New Fast Block Matching Search Algorithm For Image Video Compression

م.م. وداد عبد الخضر ناصر
كلية التربية /قسم الحاسبات

Abstract :

Successive video frames may contain the same objects (still or moving). Motion estimation examines the movement of objects in an image sequence to try to obtain vectors representing the estimated motion. Motion compensation uses the knowledge of object motion so obtained to achieve data compression.

Many block matching search algorithms have been developed to improve searching(by reduce the number of search position) and matching phases.

The current research work aims to implement the H.263 video compression by developing all the required programs .Through the implementation work of H.263 most of the well-known motion search method were tested and their performance was investigated .

In this paper, we propose a new fast block matching search algorithm for motion estimation to handle the time delay associated with the searching methods to speed up the image video compression . However the proposed algorithm did not effect the compression efficiency and image quality. In the proposed algorithm, we assume that a video frame is divided into macroblocks with a size of 16x16 integer pixels.

The Mean –Absolute Difference(MAD) is used a s the cost measure to select the best matching block in motion estimation.

In addition, the use of half-pixel instead of using integer –pixel was investigated in order to maximize the compression performance and compression ratio

1-Introduction

Video compression addresses the problem of reducing the amount of data required to represent a digital image. The objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form . Form a mathematical point of view this amounts to transforming a 2D-pixel array into a statistically uncorrected data set .The transforming is applied prior to storage or transmission of the image . At some late time , the compressed image is decompressed to reconstruct the original image or an approximation to it [1] .

All compression system requre two alogrithms : one for compressing the data at the source , and another for decompressing it at the destination . In the literature these alogrithms are referred to as encoding and decoding algorithm. The general procedure for compression image information is shown in figure (1).

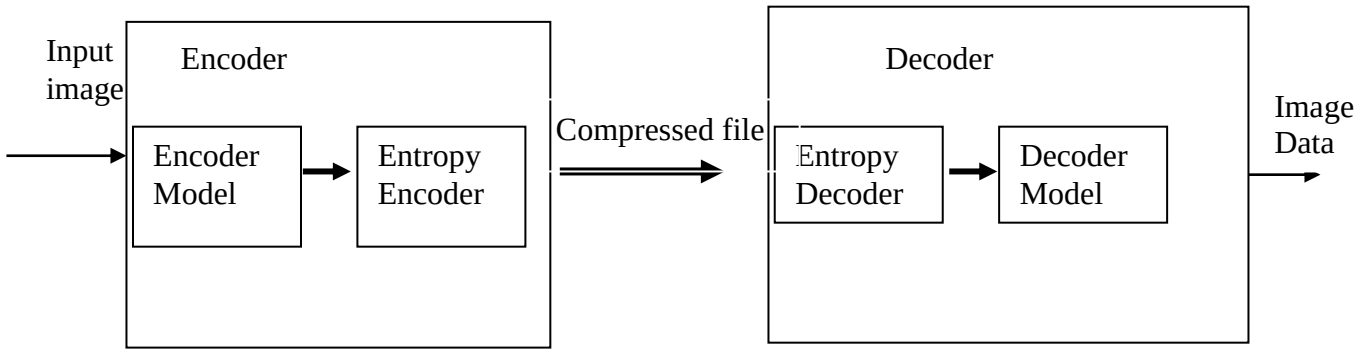


Fig.(1): Image compression model [1].

The encoder produces symbol that represent the information in the original image .

These symbols are then entropy encoded (for example using Huffman encoding) to code them as efficiently as possible , the decoder carries out the reverse procedure to recreate copy of the original image[2].

There are many video compression standards, such as CCITT H.261 and MPEG , which use block-based motion estimation and compensation to reduce the temporal redundancy in video sequences. A natural way of exploiting the redundancy between successive frames consists of using frame *n*- 1 to predict the next frame, *n*. Frames *n* and *n*- 1 are often referred to as the present and previous frames. In block-motion technique, a present frame of a sequence is divided into rectangular or square blocks of pixels. For each block

of a present frame, we look for a block of pixels in a previous frame that is the closest to it. The relative positions of the two blocks define a motion vector associated with the present block. The collection of all motion vectors defines a motion field. Many block-matching algorithm for motion estimation have been developed and evaluated in the literature. Most of them are concerned with how to define a search area where a candidate block is looked for.
Compression algorithm development stated with certain applications to two – dimensional (2D) still images . Because video and television signals consist of consecutive frames of image data , the development of compression methods for 2D still data is of paramount importance . After these methods are developed , they are extended to video(motion imaging) [3].

2- The Discrete Cosine Transform (DCT)

The discrete cosine transform(DCT)is using for image coding [8].For most continuos-tone photographic images, the DCT provide energy compaction that is close to the optimum. Anumber of fast alogrithmexist for calculating the DCT of a block of pixel. These factors have led to the wide spread use of the DCT for image andvideo compression system[9].
The DCT operation transforms a block of pixels into the set of DCT coefficients that represent the block in the spatial frequency domain . The two –dimensional DCT of an

8x8 block of pixel values is described by the following formula , where the pixel values are represented as f(x,y) and the transform coefficients as F(u,v).

$$F(u,v)= \frac{C(u)}{2} \cdot \frac{C(v)}{2} \sum \sum f(x,y)\cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16} \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

where

$$C(u)\,C(v)= \begin{cases} \frac{1}{} & \text{for u or v } >0 \\ 1 & \text{for u or v } >0 \end{cases}$$

Then , the Inverse Discrete Cosine Transform (IDCT)is implemented on dequantized coefficients to convert the image from frequency domain into spatial domain . The IDCT equation is defined as :

$$f(x,y)=\frac{1}{4}\left[\sum\sum C(u)\,C(v)\,F(u,v)\cos\frac{(2x+1)u\pi}{}\cos\frac{(2y+1)v\pi}{}\right]. \quad (2)$$

where

$$C(u)\,C(v)=\begin{cases}\frac{1}{} & \text{for } u \text{ or } v=0\\ 1 & \text{for } u \text{ or } v>0\end{cases}$$

3-Motion Estimation And Compensation

Motion estimation and compensation are widely used in various video coding standards, such as H.261, MPEG1 and MPEG2, to achieve high compression ratio by reducing the temporal redundancy as possible to reduce the size of the data required in digital video storage and transmission.

Motion estimation (ME) is a process to estimate the pixels of the current frame from reference frame(s). Block matching motion estimation or block matching algorithm (BMA), which is temporal redundancy removal technique between 2 or more successive frames, is an integral part for most of the motion-compensated video coding standards . In the block matching algorithm, the encoder first divides the current image frame into many fixed-size blocks, or so-called macroblocks (MB). Block-matching method is to seek for the best-matched block from the previous frame, usually the first single frame, within a fixed-sized of search window (*w*). Based on a block distortion measure (BDM) or other matching criteria, the displacement of the best-matched block will be described as the motion vector (MV) to the block in the current frame. The motion vector is calculated by taking the difference between the current location of the block minus the location of the match . The best match is usually evaluated by a cost function such as Mean – Absolute Difference(MAD) , Mean-Squared Difference (MSD) or Cross – Correlation Function (CCF)[6] .

In default mode , the range of the vectors is limited to [-16,15.5] , and the vectors must point to a location inside the picture . The calculated motion vectors are outputted to the variable – length for transmission across the channel .

The next step in the motion – compensated prediction block is to replace each 16x16 block with its match . This output is later subtracted from the current frame . The resulting difference is easier to compress since it contains lower energy than the original . The output is also used in the reconstruction of the

previous frame in the frame buffer . Figure(٢) illustrates a process of block-matching algorithm. In a typical BMA, each frame is divided into blocks, each of which consists of luminance and chrominance blocks. Usually, for coding efficiency, motion estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. These candidate blocks are just the displaced versions of original block. The best (lowest distortion, i.e., most matched) candidate block is found and its displacement (motion vector) is recorded. In a typical interframe coder, the input frame is subtracted from the prediction of the reference frame. Consequently the motion vector and the resulting error can be transmitted instead of the original luminance block; thus interframe redundancy is removed and data compression is achieved. At receiver end, the decoder builds the frame difference signal from the received data and adds it to the reconstructed reference frames. The summation gives an exact replica of the current frame. The better the prediction the smaller the error signal and hence the transmission bit rate[6].

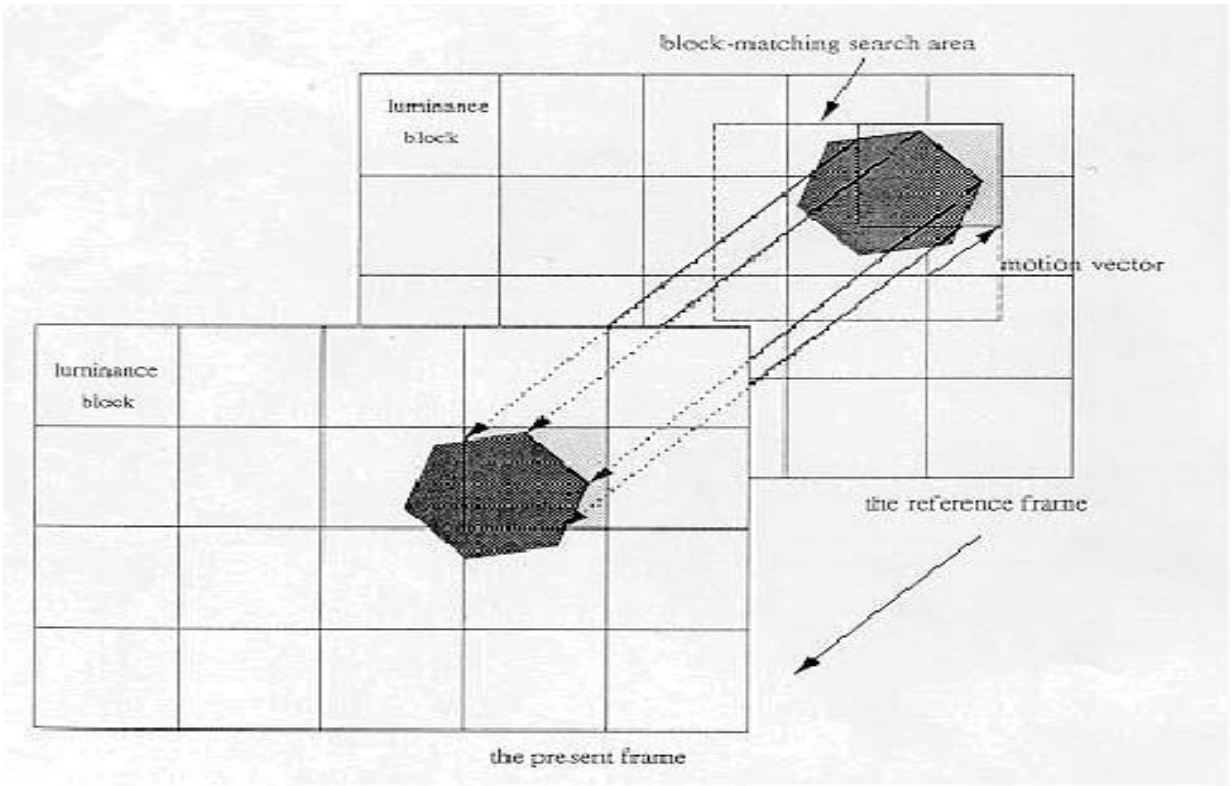


Fig. (٢): Corresponding blocks from present and reference frame , and search area in the reference frame [6].

3-1Video Sequence Structure

Video frames are typically encoded as a series of groups of pictures (or GOPs). Each frame within a GOP is encoded as one of three types, according to the type of prediction used in encoding. Figure (3) below shows a typical arrangement of video frames within a GOP. The frames are shown in the order that they would be displayed. The order that the frames are encoded in is determined by the inter estimation dependencies [7].

Intra Frames

The first frame of a GOP is an Intraframe (or I-frame). I-frames are encoded using only intra methods pixel prediction. Since all predictive pixels are from the same frame, they are coded independently of all other frames[2].

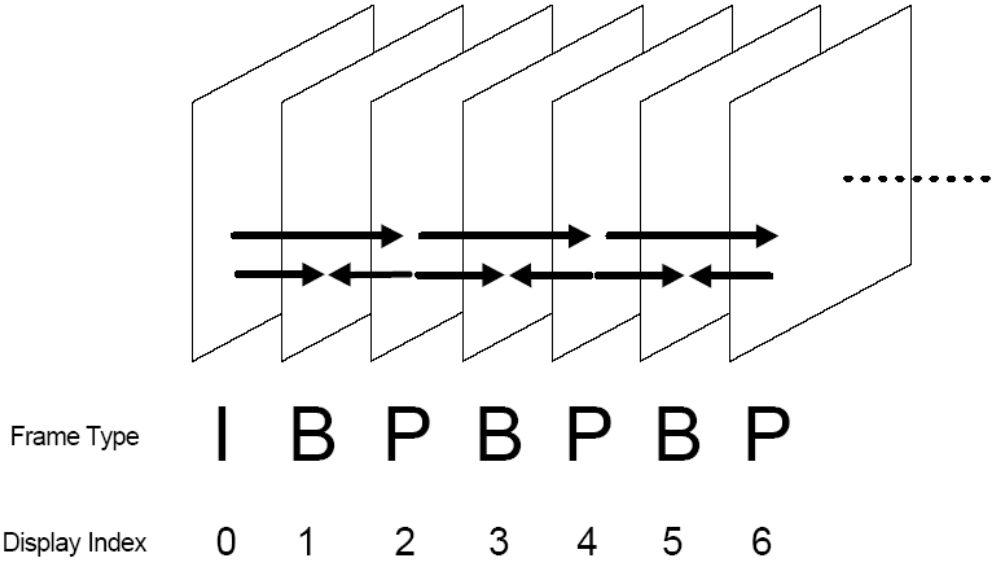


Fig. (3): Typical frame ordering of a group of pictures [7].

Predictive Frames

Predictive frames (or P-frames) use interframe prediction methods as well as intraframe methods. For P-frame motion compensation, only forward prediction is supported – frames used for prediction must temporally precede the encoded frame. In H.264, multiple reference frame prediction is permitted, i.e. P-frames pixel blocks may be predicted from any preceding I-frame or P-frame. This feature is useful for encoding transitionally covered background and periodic non-translational motion [8].

Bi-Predictive Frames

Bi-predictive frames (or B-frames) use an expanded set of inter-prediction methods compared to P-frames. Specifically, B-frames support forward and backward prediction for motion compensation – reference frames may occur before or after the encoded frame in the display order of the video sequence. In addition, H.264 B-frames support bi-predictive block compensation. In this method, the predictive blocks may be calculated as a combination of blocks that occur in different frames. In early video compression standards, B-frames were predicted from previously encoded I-frames or P-frames as shown in Figure(٣). The H.264 standard allows motion compensating prediction from

any previously encoded frame in the video sequence, including prior B-frames[9].

3-2 Selection Block Size

Block size affects the performance of techniques . The larger block size , the fewer number of blocks , and hence fewer motion vectors need to be transmitted . However borders of moving objects do not normally coincide with the borders of blocks and so larger blocks require more correction data to be transmitted[10].

Small block size results in a large number of motion vectors , but each matching block is more likely to closely match its target and so less correction data is required . Lallaret et al .found that if the block size was too small then the compression system would be very sensitive to noise[11]. Thus block size represents a trade off between minimizing the number of motion vectors and maximizing the quality , and compression ratio has been the subject of much research and is well understood. For architectural reasons the block size is taken as integer power of 2 and so block size of 8 and 16 pixels is more predominate . Both the MPEG and H.261 video compression standards use blocks of 16x16 pixels[12].

3-3Block Matching Methods

The purpose of block-matching is to match the current block of pixels with a similar block in a reference frame. Each block of pixels in a reference frame is identified by the vector that defines its position in the frame [13].

Block-matching methods are the most widely used motion estimation methods for the low computation complexity compared with other methods such as optical flow based methods and percussive methods. It is also adopted by many video coding standards (MPEG-1/2 and H.261.262/263) [13].

During block matching each target block of the current frame is compared with the past frame in order to find the matching block .

When the receiver reconstructs the current frame this matching block is used as substitute for the block from the current frame . Block matching takes place only on the luminance component of frame .The color component of the blocks are included when coding the frame but they are not usually used when evaluating the appropriateness of potential substitutes or candidate blocks[13].

Many fast block-matching methods are developed based on some basic assumptions. Among them, there are two assumed by many block-matching methods.

1. The block of pixels has the same translation motion from frame to frame.
2. Block distortion measure (BDM) increases monotonically as the checking point moves
away from the global minimum BDM point.
Figure (4): depicts the search area of the block-matching methods. The current frame is divided into many rectangle blocks. For each blocks the motion displacement is achieved by finding the displaced coordinate of a match block within the search window of a reference frame. The 'F' block is the block of the current frame , 'G' block is the block of

the reference frame ,p is the block size and the maximum number of the checking point in vertical direction of the search window is 2m+p and 2n+p in horizon direction. A best-matched motion vector is obtained by finding within a search window of size (2m+p)(2n+p) in the reference frame.
In block matching techniques , the goal is to estimate the motion of a block of size (nxm) in the present frame in relation to the pixels of the previous or the future frames, the block is compared with the corresponding block within s search area of size (m+2p)(n+2p) in previous(or the further)frame, as illustrated in figure 4[13].

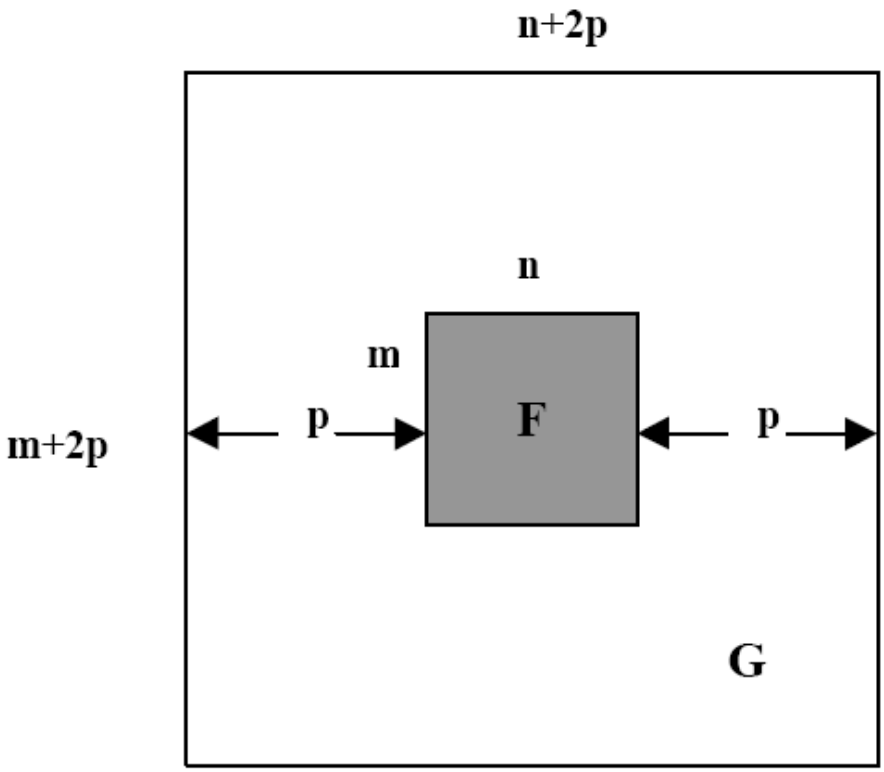


Fig .(4) : The search area in block matching methods[13].

3.4 Cost functions

In order to measure the similarity between the block of the current frame and a candidate block of the reference frame (obtain the motion vector by minimizing a cost function so that the prediction residual has less energy), three matching cost functions are defined:

i)The Mean –Absolute Difference(MAD) , is defined as :[13]

$$MAD(dx,dy)= \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} |F(i,j)- G(i+dx, j+dy)|, \quad \ldots\ldots\ldots\ldots (1)$$

Where F(i,j) represents a (m x n) macroblock from the current frame ,

G(i,j) represents the same macroblock from a reference frame (past or future)

{dx,dy} a vector represents the search location
The search space is specified by
dx= {-p , +p } , and
dy= {-p ,+p }

ii) The Mean-Squared Difference (MSD) cost function is defined as: [13]

$$MSD(dx,dy)=\frac{1}{mn}\sum_{i=-n/2}^{n/2}\sum_{j=-m/2}^{m/2}[F(i,j)-G(i+dx,j+dy)]^2 \quad , \ldots\ldots\ldots\ldots (2)$$

iii) The Cross – Correlation Function (CCF) is defined as :[13]

$$CCF(dx,dy)=\frac{\sum_{i=n/2}^{n/2}\sum_{i=m/2}^{m/2}F(i,j)G(i+dx,j+dy)}{\left[\sum_{i=-n/2}^{n/2}\sum_{j=-m/2}^{m/2}F(i,j)\right]\left[\sum_{i=-n/2}^{1/2}\sum_{j=-m/2}^{n/2}\sum_{j=-m/2}^{m/2}G(i+dx,j+dy)\right]^{1/2}} \quad , \ldots \ldots (3)$$

The mean absolute difference (MAD) cost function is considered as a good candidate for video application , because it is easy to implement in hardware . The other two cost

functions (i.e , MSD and CCF) , can be more efficient , but are too complex for hardware implementations.[13]
To reduce the computational complexity of MAD , MSD , and CCF cost functions , Gharavi and Mills have proposed a simple block matching criterion , called pixel difference classification (PDC) is implemented[12]. The PDC criterion is defined as :[13]

$$PDC(dx,dy)= \sum_{i=1}^{8}\sum_{j=1}^{8} T(dx, dy, i, j), \ldots\ldots\ldots \ldots\ldots\ldots (4)$$

For (dx , dy)= {-P , P }

T(dx ,dy , i , j) is the binary representation of the pixel difference defined as :

$$T(dx,dy,i,j) = \begin{cases} 1 , | F(i,j) - G(i + dx , j + dy) | \leq t ; \\ 0 , \text{otherwise} ; \end{cases} \quad \ldots\ldots\ldots\ldots (5)$$

Where t is a pre-defined threshold values .

In this way , each pixel in a macroblock is classified as either a matching pixel (when T=1) or a mismatching pixel (T=0) . The block that maximizes the PDC function is selected as the best matched block [13].

3.5 *Block-matching algorithms (BMA)*

Many block – matching techniques for motion vector estimation have been developed and evaluated in the literature , among these techniques are :

1. The Exhaustive Search Algorithm

The Exhaustive Search Algorithm is the simplest but computationally intensive search method , which evaluates the cost function at every location in the search area . If MSD cost function were used for estimating the motion vector , it would be necessary to evaluate (2p+1) MSE functions . For (p=6), it gives 169 iterations for each macroblock[14] .

2. The Three-Step Search Algorithm

The three-step search algorithm was proposed by Koga et al, and implemented by Lee et al [14]. In this algorithm first calculate the cost function at the center and eight surrounding locations in the search area . The location that produces the smallest cost function (typically MSD function is used)becomes the center location for the next step , and the search range is reduced by half[11] .
For illustration , a three –step motion vector estimation algorithm for p=6 is shown in figure (6). It consists of the following steps [11]:

Step1: In the first step , nine values for the cost function MAD (for simplification purposes denoted as (M) are calculated : M1=M(0,0), M2=M(3,0), M3=M(3,3), M4=M(0,3), M5=M(-3,3) ,M6=M(-3,0), M7=M(-3,-3) , M8=M(0,-3),M9=M(3,-3), as illustrated in figure (6) ,assuming that M3gives the smallest cost function , it becomes the location for the next step .

Step2: Nine new cost functions are calculated , for M3 and surrounding 8 locations , using a smaller step equal to 2 . These nine points are denoted in fig. (6) as M11,M12,M13, and M19 .

Step3: In the last step , the location with the smallest cost function is selected as a new center location (in the example shown in fig.(6) the location is assumed M15 , and 8 new cost functions are calculated surrounding this location M21,M22,M23,and M29 .The smallest value is the final estimate of the motion vector{dx,dy} equal to {1,6}. Note that the total number of computations of the cost function is 9x3-2=25 , which is much better than 169 in the exhaustive search algorithm .

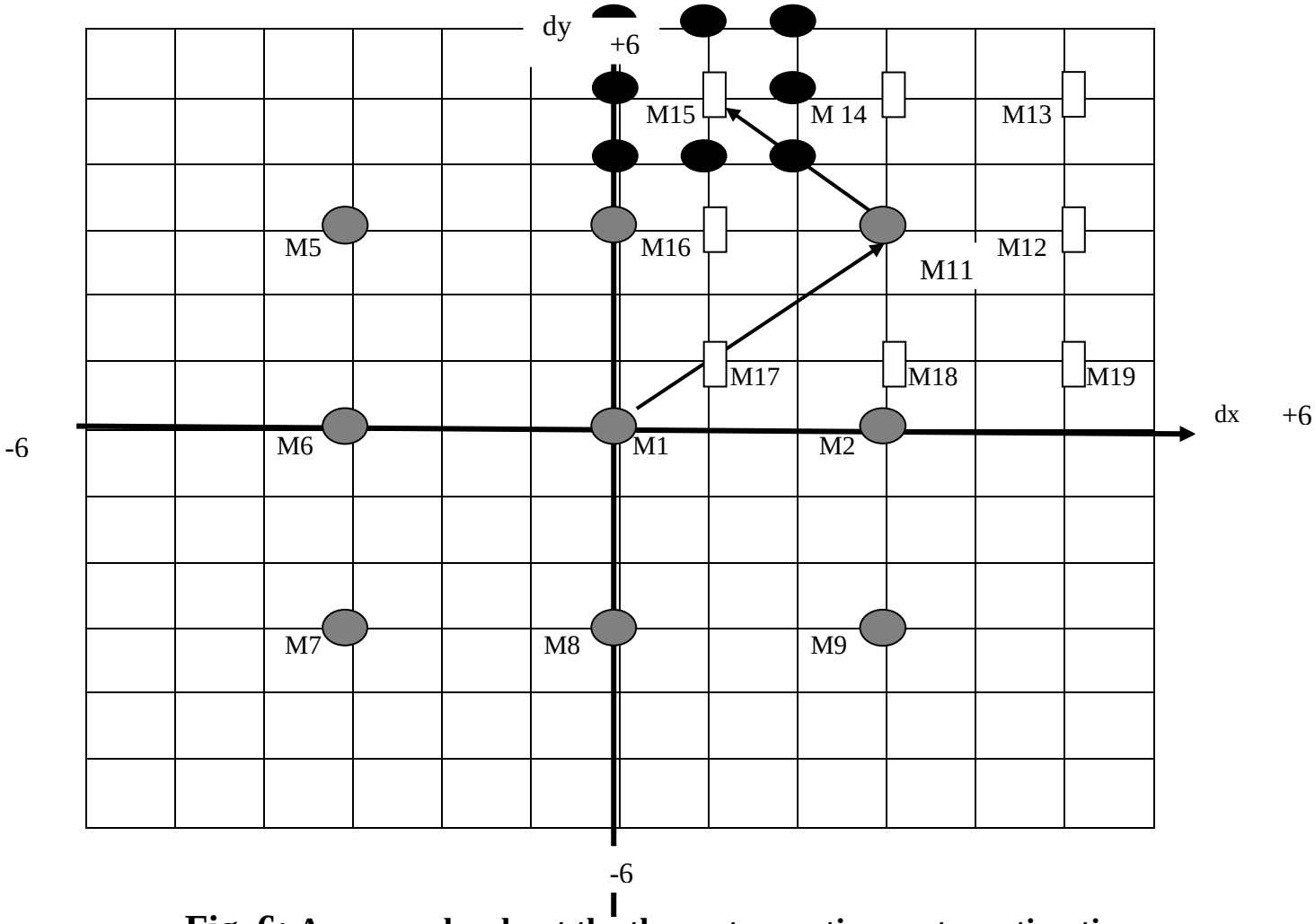


Fig. 6: An example a bout the three-step motion vector estimation algorithm .[11]

3. (2-D) Logarithmic Search Algorithm

This algorithm , proposed by Jain, uses the MSD cost function and performs a logarithmic 2-D search along a virtual direction of minimum distortion (DMD) on the data within the search area . The modified version of the algorithm described by Srinivasan and Rao[15], uses the MAD cost function , and can be described using the following steps as illustrated in figure(7) , The 2-D logarithmic search algorithm consists of the following steps:[11]

Step1: The MAD function is calculated for dx=dy=0,M(0,0) , and compared with a threshold (lets say the value is 4 out of 255): M(0.0) < T . If this is satisfied , the tested block is unchanged and search is completed .

Step-2a: The next four cost functions are calculated ,M1(4,0).M2(O,4),M3(-4,0) , and M4(0,-4) , and their minimum (M') is found and compared to M (0,0) :

M' =min(M1,M2,M3,M4)

If the minimum (M'>M(0,0)) then go to step3 , otherwise this value is compared against the threshold (T). If (M'< T) the value M' is the minimum and the search ends .Otherwise , the algorithm continues with step 2b .

Step -2b:Assuming that in the previous step 2a , the minimum M' =M1(4,0) then the next two surrounding positions are calculated : M5(4,4) and M6(4,-4) .

As indicated in fig. (7) the test for minimum and threshold are performed again , and if the minimum is found , the procedure is complete . Otherwise , step 3 continues .

Step-3: Assuming that the new minimum location is M5(4,4) , a similar search procedure (step 2a and 2b) is continued , except the step is divided by 2 . In fig. (7) , the new minimum is assumed at M(2,4).

Step -4: The step is further reduced by 2 , and the final search (step 2a and 2b) is performed . The minimum (dx ,dy) is found ; in fig.(7) it is (1,5) . For p=6 , this algorithm requires maximum (19) cost function calculations, as shown in fig.(7)

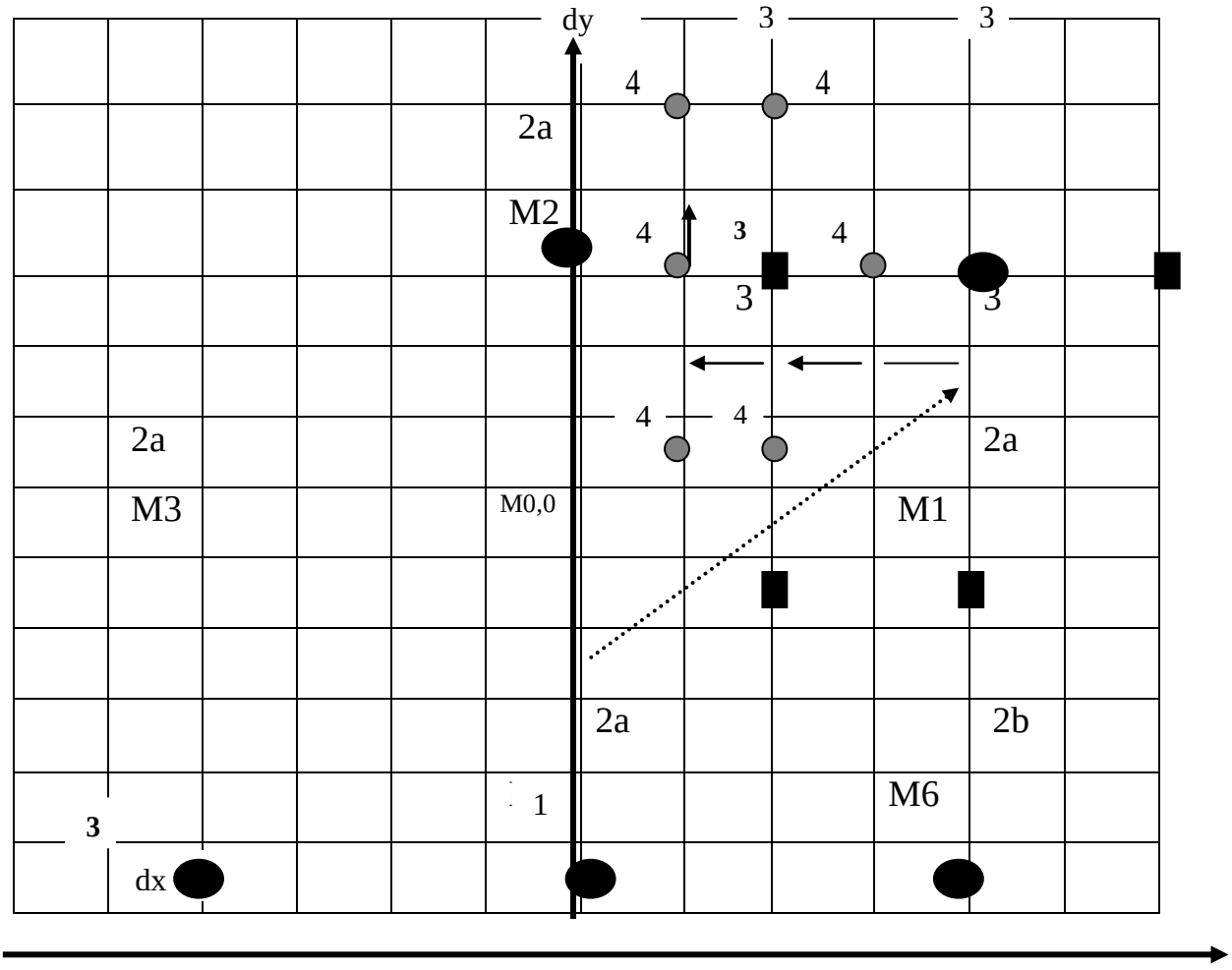

dy
3
3
4
4
2a
M2
4
3
4
3
3
4
4
2a
2a
M3
M0,0
M1
2a
2b
1
M6
3
dx

Fig.7: An example about the modified 2D logarithmic search algorithm.[11]

4- The Conjugate Direction Search Algorithm

This algorithm is designed for motion vector estimation , it is proposed by Srinivasan and Rao . It is an adaptation of the traditional iterative conjugate direction search method . This method can be implemented as one – at- a time search method , as illustrated in figure (8).
In figure (8) , the direction of search is parallel to one of coordinate axes , and each variable is adjusted while the other is fixed . This method has been adapted for motion vector estimation [15] , as illustrated in figure (8) , the algorithm consists of the following three steps :

Step-1: The values of the cost function MAD in the dx -direction are calculated , until the minimum is found . The calculation is as follows :
(a) M(0,0) , M(1,0) , and M(-1,0) ; (b) If M(1,0) is the minimum , M(2,0) is
computed and evaluated , and so on . This step is complete when a minimum in the
dx -direction is found in fig. (8) the minimum is assumed M(2,0).

Step-2: The search continues in the dy- direction by calculating cost function M(2,-1) and M(2,1) .A minimum in the dy -direction is then found at M(2,2) in figure (8).

Step -3: The direction of search is now the vector connecting the starting point (0.0) and the obtained minimum (2,2). The following cost function are calculated and evaluated next :M(1,1) and M(2,2) , and so on , until a minimum in the direction is found .In the example in figure (8) , the minimum is M (4,4) , and the obtained motion vector is dx=4 and dy=4. It may happen that the dx and dy vector , obtained in step 2 and 3 , do not constitute a square as given in figure (8) . In that case , the nearest grid points on the direction joining (0,0) and the obtained minimum point are selected [15]

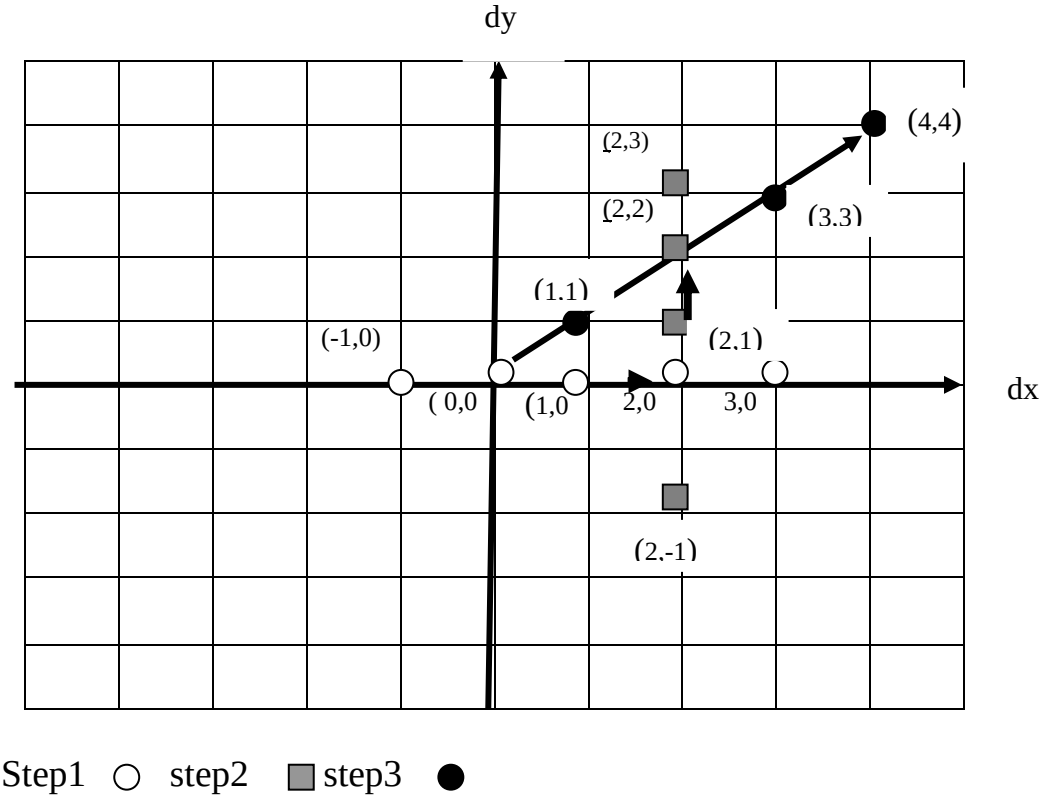


Fig.8:An example about the conjugate direction search method for motion vector estimation.[15]

3.6 Suggested Search Method:

In the current research , a new algorithm was suggested . It is based on merge two algorithms(2D Logarithmic Search Algorithm and Conjugate Direction Search Algorithm) in one algorithm to get more high compression efficiency by handling the time delay associated with all searching methods to speed up the image video compression , at the same time, we keep the image quality .

In this method , the microblock is dividing into four 8x8 sub- blocks , and then implementing searching for each block separately such that each search will give minimum value of matching criterion (i.e., MAD or MSD) ,then the average value of matching criterion (minimum) and the displacement parameters(i.e., dx,dy) are taken.

In case where the average differences(average minimum) is above the threshold specified by the user of the program ,then the blocks will be compressed by DCT ,quantization, and VLC,and transformed .

While in the case where the average minimum is less than a threshold , then these blocks will not compressed , and the search for microblock will stop.The following is the detailed algorithm.
The result of applying this mechanism indicated better performance than the corresponding result obtained by applying the whole macroblock , one at a time . The range of search (i.e , dx & dy) for each block was taken {-6 ,+6}.

The suggested search algorithm consists of the following steps:

1- The macroblock is partitioned into four 8x8 sub – macroblocks.
2- For each sub-macroblock compute the maching criterion value (MAD or MSD)for the center position.
3- If the criterion value(minimum) <=threshold then exit .
4- Take four different point and compute the matching criterion value for four point for sub-macroblocks.
5- Take the smallest value (minimum).
6- If the minimum <M(0,0) then if the minimum<=threshold then exit .
7- select three points about the centre of the search window. (horizontal direction) and Compute the matching criterion value in this direction (dx – direction) until the minimum in this direction is found and test with the threshold as in step 6
8- Compute the matching criterion value in the dy – direction (vertical direction) starting with the point that has minimum value in dx direction in previous step until the minimum in this direction is found and and test it with threshold as in step6 .
9- Compute the matching criterion value in the vector connecting the starting point(0,0) and obtained minimum in step8, until a minimum is found in this direction and test with threshold in step6 .
10- Obtained the best motion vector that represent the position of the best match block.
11- Exit

The minimum that is represented the best match block and its displacement(motion vector(dx,dy)) is reported after the search is complete

Figure(9) presents an example for choosing the pixels for a sub – macroblock . The following are steps of this example .

Step-1: The MAD function is calculated for dx=dy=0 , M(0,0) , and compared with a threshold: M(0,0) <T . If this is satisfied , the tested block is unchanged and the search is completed .

Step-2: The next four cost functions are calculated , M1(4,0) ,M2(0,4), M3(-4,0) ,

M4(0,-4) , and their minimum (M`) is found and compared to M(0,0) :

$$M` = \min (M1, M2, M3, M4)$$

If the minimum (M`> M(0,0) then go to step 3 , otherwise this value is compared against the threshold (T) . If (M` < T) the value M` is minimum and the search ends .Otherwise , the algorithm continues with step 3.

Step-3: The values of cost function MAD in the dx- direction are calculated , until the minimum is found . The calculation is as follows : a) M(1,0) and M(-1,0) ; (b) If M(1,0) is the minimum , M(2,0) is computed and evaluated , and so on .This step is complete when a minimum in dx- direction is found and test with the threshold. In figure (9), the minimum is assumed M(2,0).

Step-4: The search now continues in dy-direction by calculating cost function M(2,-1) and M(2,1) . This step is complete when a minimum in dy-direction is found and test with the threshold . A minimum in the dy-direction is then found in M(2,2) in figure(9).

Step-5: The direction of search is now the vector connecting the starting point (0,0) and the obtained minimum (2,2) . The following cost function are calculated and evaluated next :M(1,1) and M(3,3) , and so on , until a minimum in the direction is found and test with threshold .
In figure(9) . the minimum is M(4,4) , and the obtained motion vector is dx=4 and dy=4 .

This minimum is corresponding the best match block and its displacement(motion vector(dx,dy)) is reported after the search is complete.

The above process will be applied to four blocks and if the result of matching criterion value (MAD or MSD) for each block is less than threshold and then the result of MAD will be taken for each block , and then will be added and divided by four ,and the average minimum will be compared with a certain

threshold specified by the user of the program , if the result of MAD is less than threshold then the algorithm is complement , otherwise the four blocks will be compressed by DCT, quantization , and VLC.

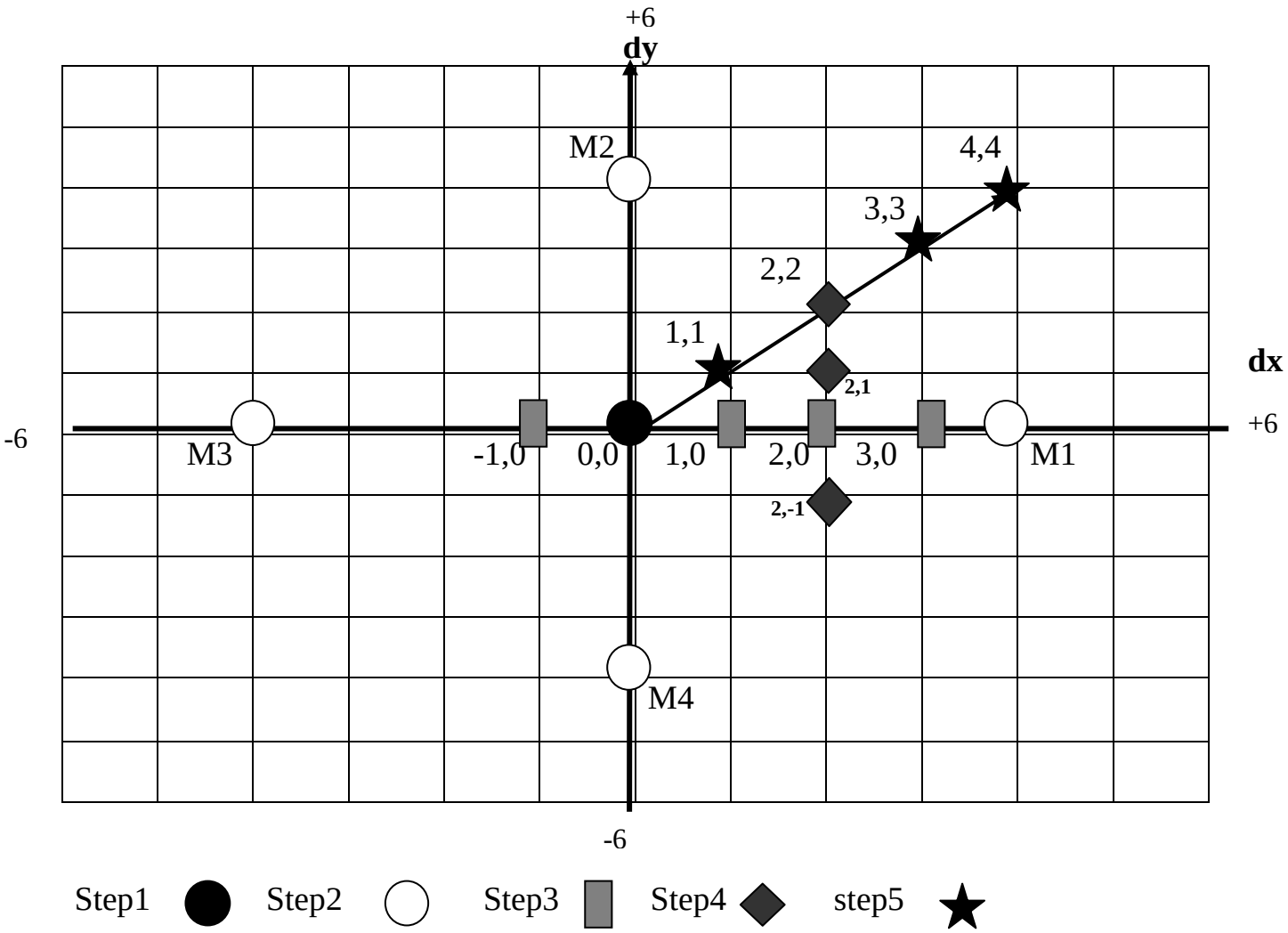
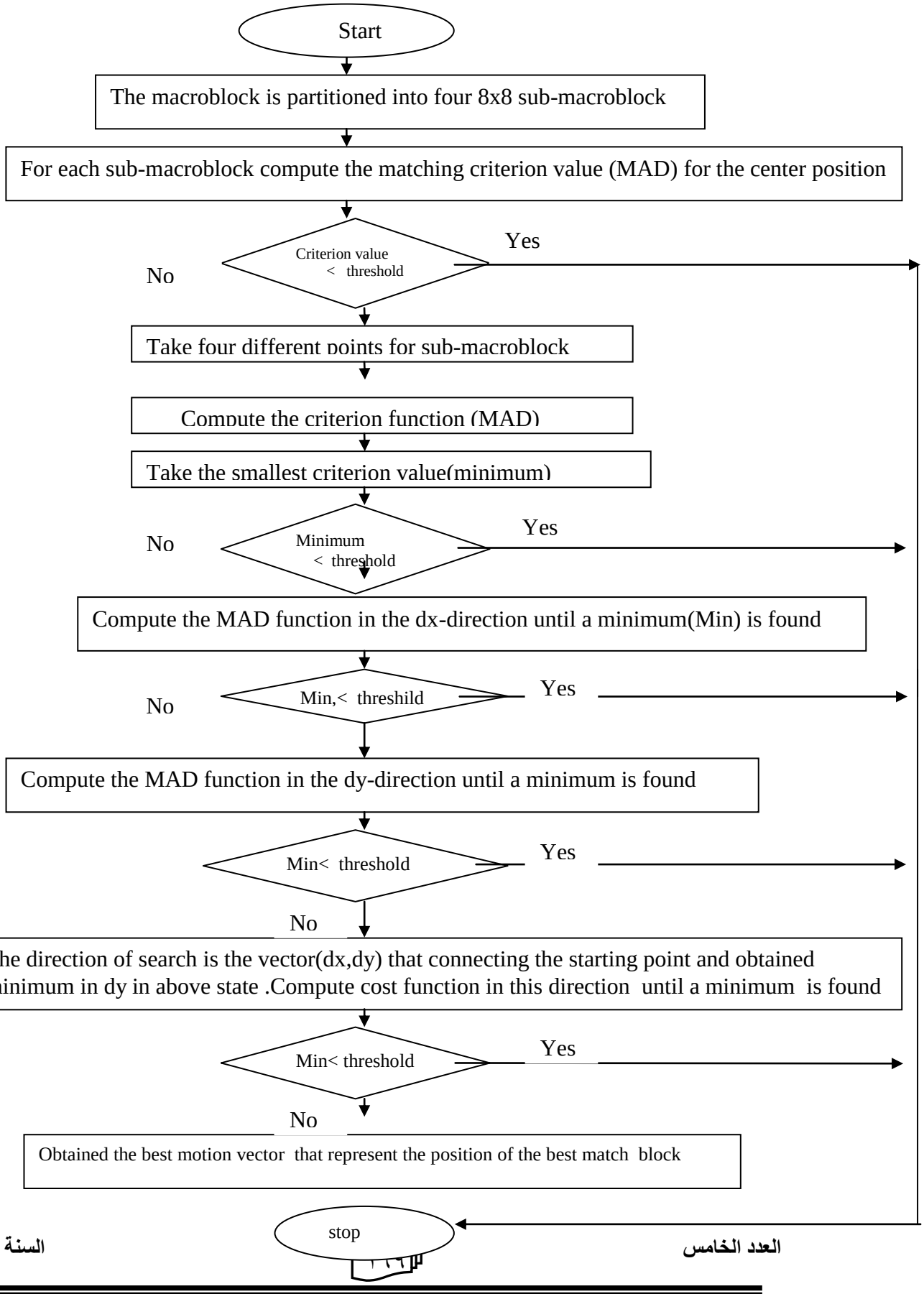


Fig . (9) An example about the suggested adaptive algorithm .

The flow graph of the suggested search method .


Start
The macroblock is partitioned into four 8x8 sub-macroblock
For each sub-macroblock compute the matching criterion value (MAD) for the center position
Criterion value < threshold
Yes
No
Take four different points for sub-macroblock
Compute the criterion function (MAD)
Take the smallest criterion value(minimum)
Minimum < threshold
Yes
No
Compute the MAD function in the dx-direction until a minimum(Min) is found
Min,< threshild
Yes
No
Compute the MAD function in the dy-direction until a minimum is found
Min< threshold
Yes
No
The direction of search is the vector(dx,dy) that connecting the starting point and obtained minimum in dy in above state .Compute cost function in this direction until a minimum is found
Min< threshold
Yes
No
Obtained the best motion vector that represent the position of the best match block
stop

4-Experimental Analysis

Six different video sequence are taken as test samples . In more detail the performance of two searching methods(2-D logarithmic and The Conjugate Direction method) are investigated for video sequences . The blocks thresholds are taken as control parameters to discuss the performance of these methods.

All the computer programming work was done by using Delphi language (version 3) . The program are implemented and executed for testing purposes using IMB personal computer , will processor Pentium III (500 MHZ)

4-1 Key parameters

In this paper we will use four key parameters as shown below:

4-1-1 Compression Ratio (C_r) :

It is the degree of image file size(data) reduction due to compression process .

This ratio represents the size of the original uncompressed image file to the size

of overall compressed data file [2]

$$C_r = \frac{\text{Uncompression file size}}{\text{Compression file size}}$$

The (C_r) parameter is an indicator the compactness ability of the compression process.

4-1-2 : Searching Time :

It is the overall time required to perform the searching process for all the blocks of

the successive frames .

The minimization of searching time is considered as the most cost criteria (MAD or

MSD) , which will indicate the efficiency of the matching mechanism .

4-1-3 Fidelity PSNR.

Peak Signal –to-Noise Ratio (PSNR) is one of the most popular measure utilized to evaluate the image quality of the coded images using the compression methods .

4-1-4 The Number of Blocks (NB) :

It represent the number of blocks handled by the motion estimation process , because

the coding of blocks by motion estimation requires a little number of bits in comparison with those blocks by using DCT.

4-1-4 Blocks Thresshold

The blocks threshold are used as parameter to decide if the matched blocks are similar or not . The overall difference between all the pixels of the blocks will be compared with the adopted block threshold to decide if the compared blocks are similar or not . Practically the effective block threshold values start from the (4) to the (7) in the program .

5- The Test Video Sequences

The mechanism of the testing procedure is to investigate the cases , which lead to a significant reduction in coding time without significant loss in the image quality and compression performance . Four different video sequence of image are taken into consideration. Table (1) present the specification of the tested video sequences

.

Table(1) : Characteristics of video sequences

Name of sequence	No. of frame	Frame size pixel	I Frames	
			Cr	PSNR
Explosion	11	352*240	28.7	66.75
Spreader	11	352*240	14.0	45.99
Mouse	11	352*240	31.5	51.35
Clock	11	352*240	16.8	42.8

6-The Experimental Result

Tables (2) to (7) present the compression parameters obtained by applying the considered motion estimation methods upon the 4-video sequences. The selection of the motion estimation method is based on the criteria that the present reconstructed images are those produced by using the motion estimation method combined with the proposed matching technique with a minimum coding time in comparison with those associated with other estimation methods**.**

Table (2) : Spreader image with threshold =4

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D logarithmic search	45.71	30.40	6470	0:1:50
The conjugate direct search	45.20	30.50	6530	0:0:50
Suggested search	43.80	30.60	6670	0:0:22

Table(3): Mouse image with threshold=4

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D logarithmic search	81.01	45.70	6860	0:1:49
The conjugate direct search	80.10	46.02	6830	0:0:55
Suggested search	80.70	45.50	6830	0:0:24

Table (4): Clock image with threshold =4

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D logarithmic search	62.10	34.72	6329	0:1:48
The conjugate direct search	62.05	34.50	6329	0:0:48
Suggested search	62.40	34.54	6322	0:0:18

Table (5): Explosion image with threshold=4

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D logarithmic search	61.34	36.14	7390	0:1:50
The conjugate direct search	60.29	36.40	7469	0:0:56
Suggested search	60.18	36.50	7469	0:0:26

Table(6) : Explosion image without half-pixel search step

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D logarithmic search	52.70	35.14	7421	0:1:44
The conjugate direct search	51.49	35.49	7505	0:0:49
Suggested search	52.47	35.07	7545	0:0:26

Table 7: Explosion image with threshold=7

Type of motion estimation	Overall			
	Cr	**PSNR**	**NB**	**Search time**
2-D logarithmic search	70.30	34.54	7012	0:2:1
The conjugate direct search	68.12	34.78	7090	0:0:50
Suggested search	68.32	34.70	7084	0:0:26

Conclusions:

1- The suggested method is found faster than classical methods, without causing a significant degradation in PSNR or Compression ratio.

2- A comparison between the results of suggested method with the results of both methods(2-D logarithmic and Conjugate Direction),indicated that the compression ratio is less by a small ratio ,but the total PSNR is higher ,the number of compressed blocks is greater or equal to number of blocks in Conjugate Direction .The search time for suggested method is faster in average by about ~ 60-80% than previous methods(2-D logarithmic and conjugate Direction).

3- We found in case of using integer pixel instead of half-pixel in the explosion image sequences for three methods(2-D Logarithmic , The Conjugate Direction and Suggested method) ,the number of compressed blocks will increase in all methods, the search time is nearly , the compression ratio and PSNR will decrease .

4- In the case of increasing threshold value in explosion image for all methods , the compression ratio will increase,the PSNR will be less and the number of compressed block is less .

5- If we don't use integer pixel method ,the compression ratio will increase while the searching time will show a small amount of increase ,also the PSNR show better results.

References

1-Gonzales, R.C., and Woods, R.E., "**Digital Image Processing** ", Addison-Wesley, 1992.

2-Tanenbum,A.S**.,"Computer Networks"** by Prentice –Hill,Inc, 1996.

3-Umbaugh,S.E., **"Computer Vision and Image Processing:A practical Approach using CVIP Tool"** prentice- Hell , INC., 1998

4- Mullen, K. T., **"The contrast sensitivity of human colour vision to red-green and**
blue- yellow chromatic gratings *Journal of Physiology"*, V ol. pp. 381-400, 1985.

5- Mitchell, J. L., W. B. Pennebaker, C. E. Fogg, and D. J. LeGall. **"*MPEG Video***
Compression Standard", Boston: Kluwer Academic Publishers, 1996.

6-http://www.cis Ohio- state.edu\jain\cis 788-99\compression\index-html**."Video compression :MPEG4 and Beyond "**.(1 to 3)[2\7\2001]

7- Marlow, S., J. Ng, and C. McArdle, **"Efficient motion estimation using multiple**
log searching and adaptive search windows," *Image Processing and Its Applications, Sixth International Conference on,* pp. 214-218, 2002.

8- Cote, G. and L. Winger**," Recent Advances in Video Compression Standards"**
IEEE Canadian Review, vol. pp. 21-24, 2002.

9- VCEG, Joint Video Team of ISO IES MPEG and ITU-T. Joint Committee Draft
(CD) of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). Wiegand, T. 2002.

10-Lin,D,W., Chen,C.T., Hsing,T.R**.:"Video on phone Lines":**Technology and application proceeding of IEEE,Vol.83, No.2,pp.175-193 and Feb.1995.

11-LallAuret,F.,Barba ,D.,**"Motion Compensation by Block matching and Vector"** post-processing in sub –band Coding of TV Signals at 15 Mbit\s proc.Vol.1605,pp.26-36,ISBN:0-8194-0742-9 , 1995.

12-Nelson,G**."The Block Matching Approach to Motion Estimation ,M.SC. Dissertation "** ,Department of Computer Science, university of Warwick ,UK.Sep.1993

13-Rodriguez,A.,Chan,E.,Ghandi,R., and Penchanathan,S**.,"Experiments on Block Matching Techniques for Video Coding Multimedia System"**,Vol.2,No.5,pp.228-241,Dec.1994

14-Koga,J., Iinuma,K.,Hirano,A.,Iijma,Y.and Ishiguro, T.,**"Montion Compensated Interframe Coding for Video Conferencing"**,Proc.of the National Telecommunication Conference,pp.G5.3.1-5 .3.5,1981

15-Srinivasan, R. and Rao,K.R**."Predicative Coding Based On Efficient Motion Estimation"** , IEEE Transactions on Communications,Vol.33,pp.888-896,1985.

الملخص :

الصوره الرقميه ذات الدقه العاليه تتطلب كميه كبيره من البيانات اثناء عمليه خزنها او نقلها الى الخارج . ضغط الصوره الرقميه تهتم بتقليل عدد البت المطلوبه لتمثيل الصوره الرقميه ،حيث ان تطبيقات ضغط البيانات تكون بشكل رئيسي في نقل او خزن المعلومات الرقميه .

البحث الحالي يستخدم نظام الضغط الفديوي H.263 بتطوير كل البرامج المطلوبه .

في هذا البحث تم اقتراح خوارزميه بحث جديده لمعالجة تاخير الوقت المرتبط بطرق البحث لتسريع عملية ضغط الصور الفيديويه . على الرغم من ان الخوارزميه المقترحه لا تؤثر على كفاءة الضغط ونوعيه الصوره .