the sub-sampled SAD computation.

4.3.3 Experimental Settings

Search parameters: Our framework is general and scalable. Hence, it can be easily extended to different block sizes. However, we consider widely adopted search parameters in our experiments. The block size as 16×16 and search range as $p = \pm 8$ is considered.

The effect of variation in thresholds for block classification is shown in Figure 4.12. As expected, fixed-threshold value resulted in judgment error. An increase in the fixed-threshold value from 256 to 2048 resulted in an increase in the proportion of blocks classified as BG. The adaptive-threshold selected with $\beta = 1$ resulted in sudden decay in the proportion of BG blocks. For addressing this, the $\beta > 1$ is considered in our experiments. We have empirically found that multiplying factor $\beta = 2$ in (4.3) has two

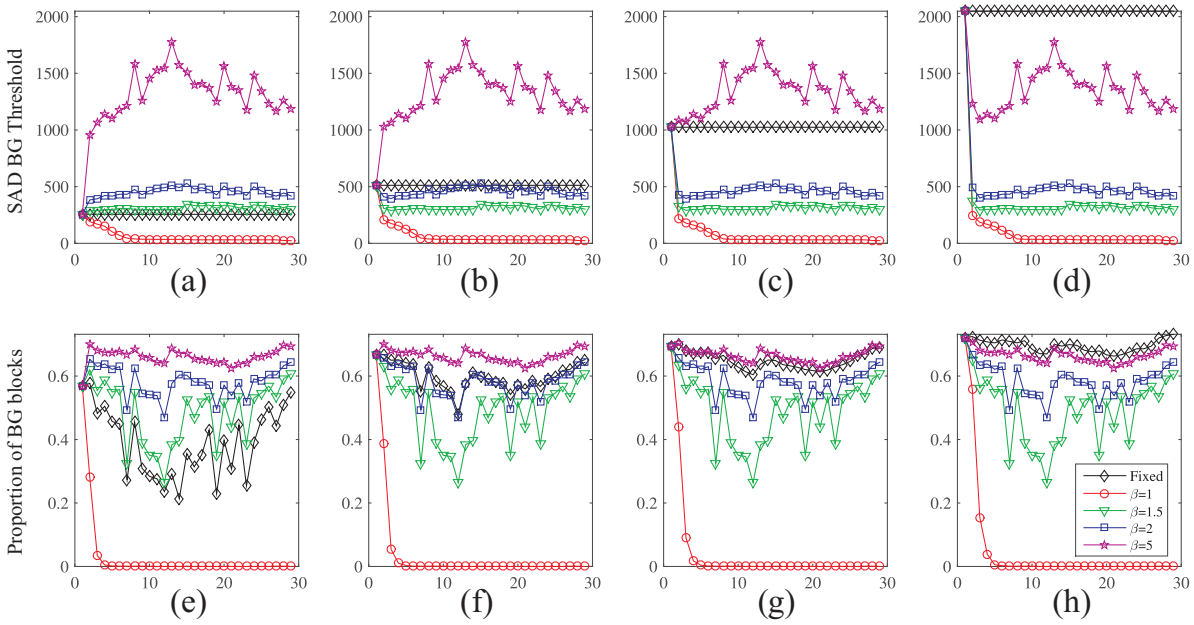

Figure 4.12 : The effect of variation in fixed and adaptive thresholds with different multiplying factor β for block classification. Column 1-4 considers the initial-threshold value as 256, 512, 1024, 2048, respectively.

advantages: (1) the adaptive-threshold value is independent of initial-threshold value, it is self-adaptive in nature and varies smoothly, and (2) as desired, the proportion of BG blocks does not change drastically for consecutive frames.

For BD block partitioning, different sub-sampling structures were explored. The criteria for the selection of a particular structure lies in three main reasons. The reasons are listed as follows: (1) the sub-regions must have considerably enough pixels in it, (2) the pixels must be close-enough or intact in particular sub-region, and (3) different sub-regions must be spaced apart. It is found that sub-region with a circle radius $r = 3$ provided the best performance. The structure is shown in 4.8 (a) that closely resembles the criteria mentioned above and hence chosen in our method.

Test setup for rate-distortion comparison: The test sequences listed in Table 4.1, 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.

4.3.4 Results for Different Components of the Proposed Method

Several experiments were carried out to validate the efficacy of the proposed method. We explore the effect of different components of the proposed method and demonstrate the performance improvement of the proposed algorithm over other existing methods. In this section, we compare the effect of various components of the proposed approach. The proposed framework contains three components: (1) 3-class block classification (BC), (2) class-based search strategy (SS), and (3) block partitioning (BP) mechanism for speed-up. To study the effect of an individual component, we keep the remaining two components in the proposed framework unchanged. The experimental results for different combinations of three components are shown in Table 4.2.

First, we evaluate the effect of our 3-class block classification framework against the traditional 2-class block classification approach. To this end, we only change component BC and keep components SS and BP unchanged. Our 3-class approach in BC+SS+BP yielded, on average, better PSNR results than the traditionally used 2-class method in SS+BP. The inclusion of additional boundary class to the existing BG and FG classes provided a better classification for surveillance videos. The improved motion search strategy for boundary class provided better matching accuracy. It is evident from Table 4.2, that improvement up to 0.25 dB in PSNR is achieved mainly because of the better classification accuracy. However, the motion search complexity for the 3-class approach is about 15% higher than the 2-class method. The subjective evaluation for the traditional 2-class approach and our 3-class approach are shown in Figure 4.13. It is evident from the results that the proposed approach can extract important characteristics in the surveillance sequences. The blocks at the object boundaries are either classified as BG or FG in the traditional approach. But these blocks contain both BG and FG partially, and hence our approach with additional boundary class addresses this problem. In turn, the block misclassification is significantly reduced.

Second, we evaluate the effect of our direction-oriented search patterns against traditional TZS patterns. To this end, we only change component SS and keep components BC and BP unchanged in our framework. For fair comparisons, the motion estimation process for background blocks in both BC+SS+BP and BC+BP is unchanged. The direction oriented search patterns with an adaptive search center is utilized for FG and boundary blocks in BC+SS+BP, whereas TZS patterns are being used in BC+BP. Our block classification based adaptive search strategy clearly outperformed the traditional search strategy. The improvement up to 3.19 dB in PSNR for Mainroad sequence and gain of about 0.28 dB in PSNR on an average is achieved mainly because of our direction oriented search patterns and adaptive search center selection. Not only this, but the motion search complexity for our search pattern is also about five times better than the traditional search pattern. Hence, we have been able to achieve both improvements in matching accuracy and a significant reduction in search complexity at the same time.

Third, we evaluate the effect of our speed-up approach against the proposed method without BP. To this end, we only change component BP and keep components BC and SS unchanged in our framework. Our partial distortion mechanism in BC+SS+BP yielded about 0.02 dB improvement in PSNR on an average over BC+SS. Moreover, our partial distortion approach also provided about 10% improvement in search complexity over BC+SS on average, mainly because of the utilization of a lower number of pixels for each distortion computation.

Overall, the proposed approach with all three components BC+SS+BP provided the best PSNR results on an average mainly due to the inclusion of direction oriented search patterns. On the other hand, SS+BP provided the least search complexity due to a 2-class classification approach and partial distortion measure. The trade-off for matching accuracy and motion search complexity is obtained by BC+SS+BP combination. Next, we compare the effect of proposed BC+SS+BP against other existing methods.

4.3.5 Comparative analysis for the Proposed and the Existing Methods

The performance of the proposed method is compared against five traditional block matching algorithms: FS, DS [Zhu and Ma, 2000], TZS [JCT-VC, 2013], modified hexagon grid search (MHGS) [Singh and Ahamed, 2018], and fast adaptive motion estimation (FAME) [Mukherjee *et al.*, 2018]. The performance is also compared against a surveillance-based block matching method BFDS [Zhao *et al.*, 2014], which uses the TZS pattern for FG blocks. There are very few block matching methods designed specifically for surveillance sequences. For better performance evaluation, we create two new methods by incorporating the latest MHGS and FAME methods in the place of TZS in the BFDS framework. The block classification in these two methods is kept the same as BFDS. Hence, in total, we compare the proposed method against five traditional and three surveillance-based block matching algorithms.

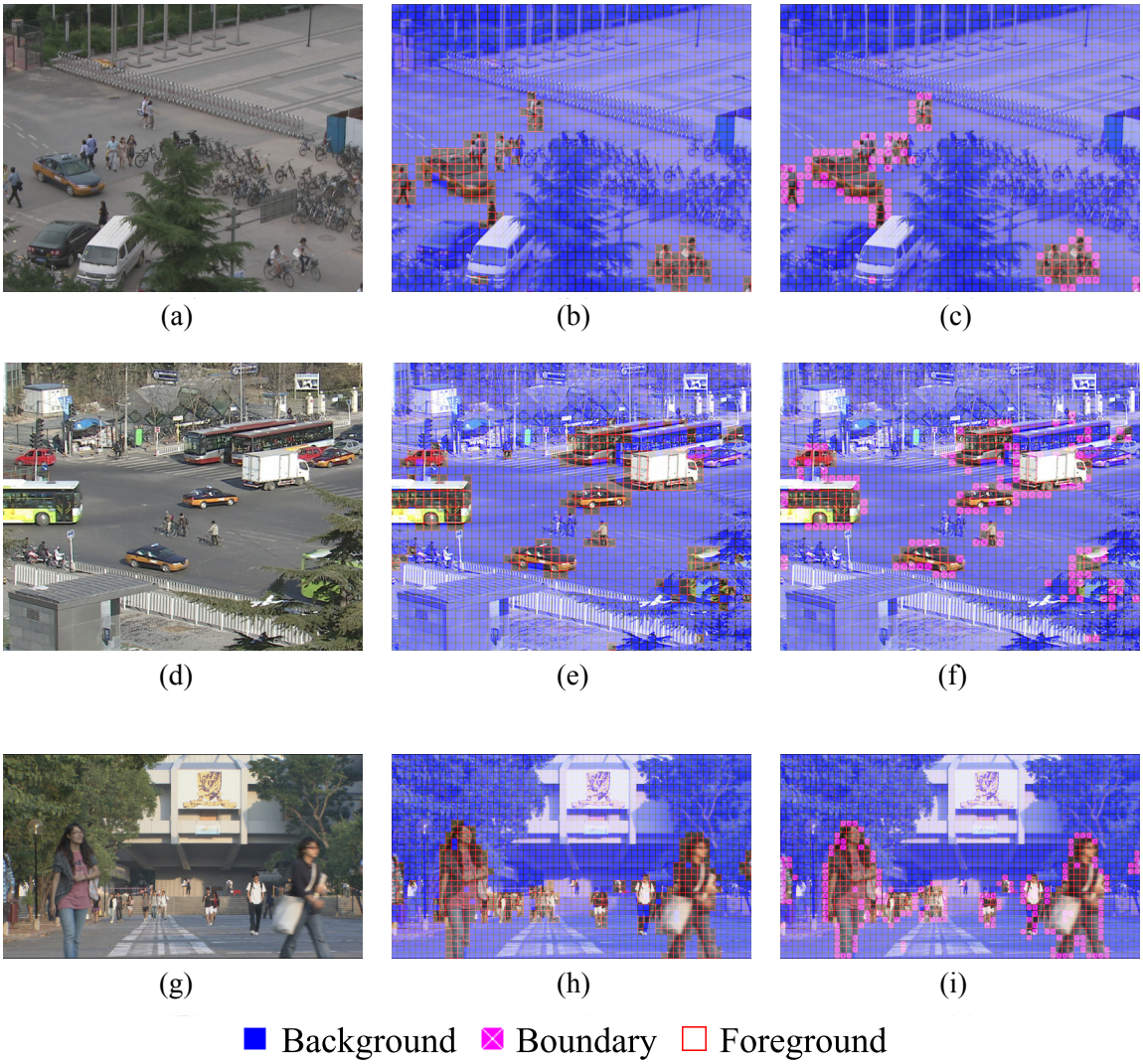

Figure 4.13 : Results of the proposed block classification method. (a) Original frame, Campus (46^{th} frame), (d) Original frame, Crossroad (6^{th} frame), (g) Original frame, Square (50^{th} frame), (b), (e), and (h) 2-Class classification, (c), (f), and (i) Our 3-Class classification.

The PSNR and motion search complexity results for different block matching algorithms are shown in Table 4.3. As expected, the block matching accuracy is the best for FS algorithm among existing algorithms due to its exhaustive search nature. The PSNR results indicate that the DS performs worst among compared algorithms. This is because DS has a typical structure that gets trapped in a local optimum. However, TZS and BFDS performed better than the DS due to the inherent capability of the TZS pattern and its variants. MHGS achieves more than 18% computational saving than TZS but comes with slightly poorer matching accuracy. The FAME algorithm outperforms DS in both matching accuracies and the search complexity due to an advanced search mechanism. The PSNR and WPSNR values for the proposed algorithm clearly outperformed existing methods on an average. It should be noted that our algorithm not only outperformed all fast block matching methods but also the FS algorithm. For the Mainroad sequence, the proposed algorithm achieved more than 3.5 dB improvement in WPSNR over FS.

The block-wise PSNR comparisons shown in Figure 4.15 indicates that the proposed algorithm outperforms existing methods for BD and FG blocks. Our approach provided better results for BD blocks due to the effective utilization block partitioning mechanism. Hence, the proposed method is observed to be superior for these foreground motion contents. It is achieved mainly because of the four reasons mentioned as follows: (1) the search starts from either zero-MV or median-MV, (2) the new search window with new search-center is chosen, (3) only FG portion in BD blocks in chosen for block matching, and (4) due to the capability of our proposed direction oriented search patterns to capture directional motion search characteristics of the moving objects.

FS is the computationally demanding method and requires $(2p+1)^2$ search points for the single MV search operation. On the other hand, the DS algorithm requires $9 + 5 \times (nSteps) + 4$ number of search points. The ANCPB results shown in Table 4.3 indicates that the computational complexity is in the order FS $>$ TZS $>$ MHGS $>$ DS $>$ BFDS $>$ BFDS+MHGS $>$ FAME $>$ Proposed $>$ BFDS+FAME. Although TZS and MHGS provided better PSNR results, it still comes at a high computational expense. The BFDS algorithm used only a single search point for BG blocks as expected and cleverly outperformed TZS, which uses at least 24 search points without any notable degradation in matching performance.

On the other hand, our method used a sub-sampling pattern for BG blocks and achieved a further improvement of 75% in computations for those blocks. Not only this, but the block-wise ANCPB comparisons shown in Figure 4.16 indicates that our algorithm significantly outperformed existing methods (except BFDS+FAME) for BD and FG blocks as well. Our method is four times computationally better than the next-best method for BG blocks. On the other hand, the proposed method is at least two times computationally better for BD and FG blocks. This is achieved mainly because of the three reasons: (1) the search could directly start from the highly probable vicinity of true MV, (2) the new direction-oriented search patterns are chosen based on the motion characteristics, and (3) region-based partial distortion measure.

Figure 4.14 presents the RD performance of the proposed and competitive algorithms for video sequences: Ice, Square, Bank, and Pedestrian area. The RD performance is computed under different QP values of 22, 27, 32, and 37. It is evident from Figure 4.14 that, increase in PSNR is proportional to an increase in bit-rate. For the Bank video sequence, the FS algorithm provided the best RD performance. All the remaining algorithms also performed at par with FS, in this case, considering the small motion area in the Bank video sequence. On the other hand, the RD performance for remaining video sequences indicates the superiority of the proposed algorithm over existing algorithms. One may observe that our algorithm outperformed not only to state-of-the-art TZS and BFDS algorithms but also the FS algorithm. The gains for large motion area sequences are higher than that of small motion area sequences due to the adaptability of our direction oriented search patterns.

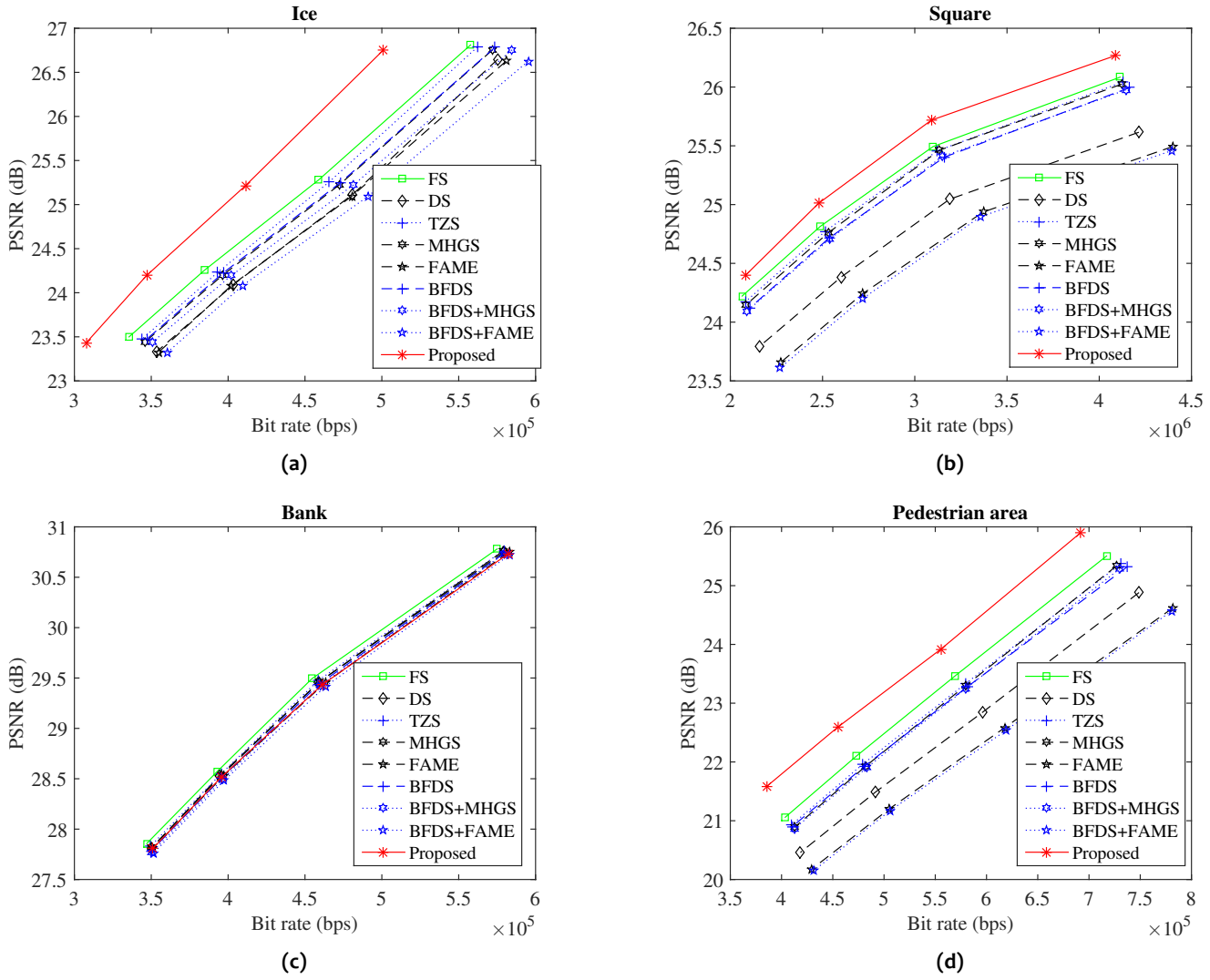

Figure 4.14 : Rate-distortion performance for video sequences (a) Ice, (b) Square, (c) Bank, and (d) Pedestrian area under different QP values of 22, 27, 32, and 37.

Table 4.2 : PSNR and motion search complexity (ANCPB) comparisons for different components of proposed method. The full version of proposed method contains all three components: 3-class block classification (BC), class-based search strategy (SS), and block partitioning (BP).

PSNR (in dB)													
Algorithm	Campus	Class	Hall	Ice	Office	Square	Bank	Crossroad	Intersect	Mainroad	Overbridge	Pedestrian	Average
Proposed	35.0525	34.2832	33.5433	28.9309	30.0303	31.6335	32.8658	27.067	32.3646	26.4812	30.972	30.2226	31.1206
SS+BP	-0.0263	0.0021	-0.0413	0.0070	-0.0971	-0.0431	-0.0155	-0.0270	0.0011	-0.2592	-0.0445	-0.0518	-0.0496
BC+BP	0.0627	0.1213	0.0643	0.0758	0.1090	-0.1014	-0.0050	0.2306	-0.3002	-3.1928	-0.0327	-0.4706	-0.2866
BC+SS	0.0135	0.0136	-0.0360	0.0080	-0.0093	-0.0215	-0.0111	0.0107	0.0030	0.0039	-0.0370	-0.1426	-0.0171
Motion search complexity (ANCPB)													
Algorithm	Campus	Class	Hall	Ice	Office	Square	Bank	Crossroad	Intersect	Mainroad	Overbridge	Pedestrian	Average
Proposed	1.1334	1.7343	1.1414	4.1082	2.1701	2.6796	0.7464	2.5171	6.3104	3.3798	1.2049	7.1805	2.8588
SS+BP	1.0254	1.2629	1.0227	3.8589	2.0002	2.5281	0.6870	2.3242	3.1179	3.3471	1.1252	6.6849	2.4154
BC+BP	5.0241	8.5913	4.9901	19.7940	8.4303	12.1350	2.7297	11.6462	28.9561	9.6702	4.2971	29.0001	12.1053
BC+SS	1.8089	1.9550	1.8386	4.6634	2.8293	3.2716	1.4665	3.0866	3.7022	4.0191	1.8945	6.9065	3.1202

Table 4.3 : PSNR and motion search complexity (ANCPB) comparisons for different traditional and surveillance-based block matching algorithms.

PSNR (in dB)													
Algorithm	Campus	Class	Hall	Ice	Office	Square	Bank	Crossroad	Intersect	Mainroad	Overbridge	Pedestrian	Average
FS	35.2069	34.4575	33.6495	29.0614	30.3416	31.5965	32.8852	27.6075	32.2507	23.3813	31.2733	29.8607	30.9643
DS	-0.1145	-0.1198	-0.0568	-0.2114	-0.6990	-0.4534	-0.0165	-1.0377	-0.8369	-0.7596	-0.7872	-0.7626	-0.4879
TZS	-0.0185	-0.0243	-0.0160	-0.0623	-0.1396	-0.0555	-0.0066	-0.0732	-0.2235	-0.2088	-0.1548	-0.1775	-0.0967
MHGS	-0.0455	-0.0590	-0.0543	-0.0657	-0.1841	-0.0734	-0.0121	-0.1304	-0.2448	-0.2443	-0.2602	-0.2167	-0.1325
FAME	-0.0645	-0.0825	-0.0799	-0.2830	-0.6673	-0.5380	-0.0185	-0.2179	-0.7251	-0.7440	-0.5217	-1.0666	-0.4174
BFDS	-0.0946	-0.0500	-0.0427	-0.0633	-0.2453	-0.0815	-0.0179	-0.1336	-0.2265	-0.2129	-0.1621	-0.2206	-0.1293
BFDS+MHGS	-0.1213	-0.0850	-0.0811	-0.0664	-0.2962	-0.0996	-0.0231	-0.1835	-0.2479	-0.2456	-0.2674	-0.2630	-0.1650
BFDS+FAME	-0.1336	-0.1063	-0.1121	-0.2824	-0.8711	-0.5588	-0.0297	-0.2819	-0.7292	-0.7515	-0.5266	-1.1168	-0.4583
Proposed	-0.1543	-0.1743	-0.1063	-0.1306	-0.3113	0.0370	-0.0194	-0.5405	0.1139	3.0999	-0.3013	0.3619	0.1562
(WPSNR)	-0.1822	-0.1918	-0.1206	-0.1414	-0.3434	0.0750	-0.0257	-0.6281	0.1339	3.5498	-0.3855	0.4163	0.1797
Motion search complexity (ANCPB)													
Algorithm	Campus	Class	Hall	Ice	Office	Square	Bank	Crossroad	Intersect	Mainroad	Overbridge	Pedestrian	Average
FS	289.000	289.000	289.000	289.000	289.000	289.000	289.000	289.000	289.000	289.000	289.000	289.000	289.000
DS	13.898	13.869	13.726	14.942	15.172	14.971	13.192	14.969	14.723	15.051	13.401	17.730	14.637
TZS	31.341	33.296	32.169	30.745	30.461	29.772	29.664	29.373	28.701	27.419	28.977	27.298	29.935
MHGS	25.366	27.059	26.168	25.406	24.962	24.196	23.740	24.339	23.391	22.457	23.313	23.447	24.487
FAME	5.539	5.765	5.587	5.758	5.670	5.419	5.155	5.635	5.401	5.192	5.119	5.652	5.491
BFDS	3.193	5.203	4.003	13.215	4.441	6.898	2.341	6.079	9.085	4.341	2.953	14.132	6.324
BFDS+MHGS	2.934	4.437	3.551	11.621	4.107	6.104	2.208	5.760	7.976	4.309	2.813	12.928	5.729
BFDS+FAME	1.462	1.660	1.503	3.249	1.820	2.158	1.249	2.249	2.594	1.887	1.395	3.735	2.080
Proposed	1.133	1.734	1.141	4.108	2.170	2.680	0.746	2.517	6.310	3.380	1.205	7.181	2.859

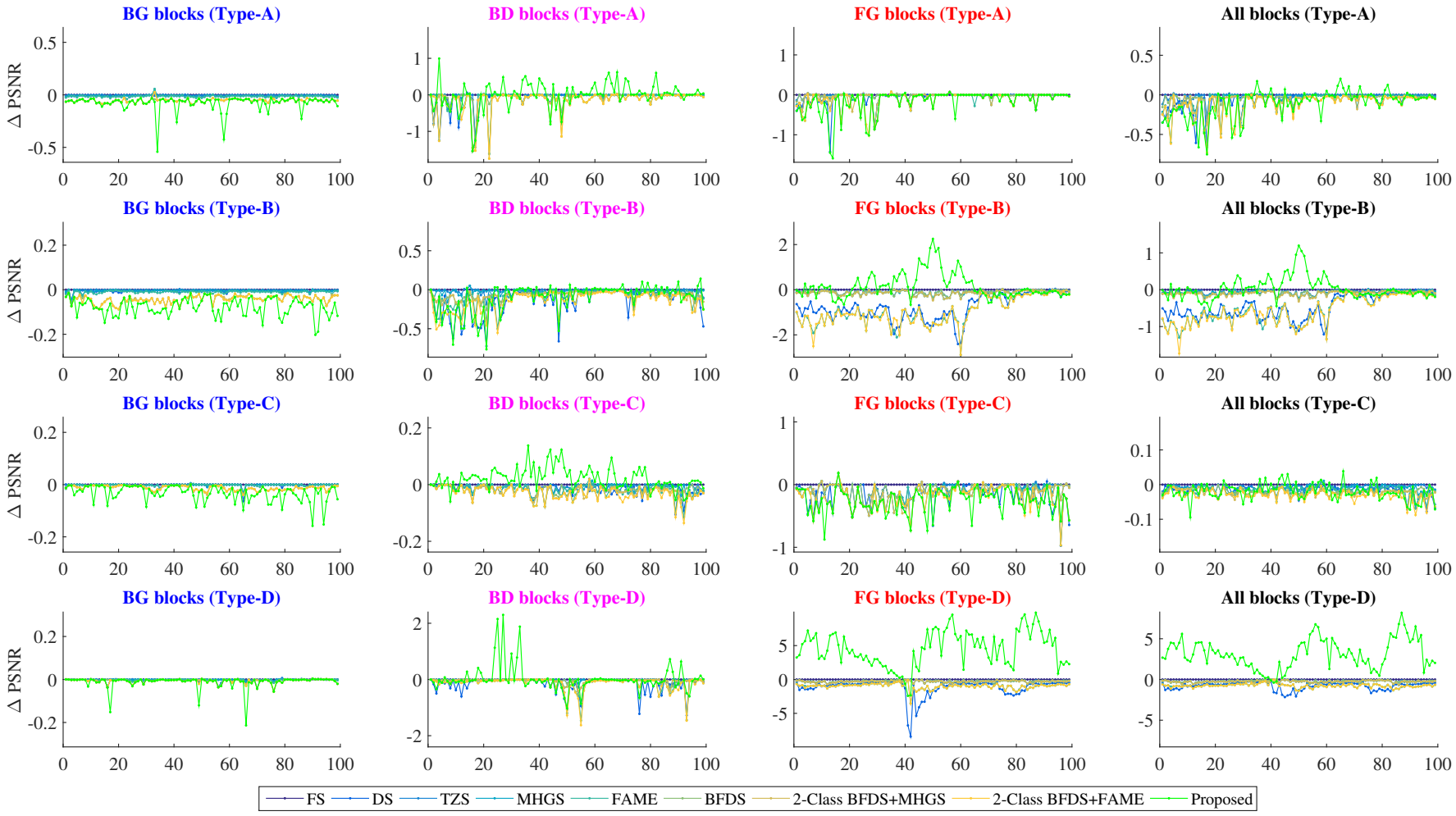

Figure 4.15 : Block-class-wise PSNR (in dB) comparison for different algorithms against FS. Row 1-4 corresponds to different sequences: Hall monitor (type-A), Square (type-B), Bank (type-C), and Mainroad (type-D), respectively. (Best viewed in color)

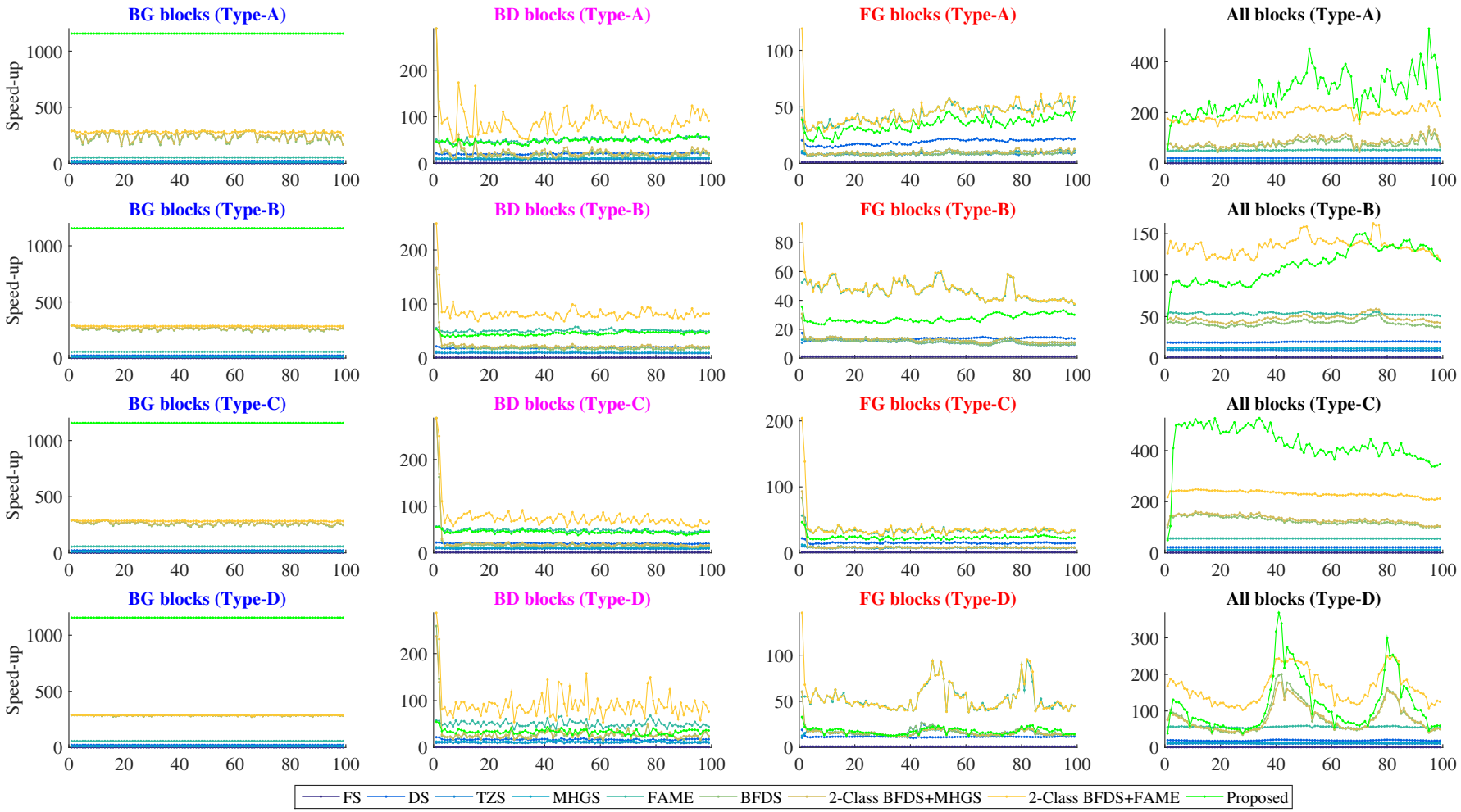

Figure 4.16 : Block-class-wise Speed-up (in times) comparison for different algorithms against FS. Row 1-4 corresponds to different sequences: Hall monitor (type-A), Square (type-B), Bank (type-C), and Mainroad (type-D), respectively. (Best viewed in color)

4.4 SUMMARY

In this chapter, a fast background-foreground-boundary aware motion search algorithm is presented for surveillance videos. The characteristics of surveillance videos are studied with primary emphasis on the MV distribution. The experiments on twelve surveillance video sequences indicated that two-level block classification is insufficient to handle MV characteristics. This created the necessary background for our proposal of using a three-level block classification scheme. We proposed to use an additional class called boundary in addition to background and foreground classes. We presented an adaptive threshold selection scheme for block classification. Then, for each class, separate search strategies were investigated to increase the efficacy of the motion search algorithms for surveillance videos. In addition, we also presented direction-oriented search patterns to handle the motion characteristics of the boundary and foreground blocks. The block matching accuracy is further improved by our boundary block region partitioning scheme. A total of five sub-regions were chosen for BD blocks, and the sub-regions belonging to the FG region are only used for the block matching process. This process has significantly improved the efficacy of the proposed scheme for BD blocks. The proposed approach provided better matching quality as compared to other existing algorithms. At the same time, it has also been able to provide four times speed-up for BG blocks and more than two times speed-up for BD and FG blocks as compared to the BFDS method. Experimental results indicate that the proposed scheme outperformed not only existing methods in PSNR but also reduced computational complexity. With this scheme, the second objective of the thesis, to develop a computationally efficient motion search algorithm to exploit special characteristics of the surveillance videos, is achieved.

Surveillance videos are essential for security purposes, not only because of video content but also due to human action movement information. The human action movement information can play a crucial role in security applications, where accurate analysis is of utmost importance. Hence, the human skeleton information extracted from surveillance videos needs to be stored in addition to original video data. Thus, it becomes a non-trivial problem to encode skeleton information efficiently. The higher accuracy in critical security applications demands original and reliable skeleton information. The presence of a large number of human skeletons in the surveillance video scenario imposes additional space constraints. Hence, the next chapter is dedicated to addressing the storage challenges that arise due to skeleton sequences.

...