

## **CHAPTER 4**

### **DESIGN OF FRAME AND REGION SELECTION ALGORITHMS**

As mentioned in Chapter 1, a video watermarking technique consists of three main steps, namely, embedding the copyright watermark, broadcasting and extraction of the copyright information to allow or prevent a person to use the service of interest. Among these steps, the first step, that is, embedding or insertion of copyright watermark into the cover medium has to be designed carefully and should make the hacking of watermark complex. The decision of where to embed the watermark is a critical question in this step. For this purpose, the following characteristics of robustness are always desired.

- (i) Should introduce minimum perceptual distortion
- (ii) The frames and regions selected for embedding should not be similar or identical and should exhibit a random pattern

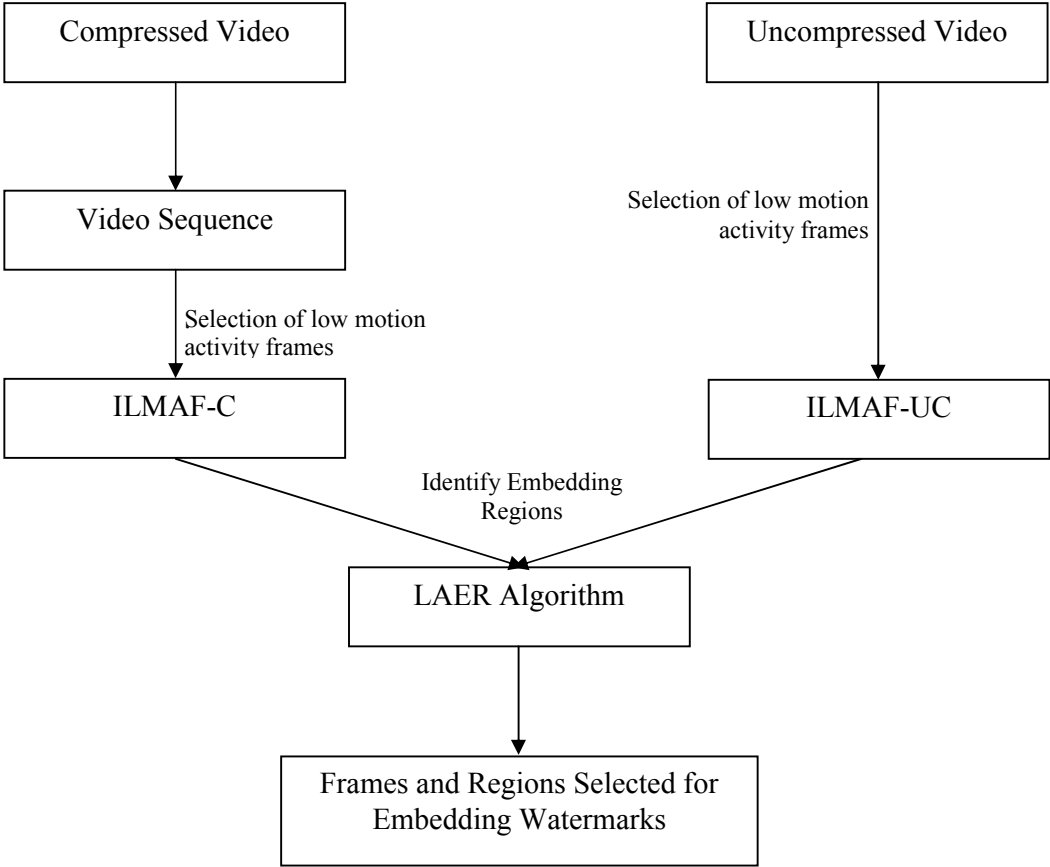
To achieve the above robustness characteristics in the proposed video watermarking algorithms, a 2-step algorithm is used. The first step uses scene-change detection algorithm to estimate motion activity in video frames. The result is used to detect optimal random frames that will introduce minimum distortion. In the next step, a labeling algorithm is used to select regions or blocks within the frames selected for embedding the copyright watermark. After selection of embedding regions, techniques like visual cryptography and transformation techniques are used to insert the watermark in different scenes of the video.

Phase I of the study focuses on the design and development of such an algorithm and the two stages involved is presented below.

Stage 1 : Identify low motion activity frames where the insertion of watermark will not be visible to human visual system

Stage 2 : Select regions from these frames that will have minimum impact on visual quality after insertion of watermark.

The method of identifying low motion activity frames differ for uncompressed and compressed video files and separate methods are used for this purpose. The proposed algorithms are respectively termed as ILMAF-C (Identification of Low Motion Activity Frames Algorithm for compressed domain) and ILMAF-UC (Identification of Low Motion Activity Frames Algorithm for Uncompressed domain) for selecting optimal frames from compressed and uncompressed domains. A Labeling Algorithm to identify Embedding Region (LAER) algorithm is used to select regions that minimizes the distortions introduced by the watermark is then used. The steps involved in the proposed algorithms to select frames and regions from raw uncompressed and compressed videos for embedding the copyright information consists of the steps presented in Figure 4.1.



**Figure 4.1 : Steps of Proposed Frame and Region Selection Algorithm**

#### 4.1. SCENE DETECTION AND VIDEO WATERMARKING

Detection of scene changes play an important role in video processing with many applications ranging from watermarking, video indexing, video summarization to object tracking and video content management (Shukla and Sharma, 2012). Scene change detection is an operation that divides video data into physical shots (Chen *et al.*, 2001). Over the last four decades, scene change detection has been widely studied and researched. As a result, many scene change detection techniques have been proposed and published in the literature (Li *et al.*, 2006). These algorithms can be broadly classified as operating on decompressed data (pixel domain) or working directly on the compressed data (compressed domain) (Lil *et al.*, 2006). Scene changes are divided into two types:

- (i) Abrupt scene change
- (ii) Gradual scene change

Abrupt scene changes result from editing “cuts” and detecting them is called cut detection either by colour histogram comparison on the uncompressed video or by DCT coefficient comparison. Gradual scene changes result from chromatic edits, spatial edits and combined edits. Gradual scene changes include special effects like zoom, camera pan, dissolve and fade in/out (Ngo *et al.*, 2000).

A video scene is composed of a number of shots, where a shot is defined as a part of the video that results from one continuous recording by a single camera. The gap between two shots is called as a shot boundary. There are mainly four different types of common shot boundaries within shots. They are,

- (i) Cut
- (ii) Fade
- (iii) Dissolve
- (iv) Wipe

There have been a number of various approaches to handle different shot boundaries (Gonzalez and Woods, 2008). These approaches can be generally grouped as Shot

Boundary Detection based on Color Diagrams, Edge Detection and Shot Boundary Detection Using Macroblocks. This study uses the third type of approach for scene change detection, where a motion activity descriptor is used on the macroblocks to classify video frames as very low, low, medium, high and very high activity frames.

The use of scene change detection with the proposed video watermarking algorithm is to make the algorithms robust against attacks like frame dropping, averaging, swapping and statistical analysis. Further, the use of region selection using labeling algorithm and combination of visual cryptography, makes sure that the watermark embedding is performed on least motionless scenes in order to make it robust against attacks. The usage of scene-based watermarking scheme and visual cryptography ensures the non-requirement of the original and watermarked medium. Additionally, this scheme also gives the perfect solution for where to do watermarking in video, so that it becomes robust against several attacks.

#### 4.2. IMAGE SEQUENCE GENERATION

The first step of the frame and region selection algorithm is to convert the input video (MPEG format) to a sequence of frames. The procedure used is given in Figure 4.2 and an example frame sequence is shown in Figure 4.3.

Input : Video file (V)	Output : Sequence of Frames
Step 1	: Read V and create a multimedia reader object (readerObj)
Step 2	: Read in video frames from V (vidFrames)
Step 3	: Get Number of Frames from readerObj (numFrames)
Step 4	: for I = 1 to numFrames
	Step 4a : Add frame at i <sup>th</sup> position into frame sequence set
Step 5	: Output video sequence set

**Figure 4.2 : Image Sequence Generation Procedure**



**Figure 4.3 : Video Sequence Extracted from Foreman MPEG File (Partial)**

### **4.3. MOTION ACTIVITY FRAME DETECTION**

The main task during the selection of frames to embed the copyright watermark is the identification of low activity frames in the video scenes. For this purpose a motion activity frame detection algorithm for uncompressed and compressed domain is proposed. Both the algorithms identify very low, low, medium, high and very high activity frames using motion intensity and activity matrix obtained from Motion Vector (MV) extracted from Macroblocks. The very low, low and medium frames are only used during the embedding process. Similarly, both the algorithms selects only medium and low complex regions for embedding. Thus, the general steps involved are

- (i) Extract Motion Vector from uncompressed or compressed video
- (ii) Estimate motion features
- (iii) Classify frames with respect to motion activity
- (iv) From very low, low and medium activity frames identify medium and low complex regions for embedding the copyright watermark

#### **4.3.1. Motion Features**

The magnitude of the motion vectors represents a measure of intensity of motion activity that includes several additional attributes that contribute towards the efficient use of these motion descriptors in a number of applications. Intensity of activity (Jeannin and Divakaran, 2001) is expressed by an integer in the range (1-5) and higher the value of intensity, higher the motion activity.

The motion features are estimated using the motion vectors extracted from uncompressed and compressed videos respectively. After motion vector extraction, the motion features are determined as follows. Consider  $x(i, j)$  and  $y(i, j)$  as motion vectors in x and y directions for a given frame, where  $(i, j)$  indicates the block indices. The spatial activity matrix is then estimated using Equation (4.1) and the average of activity matrix for each frame is estimated using Equation (4.2).

$$C_{mv} = \{R(i, j)\} \quad (4.1)$$

where  $R(i, j) = \sqrt{(x(i, j)^2 + y(i, j)^2)}$ .

$$C_{mv}^{avg} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C_{mv}(i, j) \quad (4.2)$$

where M and N are the width and height of the macroblocks in the frame. The intensity of motion for each frame is determined using Equation (4.3)

$$\sigma_{fr}^2 = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C_{mv}(i, j) - C_{mv}^{avg})^2 \quad (4.3)$$

The intensity values of motion of each frame are then classified into five categories using simple rules as mentioned in Table 4.1 The five categories are very low, low, medium, high and very high activity.

**TABLE 4.1**  
**MOTION ACTIVITY CLASSIFICATION**

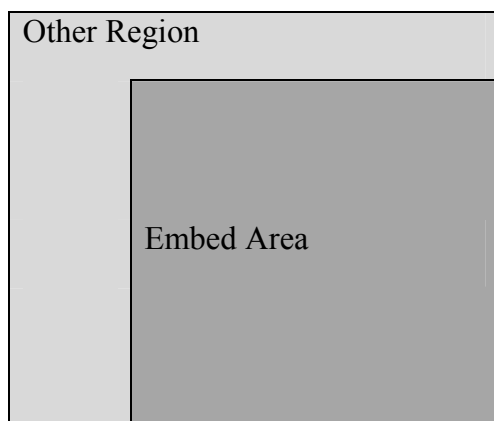
<b>Activity Level</b>	<b>Intensity of Motion (<math>\sigma</math>)</b>
Very low Intensity	$0 \leq \sigma \leq 3.9$
Low Intensity	$4 \leq \sigma \leq 10.7$
Medium Intensity	$10.8 \leq \sigma \leq 17.1$
High Intensity	$17.2 \leq \sigma \leq 32.0$
Very High Intensity	$32 > \sigma$

The motion activity is used in different applications such as video surveillance, fast browsing, dynamic video summarization and content-based querying. This information in the proposed algorithm is used for shot boundary detection and frame

classification. After frame classification the LAER is used to identify potential watermarking regions.

#### 4.3.2. LABELING ALGORITHM TO IDENTIFY EMBEDDING REGIONS (LAER)

The LAER is used to identify the embedding regions in the selected frames of the video. The algorithm treats the selected frames as a sequence of images and for each image, classifies the image as Embed Region (ER) and Other Region (OR). For this purpose, the frame is divided into 32 x 32 sized blocks. Sobel edge detector is used and the non-edge region alone is considered for further analysis. This is illustrated in Figure 4.4.



**Figure 4.4 : Embed and Other Regions**

A block is defined as embed region if the measure of Inhomogeneity is greater than 10 and average contrast is less than 50. Inhomogeneity is calculated as the ratio of the product between the mean and Uniformity and the product between standard deviation and smoothness (Equation 4.4).

$$\text{Inhomogeneity} = \frac{\mu \times U}{\sigma \times R} \quad (4.4)$$

where  $m$  is the mean,  $U$  is the uniformity measure,  $R$  is the smoothness measure and  $\sigma$  is the standard deviation of the block and are calculated using Equation (4.5), (4.6), (4.7) and (4.8) respectively.

$$m = \sum_{i=0}^{L-1} I_i h(I_i) \quad (4.5)$$

$$U = \sum_{i=0}^{L-1} h(I_i)^2 \quad (4.6)$$

$$R = 1 - \frac{1}{1 + \sigma^2} \quad (4.7)$$

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (I_i - m)^2 h(I_i)} \quad (4.8)$$

In the above equations,  $I_i$  represent the gray level intensity,  $L$  is the number of possible gray level intensities in the block and  $h(I)$  represent the histogram of the intensity levels.

The copyright bits are embedded in the data embed region. The data embed region is further divided into many blocks. Each of these blocks, are classified into three categories, namely low complex (smooth), medium complex and high complex regions. For this purpose, two threshold values ( $T1$  and  $T2$  with  $T1 \geq T2$ ) are used along with the surrounding block's variances to predict if the target block is a low complex, medium complex or high complex block. No data is embedded in the high complex block. 2/3<sup>rd</sup> of embedding is done in smooth and rest is done in medium frames.

The referring block locations used is the same as that given in Figure 4.5. Here A represents the left block, B represents the top block, C denotes top-right and D denotes top-left blocks. The current block is denoted by E. The variance is calculated as  $\sigma^2$ . The following rule based procedure (Figure 4.6) is used in the classification of low complex, medium complex and high complex blocks. In the experiments,  $T1 = \infty$  and  $T2 = 0$  after empirical evaluation.

D	B	C
A	E	

**Figure 4.5 : Block Locations Used for MV Prediction**

Rule 1 : If the four variances of blocks A, B, C and D are all larger than TH1, then E (current block) is predicted to be a complex block.

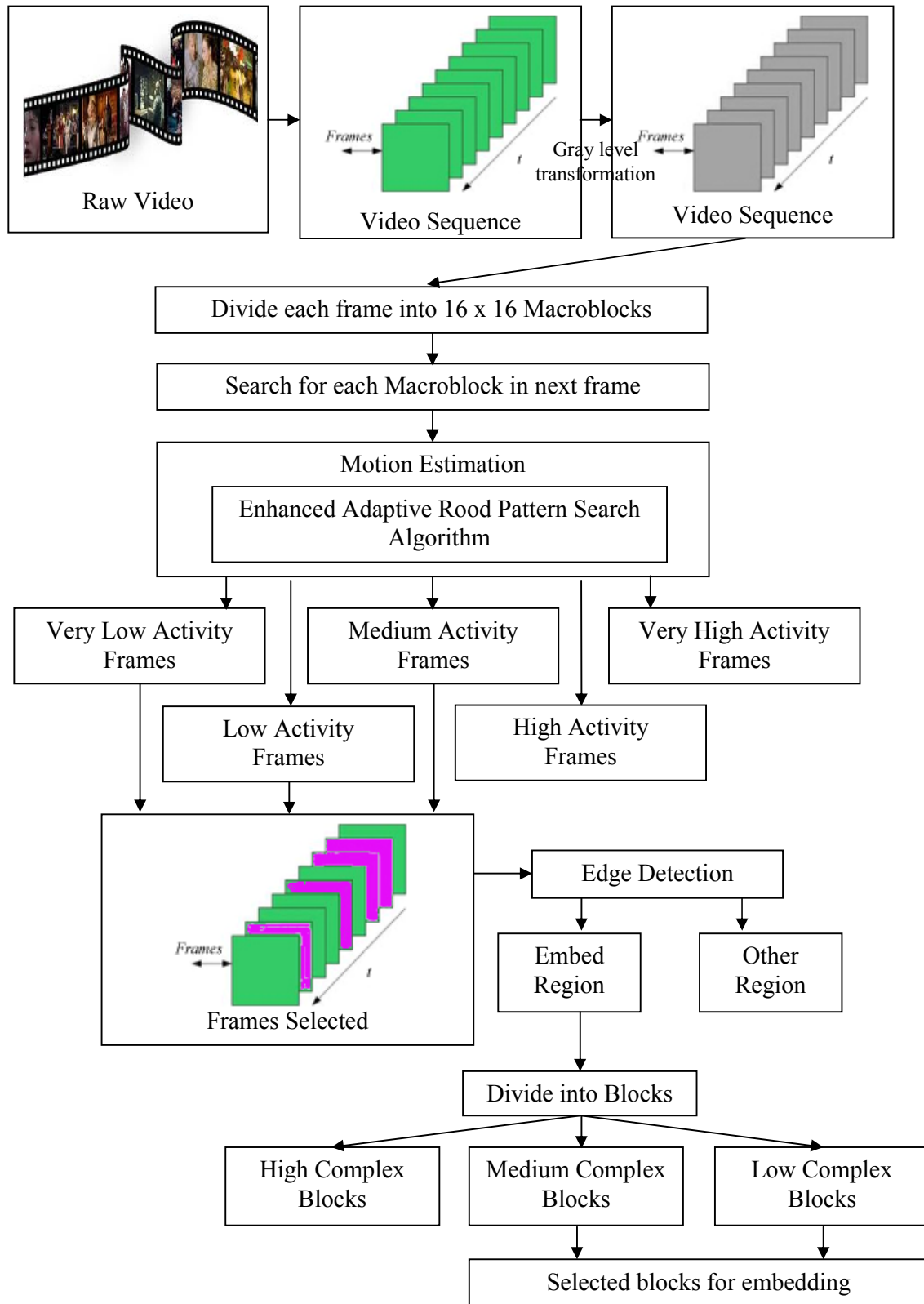
Rule 2 : If one or more variances of blocks A, B, C and D are larger than TH2 but less than TH1, then E is predicted to be a normal block.

Rule 3 : If the four variances are all smaller than TH2, then E is predicted to be a low complex block.

**Figure 4.6 : Classification of Blocks Procedure**

### 4.3.3. ILMAF-UC Algorithm

The block diagram of the ILMAF-UC algorithm is shown in Figure 4.7. Block matching techniques are employed to extract the motion vectors. To extract the motion intensity, the motion of the moving objects has to be estimated first. This is performed using the Block Matching Algorithm (BMA), which is used to retrieve an initial estimate of the image displacement. In the proposed algorithm, to obtain a dense displacement field, matching with adaptive block sizes was implemented. In this algorithm, a frame is divided into squared blocks of size  $M \times M$  pixels. It is assumed that that each block undergoes translation only with no scaling or rotation. The blocks in the first frame are compared with the blocks in the second frame. Motion vectors are then calculated for each block to determine where each block from the first frame ends in the second frame.



**Figure 4.7 : Steps in ILMAF-UC Algorithm**

In this study, for motion vector extraction, the block matching algorithm is used to extract the motion intensity feature. For this purpose, a variety of BMAs have been proposed in literature. Some examples of BMA include Exhaustive Search (ES), Diamond Search (DS), Three-Step Search (TSS), New Three Step Search (NTSS), Simple and Effective Search (SES), 4 Step Search (4SS) and Adaptive Rood Pattern Search (ARPS). In the present work, an Enhanced Adaptive Rood Pattern Search (EARPS) algorithm is used. After motion vector estimation, the motion features, as explained in the previous section are used to classify the frames.

ARPS algorithm (Ferman and Tekalp, 2003) makes use of the fact that the general motion in a frame is usually coherent, i.e. if the macro blocks around the current macro block moved in a particular direction then there is a high probability that the current macro block will also have a similar motion vector. This algorithm uses the motion vector of the macro block to its immediate left to predict its own motion vector. The conventional ARPS algorithm is presented in Figure 4.8 and an example is shown in Figure 4.9.

The predicted motion vector points to (3, -2). In addition to checking the location pointed by the predicted motion vector, it also checks at a rood pattern distributed points, as shown in Figure 4.9, where they are at a step size of  $S = \text{Max}(|X|, |Y|)$ . X and Y are the x-coordinate and y-coordinate of the predicted motion vector. This rood pattern search is always the first step. It directly puts the search in an area where there is a high probability of finding a good matching block. Peak-Signal-to-Noise-Ratio (PSNR) characterizes the motion compensated image that is created by using motion vectors and macro clocks from the reference frame.

- **Enhanced Adaptive Rood Pattern Search (EARPS) algorithm**

The proposed EARPS algorithm consists of three steps as given below.

A. Initial Search Centre Prediction.

B. Adaptive Search Pattern.

### C. Early Search Termination.

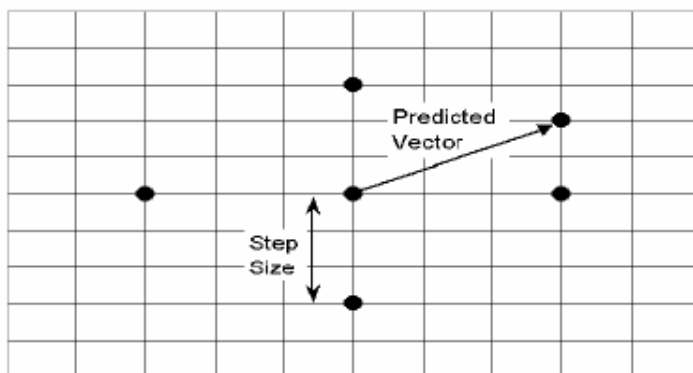
Step 1: The current block's MV is predicted by referring to the left adjacent block. The ARP's length  $\Gamma$  is determined by the following rule:  
 If the current block is a leftmost boundary block of the frame  
 $\Gamma=2$ ;  
 else  
 $\Gamma = \max \{ |MV_{\text{predicted}}(x)|, |MV_{\text{predicted}}(y)| \}$ .

Step 2: The center of the ARP is positioned at the center of the search window. The four arms are of equal length  $\Gamma$ , which has been computed in Step 1.

Step 3: The refined local search is conducted by using a fixed unit-size rood pattern (URP). Set the center point of the URP at the MME (Minimum Matching Error) point found in the previous step and compute SAD for all the four vertex points.

Step 4: If the new MME point locates on any vertex point, repeat Step 3 step until the MME point locates on the center point of the current URP. The MV is found which corresponds to the MME point.

**Figure 4.8 : Traditional ARPS Algorithm**



**Figure 4.9 : Adaptive Rood Pattern: The predicted motion vector is (3,-2) and the step size  $S = \text{Max}(|3|, |-2|) = 3$**

### **(A) Initial Search Centre Prediction**

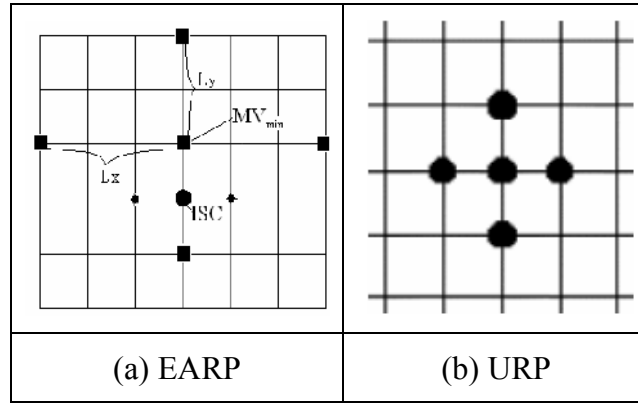
As most of the motion vectors are well correlated with those of their neighboring blocks, the algorithm takes advantage of the neighboring motion vectors to predict the Initial Search Center (ISC) of the current block. Although temporal prediction can provide promising candidates, it needs to record the entire previous MV field, which might be undesirable in practical implementation with limited storage space. Therefore, the method exploits only the spatial correlation and adopts the left (A), top (B) and top-right(C) adjacent blocks of the current block (E) as the referenced blocks for prediction (Figure 4.5).

The blocks in other positions nearby are less correlated to the current block and thus are not reliable, so they are not used in the proposed algorithm. The MV median prediction method is used to obtain MV predictor. Sometimes the median predictor may result from a block that is far away from the true MBD (Minimum Block Distortion) point. To avoid this issue, the proposed algorithm sets the position with the minimum SAD (Sum of Absolute Difference) as the initial search centre.

### **(B) Adaptive Search Pattern**

The search-point configuration used in the EARPS is divided into two different shapes: Enhanced Adaptive Rood Pattern (EARP) (Figure 4.10a) and Unit-size Rood Pattern (URP) (Figure 4.10b). Initially, the choice of the rood shape is based on the observation of the motion feature of real-world video sequences. It is known that the MV distribution in horizontal and vertical directions is higher than that in other directions, as most of camera movements are in these directions (Zhao *et al.*, 2008). As the rood shape pattern includes all the horizontal and vertical directions, it can quickly detect such motions and the search will directly “jump” into the local region of global minimum.

Next, the MV is decomposed into a vertical MV component and a horizontal MV component. For a moving object which may introduce a MV in any direction, the rood-shaped pattern can at least detect the major trend of the moving object, which is the desirable outcome in the initial search stage.



**Figure 4.10 : Search Patterns Used in EARPS**

Even if the predicted MV could be inaccurate and its magnitude does not match the true motion very well, the rood-shaped pattern which takes all horizontal and vertical directions into consideration can still track the major direction and favor the follow-up refinement process. Unlike the pattern adopted in the traditional ARPS, the enhanced ARPS defines the search pattern using Equations (4.9) and (4.10).

$$L_x = \max\left[|MV_{L.x} - MV_{min.x}|, |MV_{T.x} - MV_{min.x}|, |MV_{TR.y} - MV_{min.y}|\right] \quad (4.9)$$

$$L_y = \max\left[|MV_{L.y} - MV_{min.y}|, |MV_{T.y} - MV_{min.y}|, |MV_{TR.y} - MV_{min.y}|\right] \quad (4.10)$$

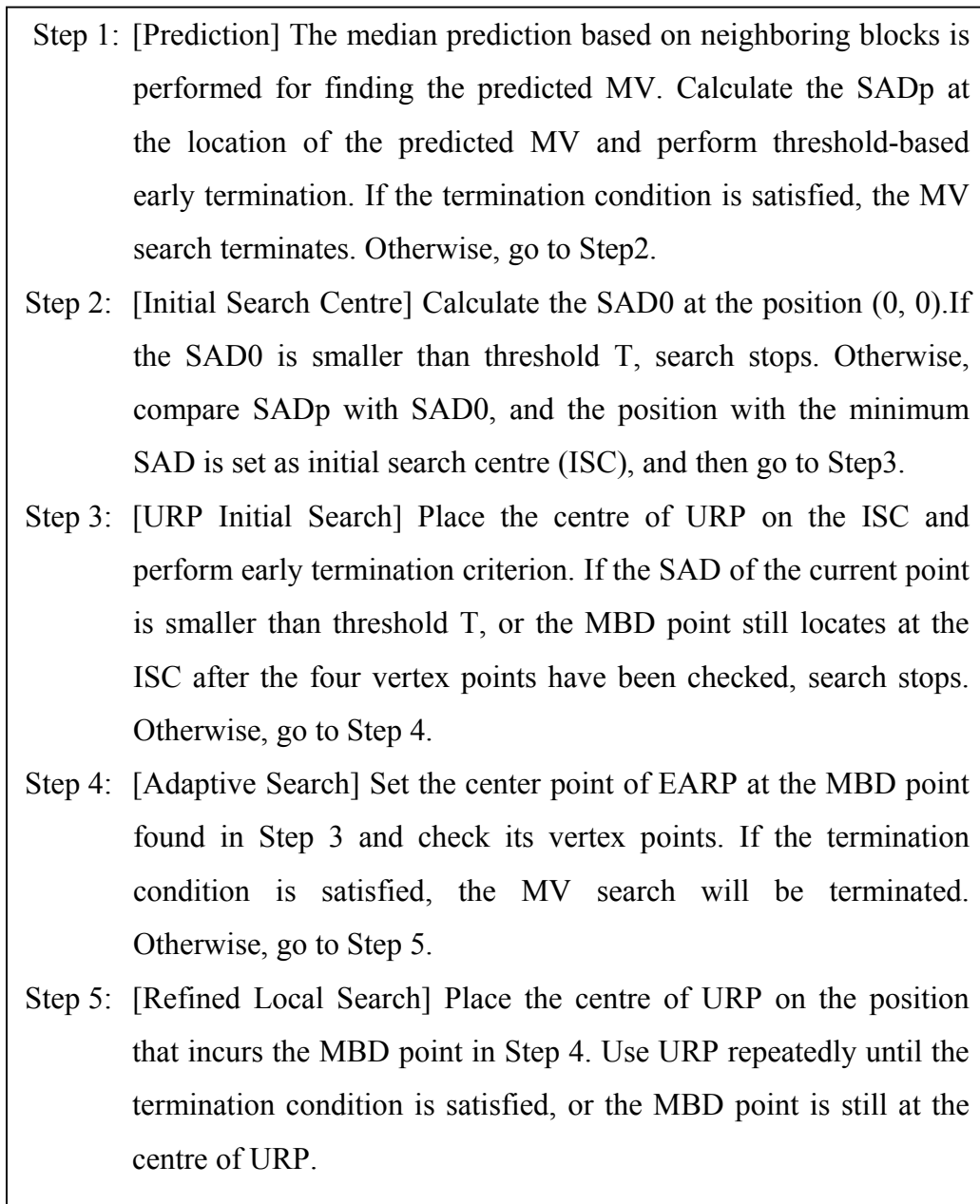
where  $L_x$  and  $L_y$  are the EARP's horizontal and vertical arm lengths.  $(MV_{L.x}, MV_{L.y})$ ,  $(MV_{T.x}, MV_{T.y})$  and  $(MV_{TR.x}, MV_{TR.y})$  are the motion vectors of the left, top and top-right adjacent blocks of the current block.  $MV_{min.x}$  and  $MV_{min.y}$  are the horizontal and vertical components of the MBD (Minimum Block Distortion) point found in the URP initial search. Operator "max" is used to find the maximum value. The patterns used in the enhanced algorithm are shown in Figure 4.5. Here, the EARP is not applicable to boundary blocks in each frame. For those blocks, a fixed-size arm length of 2 pixels is used for the EARP as ARP.

An URP is introduced in the initial search step of EARPS. Since the MBD point occurs more frequently around the ISC, an URP is used at the beginning of the search. In the initial search step, the URP is used only once. When the motion vector is not surrounded by the initial search centre during the early search, then the EARP is adopted to locate the large motion. Again, the EARP is used only once. The URP is then used again for the last refined local search. Here, the EARP is shrunk to the URP when both arm lengths are equal to 1 or even to the centre point itself, if both lengths are 0.

### **(C) Early Search Termination**

The Early Search Termination (EST) is achieved by comparing the SAD between the current block and the referenced block against a threshold  $T$ . If the SAD is smaller than the threshold  $T$ , the search stops and outputs the current referenced block as the optimal solution. It was reported in ARPS that early termination of the search for the optimal motion vector could substantially reduce the cost of block-matching without degrading the video quality much. The threshold  $T$  is set to 512, as it achieves fairly good speed-up gain without causing noticeable degradation on visual quality.

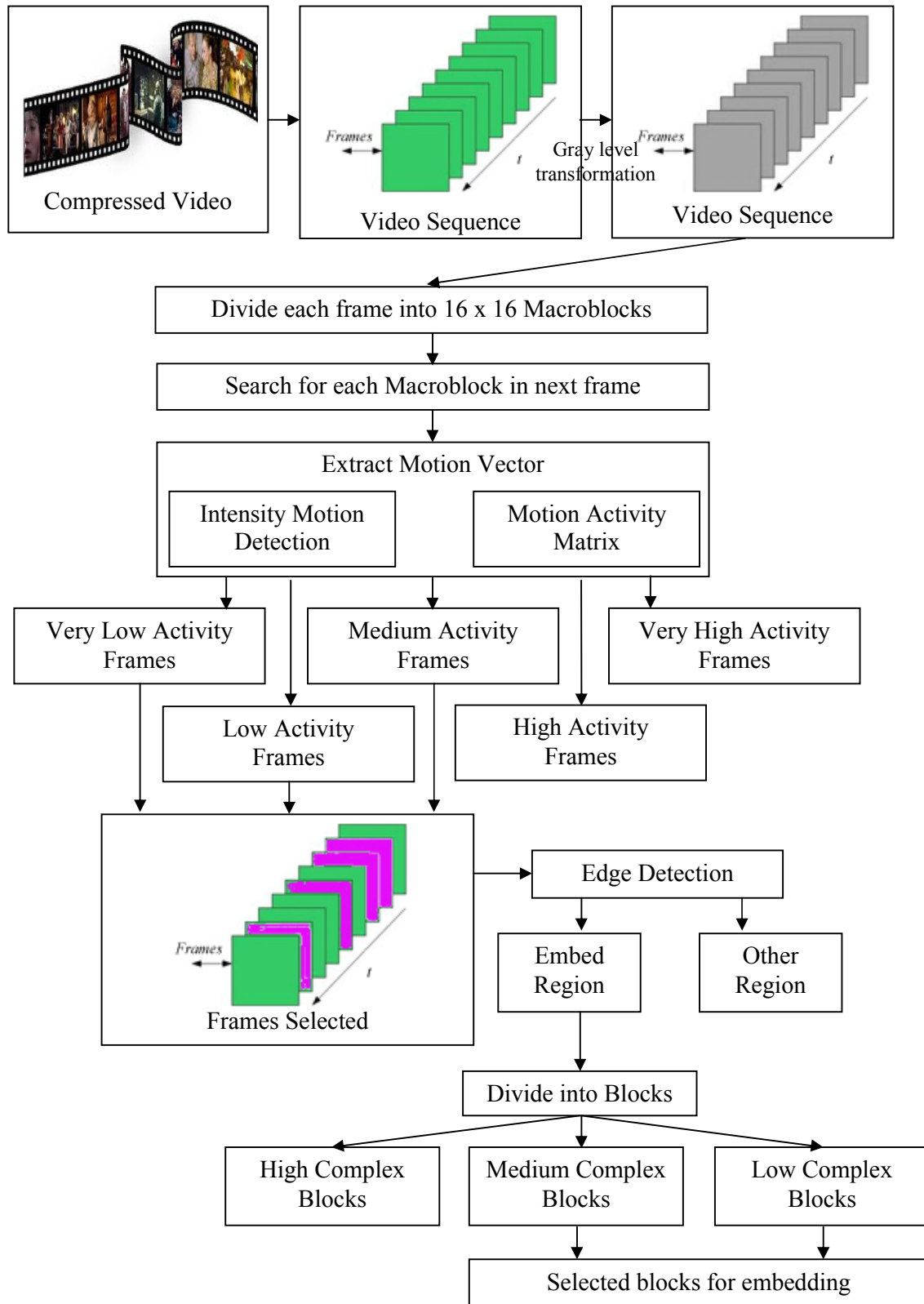
In the enhanced algorithm, three blocks are used for obtaining the median predictor and they are the left, top and top-right adjacent blocks of the current block. The predicted MV in EARPS is the median of their motion vectors. For each block, the proposed EARPS performs the steps summarized in Figure 4.11. It can be seen that the proposed search pattern in Step 4 is used only once, thus enhancing the BMA.



**Figure 4.11 : Steps in EARPS**

#### **4.3.4. ILMAF-C Algorithm**

The method used to detect low motion activity frame from compressed domain is presented in Figure 4.12.



**Figure 4.12 : Steps in ILMAF-C Algorithm**

The motion activity is one of the motion features included in the visual part of the MPEG-7 standard. It is also used to describe the level or intensity of activity, action, or motion in that video sequence. In the proposed algorithm, the low or high action in a frame is used as a measure of how much a video segment is changing (Amel *et al.*, 2010). For this purpose, a procedure that extracts motion vectors from the bit stream is used. The motion vector information is contained in the macroblock layer. The motion vectors have to be stored in an order that will allow the motion from frame to frame to be calculated.

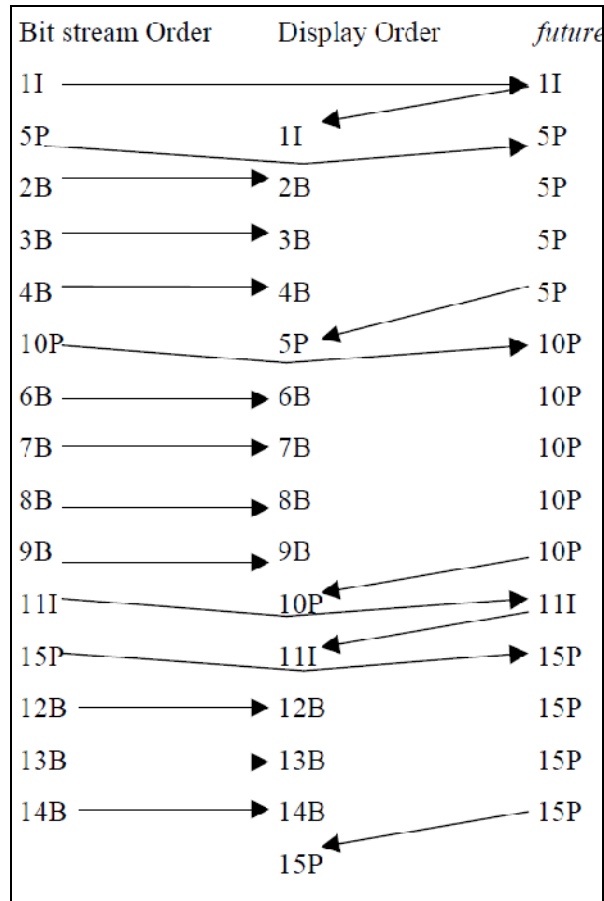
- **Reordering the bit stream order to the display order**

The frames of a video do not come into the decoder in the same order as they are displayed. Thus reordering of frames is necessary. The procedure is given below.

If an I or P frame (denoted as“1”) arrives in, it is put in a temporary storage called Future. The I and P frames always come into the decoder before the B frames that reference them. “1” remains in future until another I or P frame (“5”) comes in. The arrival of “5” indicates it is “1’s” turn in the display order. “1” is taken out of future, put in the display order. “5” is put in future until another I of P frame arrives. All B frames are immediately put in the display order. At the end whatever frame is left in future is taken out and put in the display order. A typical bit stream is shown in Figure 4.13 along with the display order number of each frame.

- **Storing the motion vectors**

For ease of handling, it was decided that the motion vectors should be stored in two dimensional arrays. The size of the array corresponds to the frame size (in macroblocks). The position of the entry in the array corresponds to the macroblock’s position in the frame.

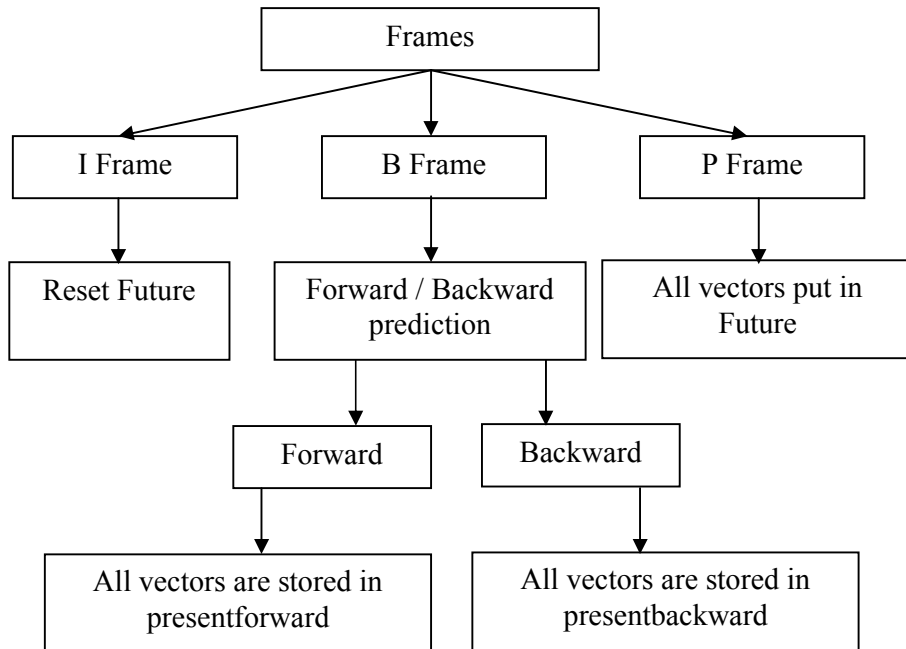


**Figure 4.13 : Converting from Bit Stream Order to Display order**

There is a separate array for the two components of the vector, one for the right component and one for the left component. To allow the storage of all the vectors that may be present in a frame, four arrays have to be created. Two arrays are needed for the storage of the forward predicted vectors, and two for backward predicted vectors. To find the motion from one frame to another, a record of the motion vectors in the previous frame has to be kept. This means four more arrays have to be created. Finally the motion vectors in a P frame have to be stored until it is the P frames turn in the display order. As a P frame can only have forward predicted vectors, only two arrays need to be created. Thus a total of 10 arrays were used (Table 4.2). The block diagram of the procedure used to store motion vectors from the bit streams is given in Figure 4.14.

**TABLE 4.2**  
**ARRAYS USED**

Array Name	Description
FutureRight	Store the motion vectors in a P frame until it is the P frames turn in the display order
FutureDown	
presentForwardRight	Stores the motion vectors of the present frame in the display order
presentForwardDown	
presentBackwardRight	
presentBackwardDown	
pastForwardRight	Stores the motion vectors of the previous frame in the display order
pastForwardDown	
pastBackwardRight	
pastBackwardDown	



**Figure 4.14 : Procedure to Store Motion Vectors from Different Frames**

After detecting motion vectors, motion features are estimated using the method described in the previous section. Using the result, the frames are classified as very low, low, medium, high and very high activity frames. From this as with uncompressed domain, the very low, low and medium frames are further analyzed to identify regions that will have minimum impact of quality. For this purpose, the LAER is used.

#### **4.4. CHAPTER SUMMARY**

This chapter presented the algorithms that select frames and regions for embedding copyright watermarks in the uncompressed and compressed video files. For this purpose, a minimum motion activity frame selection algorithm and low complex region detection algorithm were developed. After selecting the embedding region, the next step is to embed the watermark. The algorithmic details for embedding watermark in an uncompressed and compressed video file are presented in the next chapter (Chapter 5, Design of Video Watermarking Techniques).