CHAPTER III

MOTION ESTIMATION FOR VIDEO CODING

In video compression, there are many unique concepts involved for achieving the efficient standard. A large number of interactions are needed between the processing steps involved in the video coding for successful compression either lossy or lossless compression [12-13]. In perspective of the video compression, the redundancy reduction seems to be the only viable solution. To attain it, motion estimation [21] is found to be the decisive chore. One of the challenging tasks in the video compression is the motion estimation. The pixel motion in the consecutive frame is considered demanding because of the advancements made in the video codec standards [31-40]. Moreover, both the hardware and software optimization of the motion estimation procedure is obligatory to cope with the computational complexity of the video compression standard. This chapter provides a vast introduction about the motion estimation procedure and its impact on the video coding standards. Moreover, the motion estimation process of H.264/AVC video coding standard and the elastic motion estimation methods are also discussed.

3.1 MOTION ESTIMATION

Video sequence is nothing but the sequence of image frames, the image sequence in most of the consecutive frames is same along the motion trajectories. This means that the content of the scenes in the consecutive frames is not varying. The motion estimation is performed along the trajectories of the motion to utilize the redundancy in the image sequence. Compression and noise reduction are performed only after performing the motion estimation process. The sensibility of the motion estimation algorithm depends on the place where it is handled. If the motion estimation is done in consecutive image frames without object movement which increases the complexity of the estimation procedure, then it is negligible. In the viewpoint of motion estimation, pixel sites of the moving object have to be determined for the motion estimation. Motion estimation determines the motion vectors in a frame with the help of the neighbouring frame. The variation in the pixel values embraces the motion estimation in the content of the image. The pixel variation is because of lighting condition, electronic noise or object motion.

The motion vector determined by the estimation procedure is connected to the whole image or specific parts of the image. The motion vectors have some components in not only the horizontal direction but also in the vertical direction, and these components have positive or negative values. Positive value resembles the motion towards the right or the downwards likewise negative value resembles the motion towards the left or the upwards. The motion estimation is possible in the accuracy of a pixel, but it is also possible to estimate the motion to an accuracy of less than a pixel by interpolating the values of the pixels. This kind of motion estimation is named as sub-pixel motion estimation or fractional motion estimation. Based upon this condition, the motion estimation technique is classified into two. They are global motion estimation technique and local motion estimation technique. Local motion estimation deals with the fortitude of motion objects within the video frame i.e. object movement is in any of the pixels. But in global motion estimation, the motion is applicable in every pixel of images. Some of the examples of the global motion are panning, tilting, and zoom in/out. The motion vectors are represented by the translation motion model, affine motion model, parabolic motion model, etc. The model representation is necessary to signify the motion displacements.

The video sequences contain the series of moving image. The image is subjected to the frame division. The images in the various frames are varying with the time interval and frequency. To perform a video compression the redundant data in the video sequences, the frames containing the same information need to be found. The neighboring frame of the frame has the same information content. The redundant frames usually contain the unwanted information contents such as noise, repeated data, etc. The motion trajectories defined between the each frame has same information content. For the image compression, the pixels in the images are redundant. Thus for the video compression, both the interframe and the intra-frame compression need to be done. This is obtained from the fact that, in a video sequence the image content does need to vary for each frame possibly. These image data of the video is almost same in the nearest frames. This means that there is no abundant changes occur between the frames. The motion estimation process finds the redundant frames along the video sequence. The noise reduction and compression in the video compression allow effective usage of the redundant frames in the video compression process. The noise factors in the video content may arise due to the elimination of the redundant frames.

The basic criteria for performing motion estimation are,

i. The minimum object size chosen for the motion estimation has a larger size than the actual block size of the image.
ii. At the varying pixel site, the horizontal and vertical components of the image have a maximum range value for the determined motion vector.

The above criteria for the motion estimation minimize the overall complexity and increase the efficiency of the algorithm.

To perform the video compression, the motion estimation needs to be done. The main classes present in the motion estimation are block-based method and gradient analysis method.

The motion estimation in the video compression system estimates the correct motion of the video sequence by analysing each pixel in the image block. The ME finds the motion equation for performing the motion estimation. The motion equation obtained has no solution for the single pixel site. After the motion estimation is done through the block-based techniques, and then it faces the aperture problem and the presence of the motion ambiguity at the edges of the large images.

Hence, the ME is done at the corner and the edges of the images suffer maximum challenges. When the motion estimation is done through the gradient based analysis, the technique faces the following disadvantages. They are,

i. As the incoming image sequence has no gradients, and then the motion estimation cannot be done.
ii. The motion of the video sequence should be in the smaller size.

When comparing the two techniques of the motion estimation, the gradient-based motion estimation has more advantages than the block based method. The gradient analysis method has low computational time than the block matching estimation method. The multi-resolution scheme, when incorporated with the above techniques, makes the motion estimation process to be easier. It also overcomes the difficulty of small motion assumption in the gradient analysis method.

The motion estimation is classified into different types based on its ability to solve the image sequence [20]. In some cases, the motion is estimated pixel wise, and in certain cases, the motion is estimated block wise. The motion estimation types are summarized as follows:

i. Optical flow equation method
ii. Pel-recursive method
iii. Block matching method

In the abridged categorization, optical flow method is a type of pixel-wise motion estimation, and the later two are related to the block-wise motion estimation.

(i) Optical flow equation method

The optical flow method is also called as the pixel-based motion estimation approach. In this method, the motion vector is calculated for each pixel in the image. The basic assumption made in this approach is constant brightness which means that the intensity of the pixel remains constant. The constancy of assumption augments the pixel matching in all the possible directions of the image. However, in the direction of the intensity gradient, there is no match is found in the pixel of the reference frame. To overcome such effects, constraints related to the smoothness displacement vector in the neighborhood are also introduced in the pixel based motion estimation.

(ii) Pel-recursive method

The pel-recursive method is a type of block-based motion estimation. The motion vector is determined in this method by estimating the relative phase shift between the partitioned image blocks using the normalized cross-correlation function. Over a series of iterations, the pel-recursive method estimates the motion vector by making small adjustments. This type of method is unfeasible for real-time application because of the high computational overhead and phase ambiguity.

The motion estimation using the pel-recursive method is done as follows. Consider the input image sequence with the image function as m_i. The motion estimation of the image 'm' can be obtained through the update process. The updated image 'u_i' can be given by the Equation (3.1),

$$m = m_i + u_i \tag{3.1}$$

The above-updated image function can be expanded using the Taylor series and it is represented in Equations (3.2) and (3.3),

$$I_n(x) = I_{n-1}(x + m_i + u_i) \quad (3.2)$$

$$I_n(x) = I_{n-1}(x + m_i) + u_i^T \nabla I_{n-1}(x + m_i) + e_{n-1}(x + m_i) \quad (3.3)$$

Equation (3.3) gives the difference in the motion estimation of the image frames. The update function estimates the displacement in the motion of the image video sequences. The motion estimation process for each and every frame is done from the estimated equation.

(iii) Block matching method

Block matching based motion estimation is the convincing motion estimation procedure among the available motion estimation techniques [21] and it is very robust. The block matching motion estimation procedure is performed by considering two assumptions:

(a) The minimum size of the object is larger than the size of the chosen block (4×4, $8 \times$, or 16×16).

(b) The range of the horizontal components and vertical components of the motion vector is predefined.

The motion estimation of the complete block is known as block matching method. Because of this adaptability, in most of the video codec's, block matching method is utilized as motion estimation algorithm.

In block matching algorithm, the image is divided into the block (macro blocks). The partition yields blocks of the similar size; $N \times N$. Every block is examined for the motion estimation in turn and a motion vector is allocated to every block. The motion vector for each block of the image is chosen by matching the blocks with another block having same size same size (reference block). For series of iterations, the location of the block in the current frame is defined by some search patterns of the previous frame. The motion vector is chosen in such a way that it minimizes the block distortion measure [22]. In BMA, the motion vector of the whole block of pixels is calculated as a substitute of individual pixels.

The number of the motion vectors used in the BMA reduces the computational time of the algorithm. Various search algorithms were used before the block matching process. The Three-step search performs the hierarchical search strategy using the eight motion vectors. The algorithm reduces the distance between the blocks to be evaluated at the consecutive steps. The next the block for the matching is given by placing the search point at the center. The motion estimation gets further refined through the processing of the various block matching.

The noise factor acts as a major challenge in the BMA algorithm. The noise present at the corner of the edges of the frame reduces the tendency to match the block. The BMA performs as a robust estimator for the motion estimation process through the elimination of the noise parameters. The block matching is done through the block comparison. The noise in the frame dominates the matching process if the textual information is not present in it. The noise produces the spurious motion estimation. The block matching process for the motion estimation can be done by estimating the best match found (E_m) and the match with no motion (E_0). The block matching found between the blocks are different from each other than the motion estimation is done otherwise it is considered that the block contains no motion. The threshold for defining the motion estimation is defined by the Equation (3.4),

$$r_b = E_o E_m \quad (3.4)$$

The motion estimation through the block matching algorithm has different errors due to the presence of unwanted information in the frame.

An example for the block based motion estimation is shown in Figure 3.1.

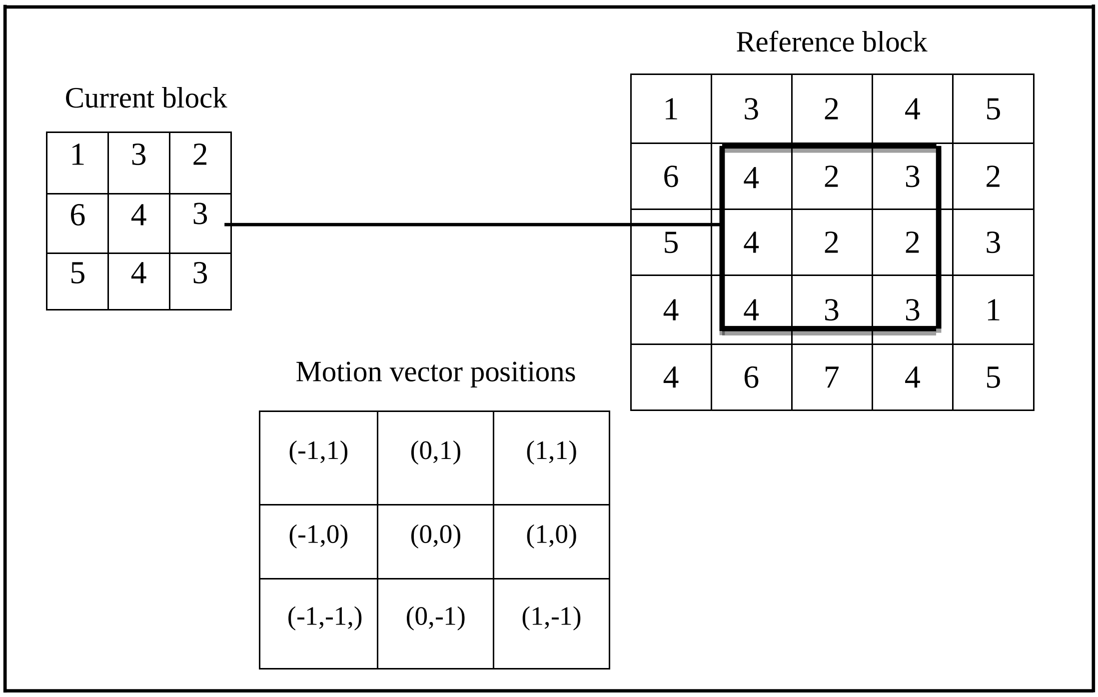

Figure 3.1: Example for Block-Based Motion Estimation

The matching process in the block matching algorithm is limited to the search window. The size of the search window is smaller than the size of the image frame.

3.1.1 Block matching criteria

The block matching techniques allow the video compression through the coding of the image frames in the video content. The video coding standards, namely H.264/AVC and HEVC use the block matching criteria for the video compression. The block matching process has low complexity. The block matching process reduces the dissimilarity measure between the video frames. The motion equation contains the various motion vectors obtained through the block matching process. The motion vector allows for the complete solution of the motion equation and thus the video compression is more accurate. The block matching process is considered to be more robust than the other techniques such as global optimization and the pel-recursive techniques.

Consider an image of size $\mathrm{N} \times \mathrm{M}$ for the block matching. The block $B_{x,y}$ is defined as the $\mathrm{X} \times \mathrm{Y}$ and has the indices of the value (x, y). The defined block is represented as follows in Equation (3.5),

$$B_{x,y} = \{x, X+1, \ldots, y+Y-1\} \tag{3.5}$$

The values chosen for the block sizes may be 4, 8 or 16. But the block size of the HEVC standard is very large. The image frames in the blocks have all the pixels from which the required motion vector is calculated.

The motion vectors are presented within the candidate vectors, and they get selected from a suitable set. The set from which the motion vectors are obtained is represented as a search window. The search window has a vast area and the similar blocks in the candidate vector set get selected through the search process. The search window has a rectangular area which is mostly placed in the center of the block. The size and shape of the search are a larger impact while estimating the motion. Hence the choice of the search algorithm and the search area is must be done with good care.

The block matching algorithms for estimating the motion have different characteristics based on the frame size of the image and the rate of the video sequences. The block matching algorithm differ by the following factors as follows,

i. The search strategy is used to determine the motion vectors in the block. The defined search window is subjected for the scanning process to find the most suitable vector for the motion estimation equation.

ii. The various matching criteria used for determining the block function and the candidate vector.
iii. The size and shape of the blocks used also impacts the searching process.

The best matching block in the reference frame is matched to a current frame using a suitable matching criterion. The matching criterion is also called as the distortion measure. There are different types of matching measures. Moreover, there are several criterions for estimating the motion but there is no uniqueness. The criterion selection is based on the task or the application-oriented environment upon which the motion estimation is performed. Some of the block matching measures [24] are shown in Equations (3.6), (3.7), (3.8) and (3.9).

Means Squared Error:

$$MSE = \frac{1}{N^2} \sum_{u=1}^{N-1} \sum_{v=0}^{N-1} \left(C_{u,v} - R_{u,v} \right) \tag{3.6}$$

Mean Absolute Difference:

$$MAD = \frac{1}{N^2} \sum_{u=1}^{N-1} \sum_{v=0}^{N-1} \left| C_{u,v} - R_{u,v} \right| \tag{3.7}$$

Sum of Absolute Difference:

$$SAD = \sum_{u=1}^{N-1} \sum_{v=0}^{N-1} \left| C_{u,v} - R_{u,v} \right| \tag{3.8}$$

Rate-Distortion Optimization:

$$D = \sum_{u=1}^{N-1} \sum_{v=0}^{N-1} \left(C_{u,v} - R_{u,v} \right)^2 \tag{3.9}$$

3.1.2 Search algorithms

The matching block in the motion estimation procedure is explored using different searching techniques [25-28]. When estimating the motion, the minimum motion vectors are investigated using the adopted searching algorithm. There exist different kinds of algorithms for the searching task; each of the technique gives a different motion vector.

Search algorithms in the motion estimation are categorized as follows:

The fixed set of search pattern:

In this type of searching technique, search location is selected depends on the adopted search algorithm. Many of the search algorithms with different search patterns are utilized in this searching, namely full search, three step search, four step search, and diamond search [55, 58, 59, and 71].

Full search:

The full search algorithm finds the motion through the complete analysis of the grid area is shown in Figure 3.2. The search area taken for the process refers the whole reference image from the video sequence. The block $B_{x,y}$ is taken from the image of size $N \times M$. The algorithm compares the block $B_{x,y}$ with the reference image to acquire the best candidate motion vector. The similarity measures for the every comparison of the candidate vector need to be noted. The full search does not consider every pixel of the image frame. The full search process uses the rectangular search space area. The search area size is given by (x, y). The search window W in the algorithm is constructed using the set of vectors. The search window for the full search algorithm is given in Equation (3.10).

$$W = \{-P, \ldots, -1, 0, 1, \ldots P\} \times \{-Q, \ldots, -1, 0, 1, \ldots Q\} \tag{3.10}$$

The terms P and Q defines the vertical and horizontal search area of the search space. The search window has the varying values of the P and Q. In the real-time video contents, the horizontal part of the video varies with the higher rate than the vertical part. For the simplification process, the value of the P and Q is taken to be similar. The width of the rectangular search window is given as follows,

$$n = 2P + 1 \tag{3.11}$$

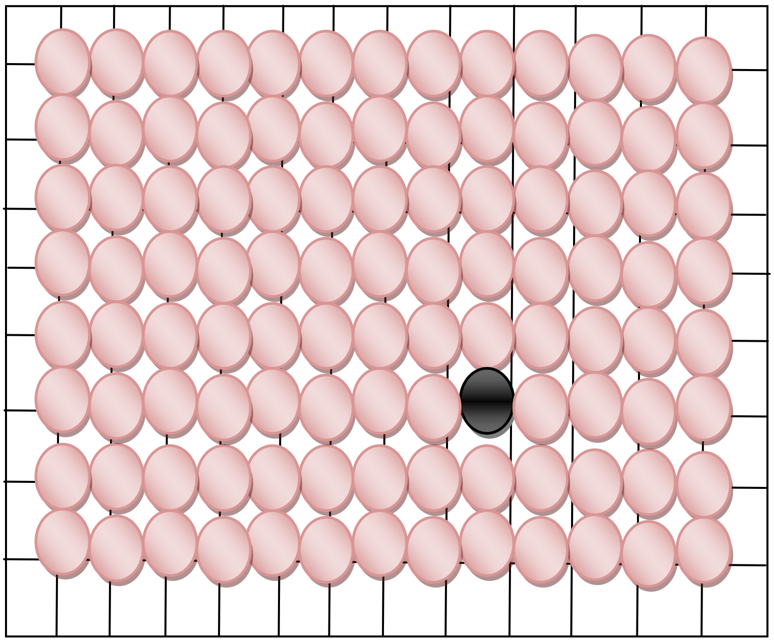

Figure 3.2: Full Search Algorithm

Fast search:

The full search algorithm implemented has an increased computational complexity. The full search algorithm has the high computational complexity which is reduced by the suboptimal techniques. The fast search algorithm selects a subset of the search window for finding the candidate vectors. The selection of the subset has lesser computational complexity than the full search algorithm. The processes in the fast search process are repeated till the best motion vector is determined. The constructed search window is made up of the large motion vectors in fewer numbers. The initial candidate vector was chosen for the search process is (i_0,j_0).

The search process gets upgraded for every step. The search window W_k selects the candidate vector (i_k,j_k) at the k^{th} step. The search window gets altered for every step. The best candidate vector is obtained at the search window's center point with the likely scaling process. The fast search algorithm requires a stop function for the stopping of the process at the right time. The stop function defines the total number of iterations and the vectors in the search window.

Three step search:

The Three Step Search is one of the suboptimal search processes shown in Figure 3.3 and Figure 3.4. The algorithm finds the best optimum candidate vector in the three iterative steps for performing the video compression process. The TSS approach has a uniform complexity throughout the process. The complexity of the algorithm can be reduced with the compromise in the motion vector. The TSS approach has similarity with the 2D- logarithmic approach with the following changes,

i. The search window of the TSS model has nine elements which are formed during the block matching process.
ii. The search window is given in Equation (3.12) and (3.13),

$$W_{k+1} = \{(i_k, j_k) + (i, j) \mid (i, j) \in \nabla W_k\} \tag{3.12}$$

$$\nabla W_k = \{-2^k, 0, 2^k\}^2 \tag{3.13}$$

iii. The search radius r_k gets changed as half for the consecutive steps.

If the value of the window is $\nabla W_1 = \{-4,0,4\}^2$ in the first iteration, then the algorithms get stops in three steps. Otherwise, the algorithm complexity gets increased. The TSS algorithm requires only 25 computations for calculating the displacement changes of the ±7 pixels in both directions.

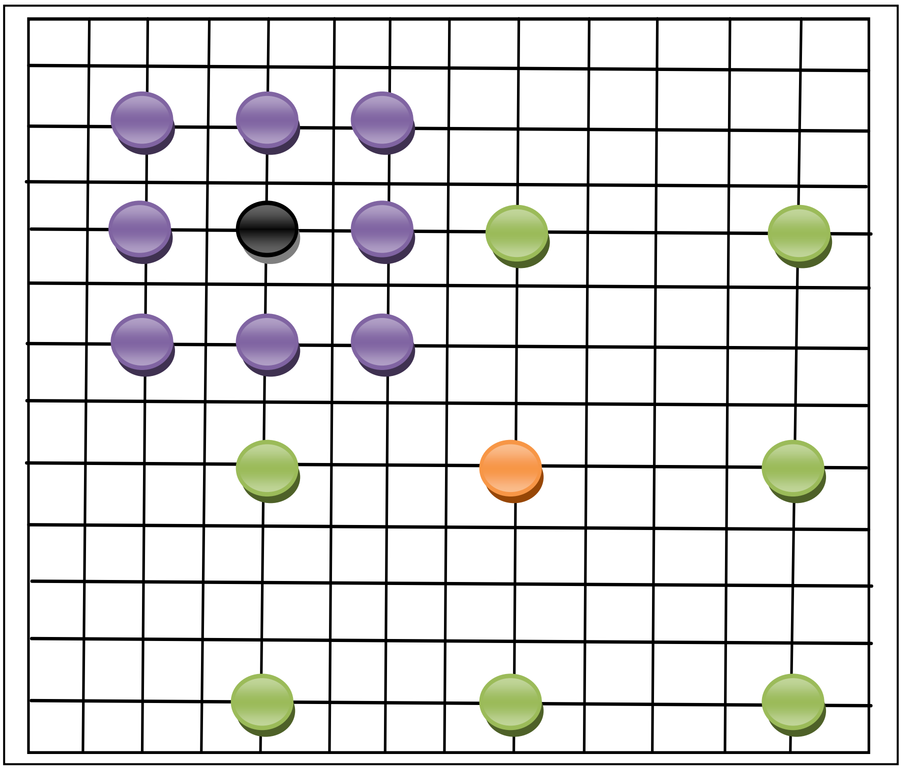

Figure 3.3: Three Step Approach (Step two (after initial search))

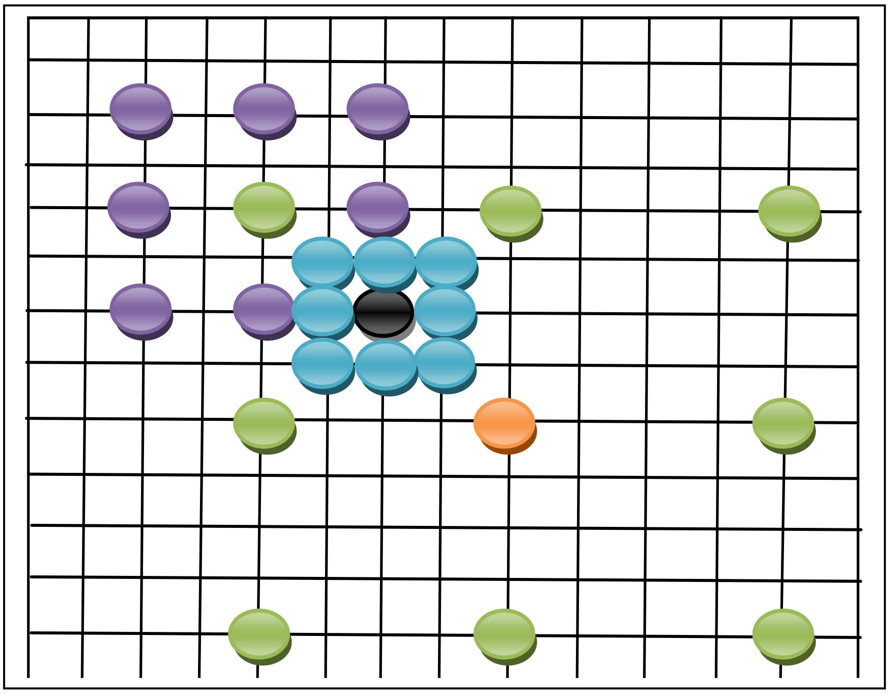

Figure 3.4: Three Step Approach (Step three)

Four step search:

The three-step search algorithm is modified to obtain the four-step search algorithm is shown in Figure 3.5. The four-step search algorithm is as similar as TSS algorithm but it utilizes the little initial step size. This increases the iteration step to four in need to find the candidate vector.

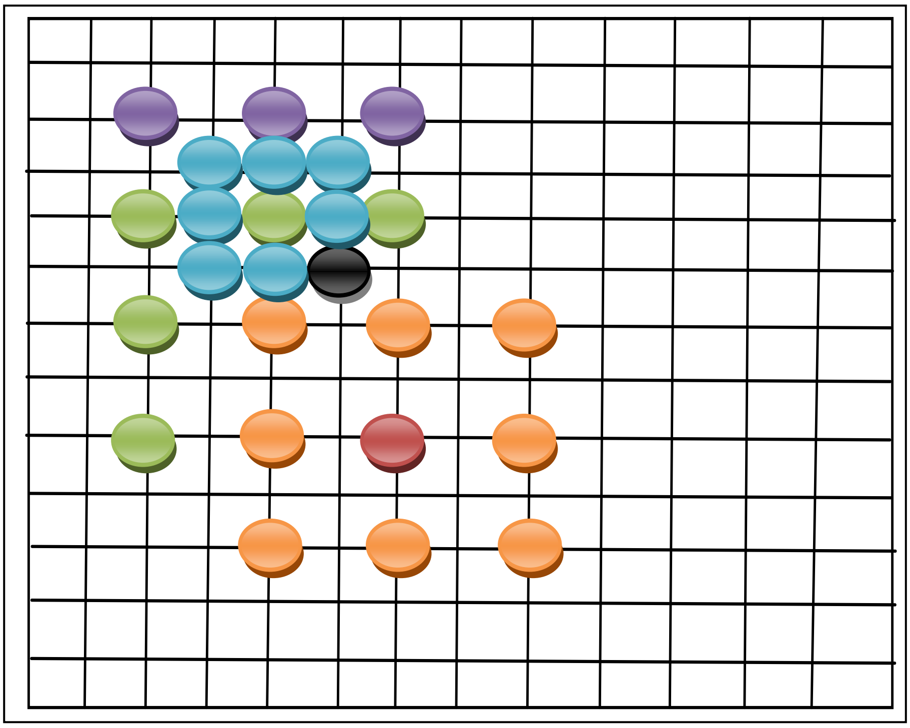

Figure 3.5: Four Step Approach

Diamond search:

The suboptimal search algorithms such as TSS and FSS have a limited accuracy when the search area gets expanded to the value of 15 x 15. Hence, for the real time practical applications these algorithms produce an inefficient compression. To avoid these disadvantages, the DS algorithm can be used. The DS algorithm finds application in the video encoders. The DS algorithm has two search steps for finding the motion estimation of the video content is shown in Figure 3.6. The DS algorithm uses a diamond-shaped search for the motion estimation. In the primary step, an image with the diamond shape is considered. The best candidate vector may or may not be present at the search area's center. I need to find the candidate vector the diamond-shaped search area gets shrink at every step. The main advantage of the DS algorithm is that the entire diamond search area need not be checked since the consecutive diamond search areas are overlapped with each other. Thus the subset within the search area provides the best candidate vector for the motion estimation. The DS algorithm does not stop to find the vector until the best candidate vector is found. The reference image used for finding the motion vectors gets moved to the search area’s center point to find the candidate vector. The advantages of the DS algorithm are stated as follows,

i. The DS algorithm has a medium spaced search area. Hence the complexity is greatly reduced than the TSS and the FSS algorithms.

ii. The search path of the algorithm is diagonal, and hence search process has a smaller time than other algorithms.
iii. The search area is not limited.

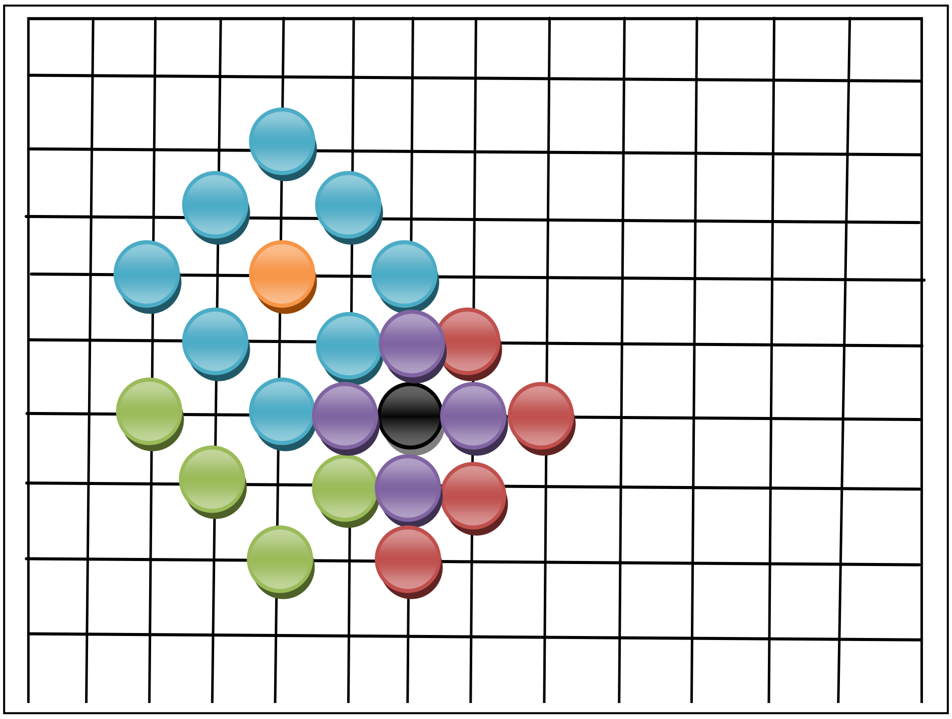

Figure 3.6: Diamond Search

Hexagon search:

The use of the DS algorithm for the motion estimation has greatly reduced the complexity of the search process. But the disadvantage of the DS algorithm is that the DS moves at a speed of the 2 pixels for every step along the search space area. This may lead to the increase in the iteration for the search process. This can be avoided with the use of the hexagonal search algorithm is shown in Figure 3.7. In the hexagonal search algorithm, the search space determines the best candidate vector. The search space is neither large nor small and is chosen accordingly with the video pattern size. The hexagonal search pattern has mores similarity with the DS algorithm. But it differs by the fact that, it does not skip 2 pixels during the search. The size of the hexagonal pattern alters for each and every iteration process. The hexagonal search approach has the following advantages than the DS algorithm. They are,

i. The hexagonal search pattern tests only three points for every iteration process. This reduces the complexity.
ii. The search process is not diagonal.
iii. The computational complexity is 40 % less than the DS algorithm.

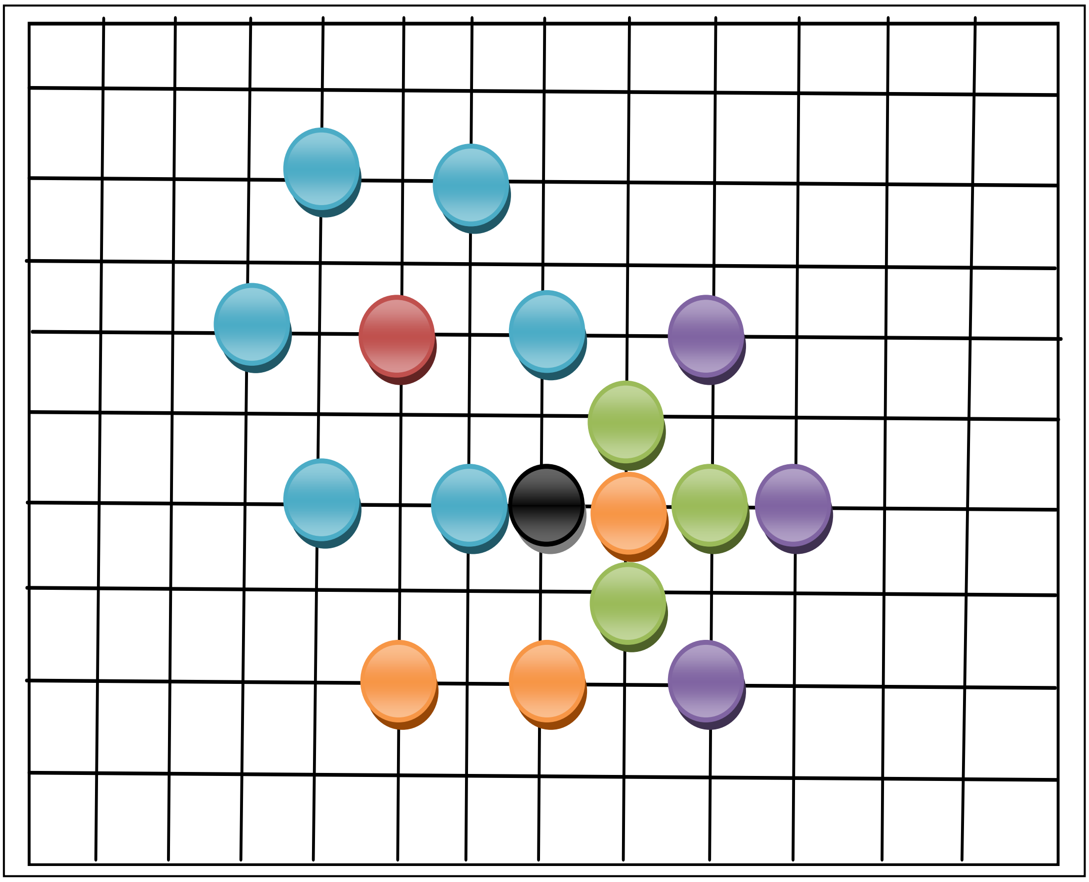
Figure 3.7: Hexagon Search Approach

Predictive search:

In the prediction search technique, the target motion vector is predicted depends on the correlation between the current block and the reference block. Based upon the prediction, the initial motion is estimated.

Sub-sampled pixel:

Previous three search algorithms aim to determine the motion vector by reducing the search locations by the proper search pattern. But this method, as the name suggests, reduces the number of pixels to compute the error among the current block and the reference macroblock depending on this, the vector is estimated.

Fast full search:

It is one of the stable searching techniques which aim to improve the searching speed without any compromise in the quality. The significant advantage of this type of searching algorithm is the capability to check whether the candidate block is feasible enough for matching procedure. Thereby, large numbers of unnecessary computations are avoided.

3.1.3 Motion estimation, compensation, and prediction for video coding

The fundamental process in video compression and other video processing is the motion estimation. The motion estimation is used to extract the motion information from the video. Motion in the video sequence is expressed as motion vectors. It indicates the displacement of the pixel from the current location to the other location because of motion of the objects in the image (content). The estimated motion information is utilized in video coding to determine the best matching block in the reference frame for constructing the temporally interpolated frames. This procedure is called the motion compensation [19 and 20].

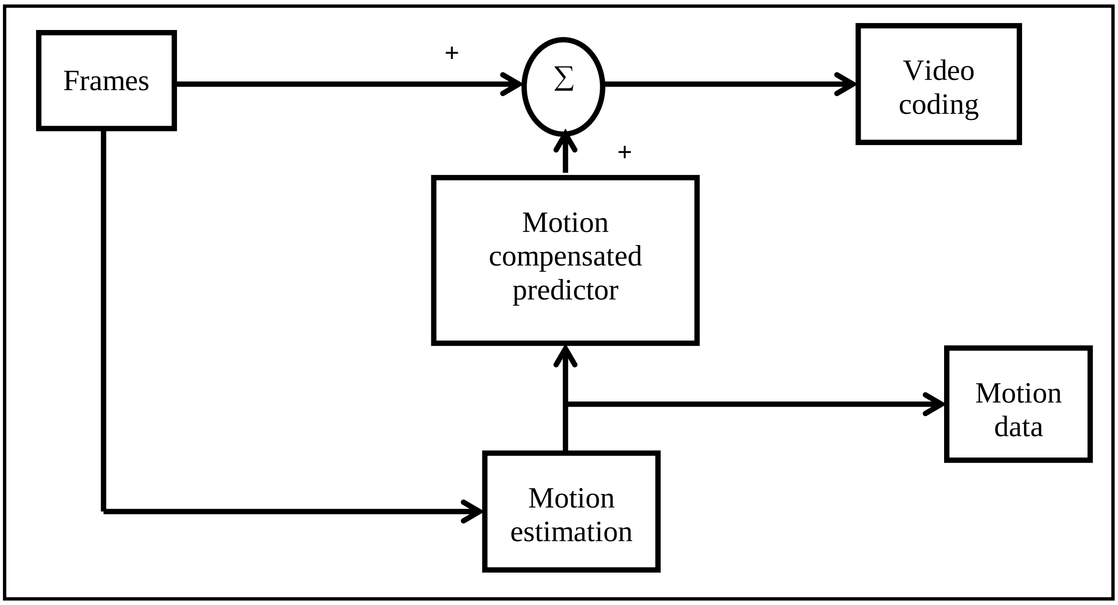

Figure 3.8: Motion Estimation and Compensation Model for Video Coding

The bit rate of the video signals in the modern video technology is reduced by the motion compensation prediction. The motion compensation prediction utilizes the motion estimation to estimate the motion vector which is utilized for the prediction process. The impact of the motion estimation in video coding standard is illustrated in Figure 3.8. From Figure 3.8, the fact is evident that for the video coding, estimating the motion is the possible solution to remove the redundancy existing in each frame.

Likewise, it is utilized in different applications, such as motion tracking, video stabilization, and motion compensated de-interlacing. The motion estimation is adapted with the most video coding standards to remove the temporal redundancy between the successive frames. However, the intense computational part of motion estimation leads to the following problems such as: searching time, video quality, and matching criterion.

3.2 H.264/AVC MOTION ESTIMATION FOR VIDEO CODING

H.264/AVC is one of the recent video coding standards [40]. The utmost intent of the H.264/AVC video codec is providing an improved compression performance without any compromise in the video quality for conversational and non-conversational applications. The motion estimation procedure of the H.264/AVC video codec is different from the motion estimation process of the proceeding video codec standards.

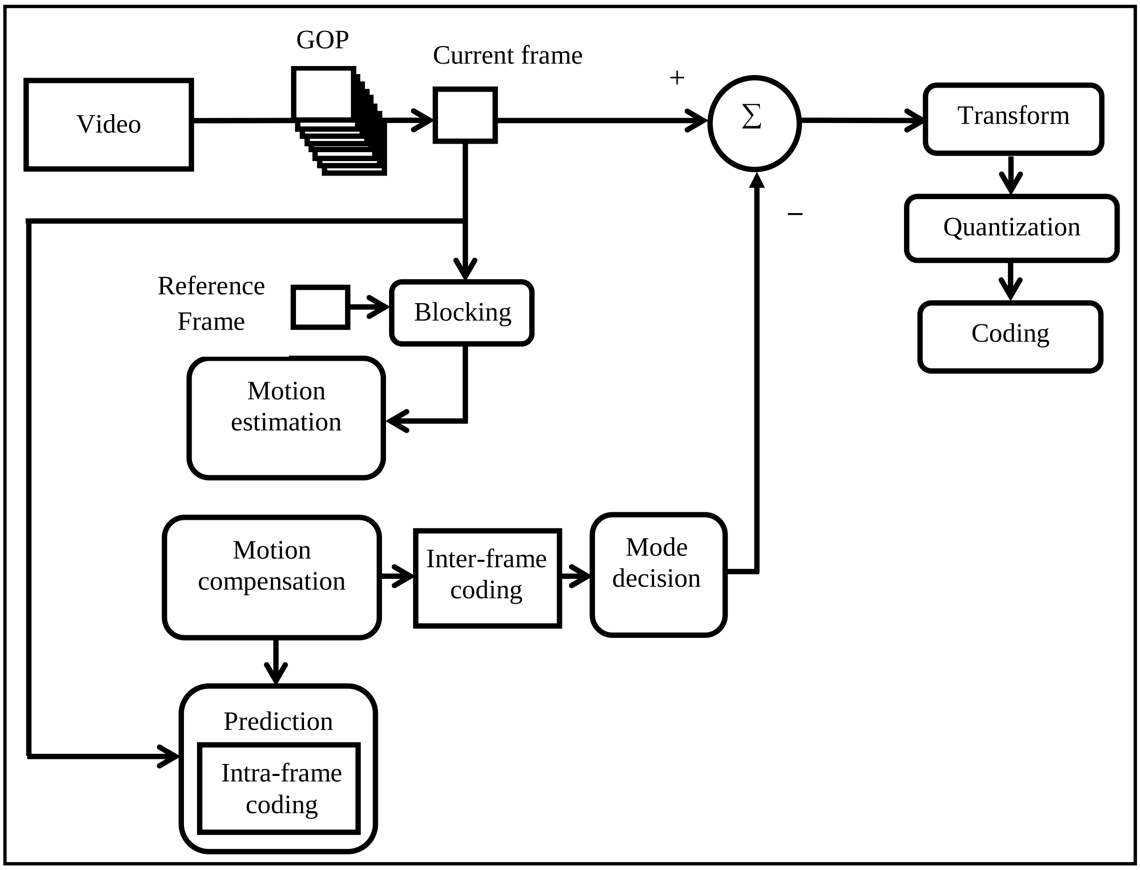

Figure 3.9: H.264/AVC Encoder

Block-based motion estimation scheme is utilized in the H.264/AVC since the superiority of the compression performance is contributed by the motion estimation. Figure 3.9 shows the encoder of H.264/AVC video codec standard. In the motion estimation procedure of H.264/AVC, multiple reference frames are considered. The description of the motion estimation in H.264/AVC rises with the block partitioning, where the macroblocks of the images are divided into smaller blocks. In the encoder, by the combined action of the motion estimation and motion compensation prediction, Intra and interframe coding are performed in the H.264/AVC, which reduces the spatial and temporal redundancy of the image frames. In H.264/AVC, the block-based motion compensation is performed in two parts: Block matching motion estimation and Motion compensation. The motion estimation finds the motion vector of the current block. The motion vector is used in the motion compensation to calculate the residual data. The redial data together with the motion vector is encoded.

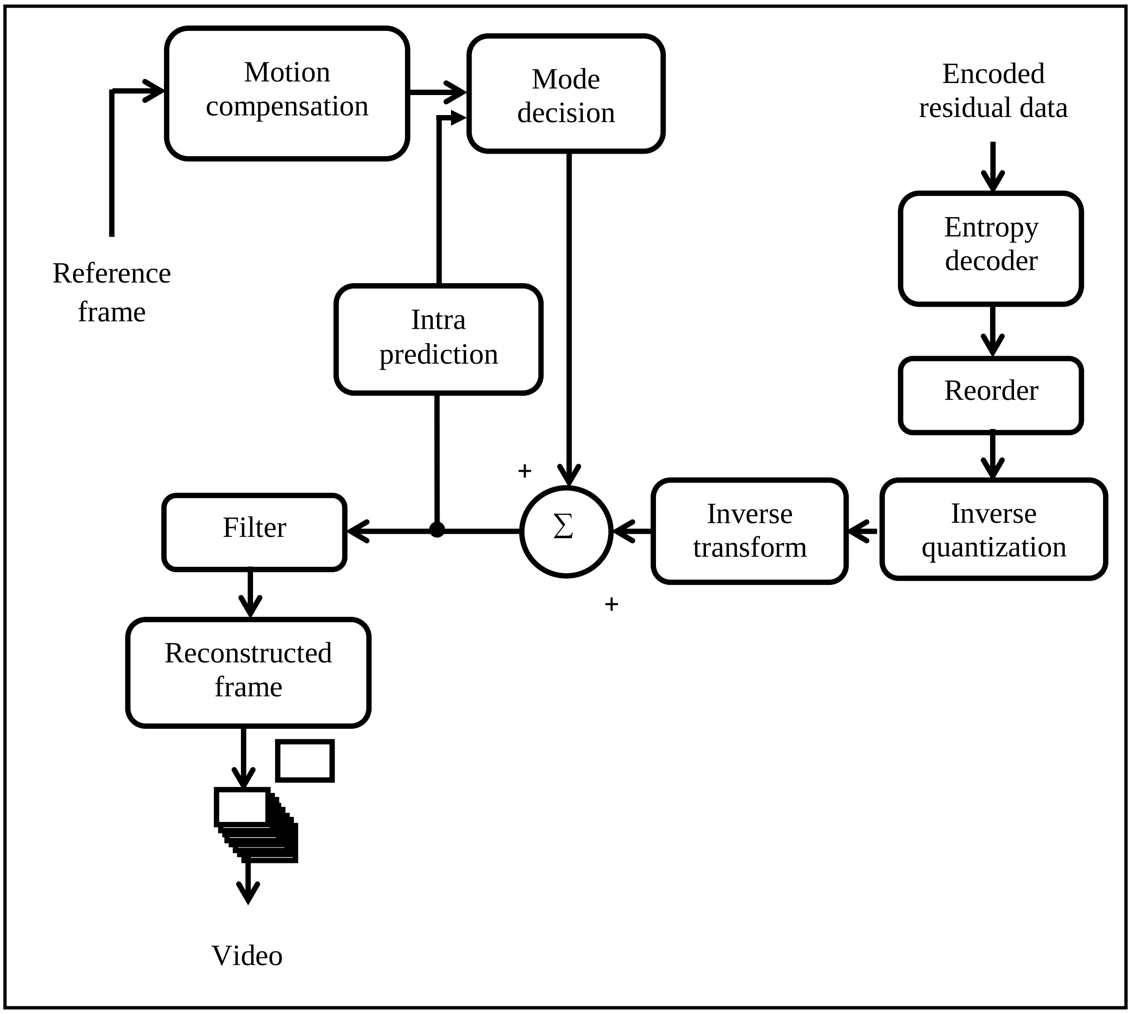

Figure 3.10: H.264/AVC Decoder

Figure 3.10 shows the H.264 decoder. The encoded residual data with motion vector at the encoder side is decoded by the decoder to reconstruct the frame. The frame reconstruction is performed using the motion compensation prediction. The motion compensated data from the reference frame is subtracted from the residual data to obtain the reconstructed frame. The reconstruction procedure is reiterated collecting the consecutive frames. The collective frames are reappointed into the video sequence. The adaptation of the efficient motion estimation algorithm increases the efficiency of the video coding standard.

From the encoder and decoder of H.264/AVC, the influence of the motion estimation, compensation and prediction are perceptible. The significant features of the motion estimation algorithm of the H.264/AVC standard are: variable block size motion estimation and quarter pixel prediction.

3.3 ELASTIC MOTION ESTIMATION FOR VIDEO CODING

In this section, the motion estimation procedure of elastic motion model is depicted. In H.264/AVC, the translational motion vector is used for the representation. But the translation motion vector representation is not feasible enough to capture the complex motions in the image frames. Moreover, the discontinuity perseverance, block-based motion model of the H.264/AVC is left overlooked. Elastic motion model is a video coding standard which overcomes these issues of H.264/AVC [79]. The change is initiated in the block partitioning of the elastic model itself. Multilevel tree block partitioning is adopted in elastic motion model which selects the blocks with larger level as well as a smaller level for block matching covering the complex motions. The frame to be coded is divided into blocks of equal sizes as shown in Figure 3.11. Figure 3.11 illustrates the marco block segmentation.

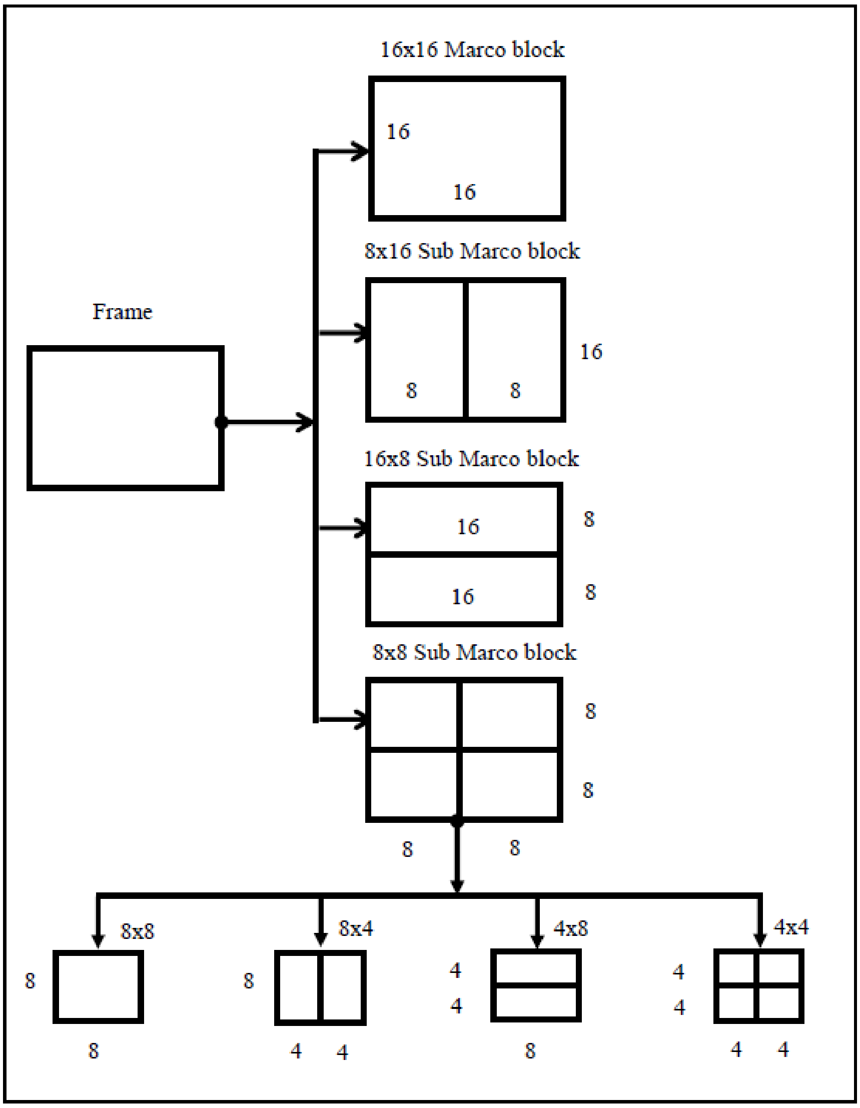

Figure 3.11: Marco Block Segmentation

The encoder and decoder structure of the elastic motion model is same as that of the H.264/AVC video codec standard. The variation is in the motion estimation process. Here, elastic image registration technique is adopted for the motion estimation. The motion estimation procedure of elastic motion model is depicted in Figure 3.12. In addition to the translation motion vectors determined by the video codec (as in H.264/AVC), motion data (parameters) that describes the non-translational motion model is also described in the elastic motion estimation.

The motion parameter is determined based on the elastic image registration technique. Primarily, the motion model is defined using the registration technique with discrete cosine to capture the elastic motions. After modeling, the motion parameter which enables the best prediction of a block is estimated by iteration of first order taylor approximation criterion.

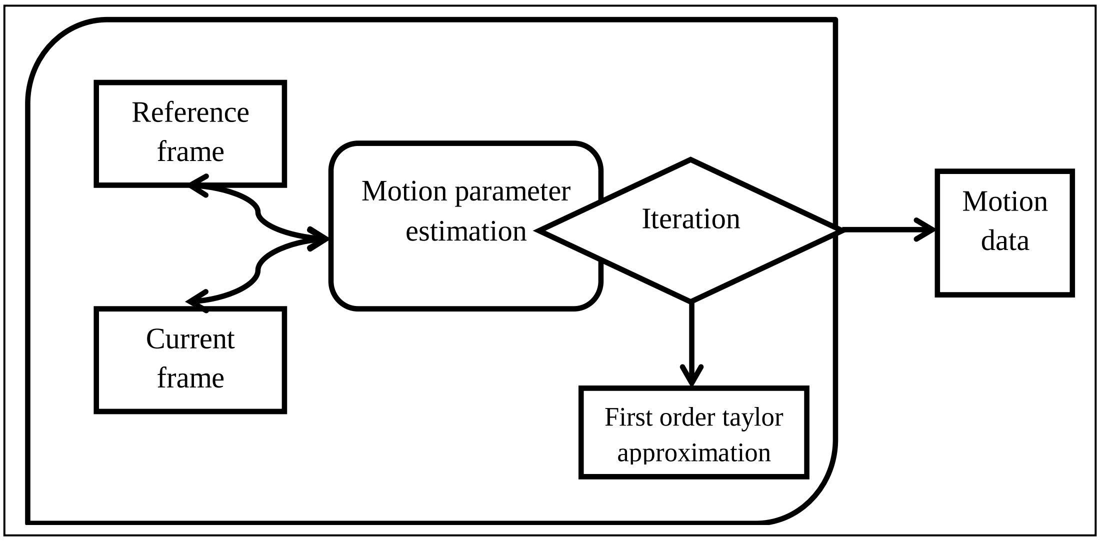

Figure 3.12: Motion Estimation in Elastic Model

In the viewpoint of elastic video coding, the model selection selects the motion vector with less cost (translation motion vectors or non-translational motion vectors).

3.4 SHORTCOMINGS OF H.264 AND ELASTIC MOTION ESTIMATION METHODS

In both H.264 and elastic motion estimation methods are well thought out as the efficient video compression standards [40, 79], some inadequacies are present. The utmost intent of the motion estimation in video coding is to reduce the redundancy in frames with proper block matching algorithm. However, in H.264 video codec and elastic motion model, the problem ascends because of the adopted motion estimation algorithm. The shortcomings of the H.264 motion estimation algorithm are,

i. Search classification problem: Refining of the search classification affects the motion vector vigorously.
ii. Block partitioning problem: Small level partitioning-unable to handle complex motions.
iii. Searching time problem: The search time for motion vector estimation is large because of the wide range of search window and iterative searching procedure.
iv. Difficulty in preserving tricky motions: Motion estimation in the image containing, zooming, rotation, fast moving objects, etc. is difficult.
v. Search point problem: Increased number of search point increases the complexity of motion estimation procedure.
vi. Computational overhead: The computational time of the H.264/AVC video coding standard is increased because of the increase in searching time for determination of the matching blocks.

Most of the disadvantages of H.264/AVC video codec standard are overcome by the elastic motion model which combined uses the translation motion model and elastic motion model for the coding procedure. However, it also embraces some of the disadvantages such as:

i. The trade-off function: Evaluation function determining the matched search points and blocks are not up to the mark in the elastic motion model.
ii. Block distortion measure: The block matching measure considered in elastic motion model is completely predefined and affects the efficiency of the block matching search algorithm
iii. Search algorithm: Search algorithm adapted in elastic motion model preserve the motion discontinuities with the increased number of searching points for the best matching.
iv. Computational complexity: The computational overhead of the elastic motion model increases because of the inverse compositional gradient descent optimization [18] used to compute the elastic motion parameters.
v. Reference Frame: Large numbers of reference frames are required in the matched block estimation of the elastic motion model.

3.5 SUMMARY

In this chapter, a brief introduction to the motion estimation and its impact on the video coding/compression is deliberated. In the main, block-based motion estimation is found to be the better option in the video compression standards other than the optical flow or pel-recursive motion estimation methods. Firstly, the motion estimation procedure of the H.264/AVC video codec is discussed. The motion estimation relies on the best optional motion compensation for the video coding. The motion estimation algorithm in the H.264/AVC video standard is found to be more advantageous. Besides, most of the shortcomings present in the H.264/AVC video compression standard are because of the motion estimation algorithm adopted in it. Secondly, the motion estimation procedure of the elastic motion model is discussed. In the elastic model, the discontinuities in the image content of the video to be compressed are well preserved by the adopted motion estimation algorithm. Comparing H.264/AVC and elastic motion estimation procedure, the elastic motion estimation is found to be highly efficient. On the other hand, the search algorithm in both the compression standards is not adequate to capture the complex motion with the reduced number of search points and reference frames, the better trade off and matching criteria function without any compromise in the video quality. The research direction in the way of new search algorithm for the block matching with the reduced computational complexity is obligatory in the motion estimation algorithm of the video compression standards to improve the visual quality and compression performance.