Research Journal of Applied Sciences, Engineering and Technology 12(4): 498-506, 2016
DOI: 10.19026/rjaset.12.2390
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 30, 2015 Accepted: October 30, 2015 Published: February 25, 2016

Research Article

Block Matching Algorithm Using Mean and Low Order Moments

Zainab J. Ahmed and Loay E. George
Department of Computer Science, College of Science, Baghdad University, Baghdad, Iraq

Abstract: In this study, a fast block matching search algorithm based on blocks' descriptors and multilevel blocks filtering is introduced. The used descriptors are the mean and a set of centralized low order moments. Hierarchal filtering and MAE similarity measure were adopted to nominate the best similar blocks lay within the pool of neighbor blocks. As next step to blocks nomination the similarity of the mean and moments is used to classify the nominated blocks and put them in one of three sub-pools, each one represents certain nomination priority level (i.e., most, less & least level). The main reason of the introducing nomination and classification steps is a significant reduction in the number of matching instances of the pixels belong to the compared blocks is achieved. Instead of pixels-wise comparisons a set of hierarchal similarity comparisons between few descriptors of the compared blocks is done. The computations of blocks descriptors have linear complexity, O(n) and small number of involved similarity comparisons is required. As final stage, the selected blocks as the best similar blocks according to their descriptors are only pushed to pixel-wise blocks comparison stage. The performance of the proposed system was tested for both cases: (i) without using prediction for assessing the initial motion vector and (ii) with using prediction that based on the determined motion vectors of already scanned neighbor blocks. The test results indicated that the introduced method for both cases (without/ with prediction) can lead to promising results in terms of time and error level; because there is reduction in search time and error level parameters in comparison with exhaustive search and three step search (TSS) algorithms.

Keywords: Block Matching Algorithm, Descriptor, Exhaustive Search, Hierarchal Filtering, Moments, Predicted Motion Vector, Three Step Search (TSS)

INTRODUCTION

Motion estimation is the one of the key elements to achieve video compression by exploiting temporal redundancy of video data. Most of the practical applications on video coding, motion estimation are based on block matching (Reddy and Sengupta, 2008).

Block matching motion estimation was one of the most important modules in the design of any video encoder (Jagiwala and Shah, 2012). The purpose of a block matching algorithm is to find the best matching block, that belong to a reference frame, to represent certain block lay in some other frames, which may appear before or after. This can be used to discover the temporal redundancy in the video sequence, increasing the effectiveness of inter frame video compression and motion detection (Kiran *et al*., 2014). The idea behind block matching is to divide frames into equal sized non-overlapping blocks and calculates the displacement of the best-matched block from the previous frame as the motion vector of the block in the current frame within the search window (Manikandan and Selvakumar, 2014). It matches blocks from the current frame with blocks belong to a reference frame. The displacement (Δx, Δy) in block location from the current frame to the location in the reference frame is called the motion vector, using the fact that the motion between consecutive frames is statistically small and the search range is confined to this area. After each searching process instance, the best match is nominated for each block within the area. The matching criterion means having lowest energy in the sum of residual formed by subtracting the candidate block in search region from the current block located in current frame (Love and Kamath, 2006; Aziz and Dolly, 2012).

Full search (or Exhaustive Search) algorithm is the most computationally expensive block matching algorithm; but it finds the best possible match. The algorithm measures the cost function at each possible location in the search area. Also, it delivers good accuracy in searching for the best match. But, because of the large amount of computation is involved, it is not suitable in real time video coding. The main drawback of this method is that "the large search area needs more computations"(Khammar, 2012). Kilthau *et al*. (2002) presented a new algorithm for solving the block

Corresponding Author: Zainab J. Ahmed, Department of Computer Science, College of Science, Baghdad University, Baghdad, Iraq matching problem which is independent of image content and is faster than other full-search methods. The method employs a novel data structure called the *Windowed-Sum-Squared-Table*, it uses the Fast Fourier Transform (FFT) in its computation of the Sum Squared Difference (SSD) metric (Kilthau *et al*., 2002). Ahmed *et al*. (2011) proposed a new technique called Fast Computations of Full Search (FCFS) algorithm. This technique keeps the resolution of the decompressed videos the same as the one generated from using Full Search Block Matching Algorithm while decreasing the computational time required to determine the matching macro block from the reference frame to the current macro block. This performed by stopping the calculation of the sum absolute difference between the pixels in the current macro block and the macro blocks in the reference frame when the current uncompleted sum absolute value is greater than the previously calculated one (Ahmed *et al*., 2011).

The Three Steps Search (TSS) algorithm was introduced by Koga *et al*. It became very popular because of its simplicity, robustness and nearly optimal performance. It searches for the best motion vectors by following a reducible grid search strategy (Manjunatha and Sainarayanan, 2011). TSS algorithm is used for motion estimation; its saving factor is 100 times greater when compared with Full Search Block Matching Algorithm (FSBMA). The search rounds are fixed to be, almost, three search steps. TSS is inefficient for the estimation of small motion because it utilizes a uniformly allocated pattern in the initial step (Vijay *et al*., 2014). Kulkarni *et al*. (2014) have made comparisons between different three steps search algorithms. The compared algorithms are the Three Step Search algorithm (TSS), New Three Steps Search algorithm (NTSS), improved TSS algorithm, enhanced TS Salgorithm and fast TS Salgorithm. The performance of each algorithm is a compromise between the peak signal to noise ratio and the consumed search time. Choosing an algorithm depends on what we require in application. Some applications need to be executed within real time constraints, while some other applications require high preservation of video fidelity (as in the applications of medical image processing domain) (Kulkarni *et al*., 2014). JieRong and Chang Qing proposed an Improved Motion Estimated Three Steps Search Algorithm as an efficient and fast block matching motion estimation algorithm. The strategy of this search algorithm is to begin searching from two positions the predictive search center and the (0, 0) position, which still hold direction when the predictive error is relatively big. The other strategy is to adopt the big small square search pattern, which chooses search step according to the moving state of objects (Jie-Rong and Chang-Qing, 2011). Bhavsar and Gonawala (2014) proposed algorithmic simulation of Three-Step Search (TSS) block matching algorithm for motion estimation. This method is based on the real world video frame sequence's feature of Center-biased motion vector allocation and uses Center-biased checking point patterns and a small number of search locations to perform fast block matching (Bhavsar and Gonawala, 2014).

The objective of this study is to develop an efficient block matching search using the mean and low order moments of each block in the coded frame. The mechanism of blocks categorization and multistage filtering mechanism are adopted in order to speed up the search process by reducing the overall computational complexity without making significant sacrifice in accuracy. Also, in this study two fast hierarchical schemes for block matching that based on the proposed descriptors comparisons are investigated. The first search scheme is without using predictive motion estimation for initiation the motion vector of each block; and it is called Blind-Search Scheme. The second scheme uses predictive motion estimation and it is called Intelligent-Search Scheme.

THE PROPOSED SYSTEM

The layout of the proposed system is illustrated in Fig. 1. The strategy of block matching is based on determining the mean and centralized low order moments as blocks' descriptors for each overlapped block found in the tested frame and the previous frame's

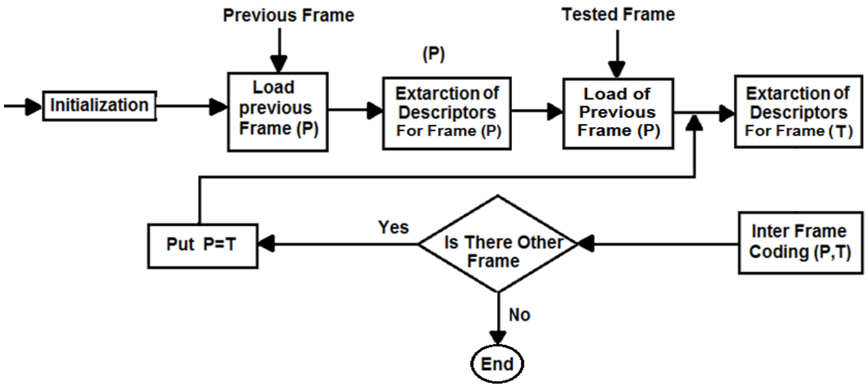

Fig. 1: The layout of proposed system

blocks; with taking into consideration that the descriptors values of each block in the previous and the tested frame are determined only one time. Then, the descriptors of each tested block are compared with corresponding descriptors of the blocks lay within the search area of the previous frame. The descriptors matching process is done in cascade way (i.e., at each stage the similarity of only one descriptor value is considered to filter in/out the scanned blocks and in the next stage the value of other descriptor is used to test the similarity of the only filtered-in blocks to accomplish additional filtering). The main advantage of the proposed mechanism is doing multi stage blocks filtering task using single value comparison operation instead of multi-values comparison (i.e., avoiding the determination of MAE for all pixels belong to the two compared blocks). The proposed system consists of three main stages:

- Initialization stage
- Calculation of descriptors list stage
- Motion assessment stage

Stage-1: Initialization: The first stage in proposed systemis done at beginning of the system; it implies two essential steps:

Determination of spiral search ordering sequence: Spiral search starts with zero displacement and moves spirally to the candidates have larger displacements. Hence, the spiral search, instead of horizontal-vertical search, offer optimized profile for the search order and consequently reduces the search time. The combination of early termination and circular search can significantly speed up block matching without losing prediction performance. In addition, a motion vector of a block will probably have a value close, if not equal, to the motion vector of another adjacent block. The list of spiral sequencing consists of three items:

- The x-shift (Δx)
- The y-shift (Δy)
- The shift distance (D)

Euclidean distance measure is used for ordering search sequence from nearest to farthest block because nearest blocks are more probably similar to tested block and consequently they should treated as most nominated blocks (especially when using predication for determining the initial displacement vector). The reisother type of search ordering called diamond search; if we want to use this type of search instead of circular ordering, we just have to change the distance measure equation. The following equation is for distance computation for both search ordering cases (i.e., circular or diamond):

$$D(i) = \begin{cases} \sqrt{(\Delta x(i))^2 + (\Delta y(i))^2} \text{for circular search} \\ |\Delta x(i)| + |\Delta y(i)| \text{in diamond search} \end{cases} \tag{1}$$

where, $i \in [0, L\text{-}1]$, L is the number of scan locations fall within the search window; $\{Dx(i), Dy(i)\}$ represents the sequence of relative locations lay within search window, $D(i)$ represents the corresponding distance for each location in the search window.

Determination of weights arrays: Weight is defined as the contribution effectiveness of the value of a pixel in determination of the moment of the block holds that pixel. Similar to the moment concept, the introduced weight value is taken dependent on the pixel position relative to block center. In this study, two sets of weights, $w_1()$ and $w_2()$, are taken; they calculated using the concept of odd-symmetric centralized moment, that is:

$$w_1(i) = A(0.5 + i)^{\gamma}\ 0 < i < \frac{L}{2} \tag{2}$$

$$w_1(L - 1 - i) = -w_1(i)\ 0 < i < \frac{L}{2} \tag{3}$$

And:

$$w_2(i) = A\left(\frac{L}{2} - i - 0.5\right)^{\gamma}\ 0 < i < \frac{L}{2} \tag{4}$$

$$w_2(L - 1 - i) = -w_2(i)\ 0 < i < \frac{L}{2} \tag{5}$$

where,

$$A = (L \textstyle\sum_0^{L-1} w_i(i))^{-1} \tag{6}$$

γ is the weight order (i.e., γ<1), due to conducted tests it was found that the values around (γ = 0.15) leads to best performance states. The normalization factor A is imposed to ensure the sum of absolute values of w () coefficients is 1. Figure 2 and 3 illustrate the profile of both weights arrays.

The basic idea behind using two sets of weights is, in first set we give a high weights to pixels located near to the center of block and low weights to pixels near the edge of block to support the inner pixels (pixels near the center). While, the second set of weights supports the outer pixels (close to block outer edge). The use of these two sets is useful to get a good set of features which can characterize.

The blocks and can assess the differences between them. Through the suggested two sets of weights each part of the block participate in different way when determining the set of moments, such that each

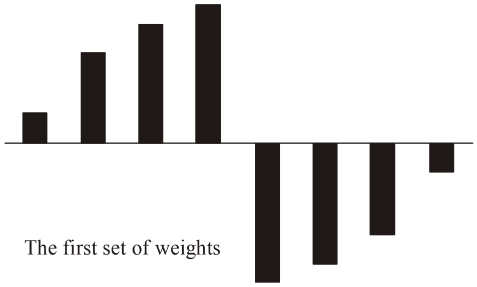

Fig. 2: The 1st set of weights {w$_1$()}

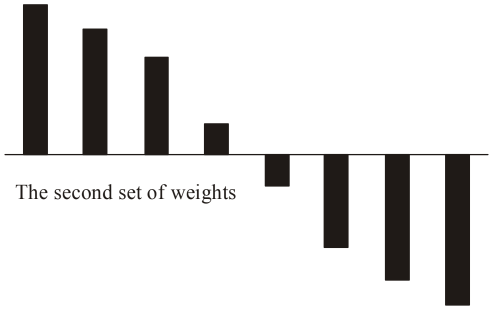

Fig. 3: The 2nd set of weights {w$_2$()}

determined moment is mainly affected by certain parts of block.

The determined weights array are stored in two arrays called "W$_1$()" and "W$_2$()", also they quantified to be integer valued by multiplying the outcomes of Eq. (2-5) by an integer (called Number_of_Bins = 100) and the results are rounded to closest integers. The reason for integer quantification is to avoid floating point operation and keep all calculations to be of integer type.

Stage-2: descriptors calculation: Descriptor is a piece of stored information that is used to identify an item in an information storage and retrieval system. It can save image features that are essential for search and comparison, such that same objects have the similar descriptors. For example, if two different image snapshots taken for same object; then the use of proper set of descriptors can lead to a decision "whether the two objects appeared in the snapshots are similar or not".

In the proposed method, a set of blocks' descriptors have been introduced to accomplish fast similarity assessment between each tested block, belong to predictive frame, with the blocks belong to reference frame (s) and lay within the corresponding search area. The set of used descriptors consists of: (i) mean $M(0)$and (ii) four centralized moments{$M(1)$, $M(2)$, $M(3)$, $M(4)$}, they determined using the followings:

$$M(0,x_s,y_s)=\frac{1}{L^2}\sum_{x=0}^{L-1}\sum_{y=0}^{L-1}P(x+x_s,y+y_s)\quad (7)$$

$$M(1,x_s,y_s)=\sum_{x=0}^{L-1}\sum_{y=0}^{L-1}P(x+x_s,y+y_s)w_1(x)\quad (8)$$

$$M(2,x_s,y_s)=\sum_{y=0}^{L-1}\sum_{x=0}^{L-1}P(x+x_s,y+ys))\quad w1(y)\quad (9)$$

$$M(3,x_s,y_s)=\sum_{x=0}^{L-1}\sum_{y=0}^{L-1}P(x+x_s,y+y_s)w_2(x)\quad (10)$$

$$M(4,x_s,y_s)=\sum_{y=0}^{L-1}\sum_{x=0}^{L-1}P(x+x_s,y+ys))\quad w2(y)\quad (11)$$

where, (x_s, y_s) represent the coordinates of the reference point (e.g., top-left corner or center point) of the block, M(0, x_s,y_s) represents the mean descriptor (sometimes called average); M(1, x_s, y_s) represents the moment along x-axis with high weights give to pixels close to block's center; M(2,x_s,y_s) represents the moment along y-axis with high weights given to pixels close to center; M(3,x_s,y_s) represents the moment along x-axis with high weights assigned to pixels close to block boundary; *and* M(4,x_s,y_s) represents the moment along y-axis with high weights given to pixels close to block boundary.

Since, the use of one descriptor is not enough to categorize a block. Therefore, five descriptors have been used in cascade pattern to gradually filter in the blocks and keep the most nominated similar ones. As first descriptor the mean (M_0) is adopted because it reflects the local brightness in the frame which should change slightly when moving from one frame to the next one. Also, the results of conducted tests to determine the candidacy of using each descriptor alone for discriminating the blocks belong to subsequent frames indicated that the mean descriptor has higher candidacy than the moment descriptors.

At the starting phase of inter-frame coding of each Group Of Pictures (GOP) the descriptors of all blocks belong to the first two frames of GOP are calculated {$M_{prev}(i, x_s, y_s)$, $M_{current}(i, x_s, y_s)$| i=0,2,..4 &∀(x_s, y_s)}; while when encoding the next frames only the descriptors {$M_{current}(i, x_s, y_s)$} of blocks belong to the tested frames are determined and the old values $M_{current}(i, x_s, y_s)$ *are copied to be the* $M_{prev}(i, x_s, y_s)$.

Stage-3: Motion compensation process: In this study, two schemes for blocks motion assessment have been applied, the first is called blind search scheme and the second is called intelligent search scheme.

Blind-Search Scheme (BS): The notation used in the following paragraphs implies the use of prime (') to denote the parameters belong to previous frame. The involved steps of this scheme are:

- For each tested block open a search window for holding all possible overlapped blocks lay in the window and belong to previous frame. Then, put all overlapped blocks in a buffer called the main pool. The blocks are arranged in the pool according to the relative ordering shift array {Δx(), Δy()} calculated in the initialization stage.

- Then, construct three sub-pools, each one will hold a sub set of blocks according to their degree of similarity with the tested block (in terms of descriptors). Each sub-pool has certain priority level when the matching process passes through the whole pixel-wise matching step. For each sub-pool certain threshold value ($V_i|i$ = 1, 2, 3 and $V_1<V_2<V_3$) for descriptors absolute differences and maximum blocks occupation number (N_i) are assigned. The first sub-pool is given the highest priority because it holds the blocks show small absolute difference (*Dif*) in terms of the mean descriptor, such that ($Dif<V_1$). The second sub-pool is given less priority because it holds the remaining blocks which have mean absolute difference less than threshold value (V_2).The third sub-pool has the lowest priority because it holds the remaining blocks which have absolute difference less than (V_3).
- For each block in the main pool:
  - Determine the absolute difference between the mean descriptor of the tested (current) block with mean descriptor of the investigated (pool) block, that is:

$$Dif_0 = \left|M_{tested}(0, x_s, y_s) - M_{pool}(0, x_s', y_s')\right| \quad (12)$$

  - Put the investigated block in one of three sub-pools according to value of Dif_0; if it is less than (V_1) then push the block to1st sub-pool, else if it is less than (V_2) push it to 2nd sub-pool, otherwise if it is less than (V_3) send it to 3rd sub-pool; in case it is more than V_3 keep it in the main pool.
  - After shuffling each investigated block to 1st or 2nd sub-pool, then check the population of both sub-pools; in case they exceed the corresponding maximum allowed numbers (N_1, N_2), then the loop of investigating the blocks in the main pool is stopped and the matching process goes to next step. In case the first two sub-pools are not filled continue testing the main pools till reaching the last block.
- Construct the 4th sub-pool. First, the blocks listed in 1st sub-pool are sequentially arranged in the 4th sub-pool and then the blocks of 2nd are arranged and in case there are empty slots in 4th pool the blocks sorted in 3rd sub-pool are pushed to fill the empty slots. For each listed block in this new sub-pool do the following:

Calculate the absolute difference for the 1st moment descriptor, as follows:

$$Dif_1 = \left|M_{tested}(1, x_s, y_s) - M_{pool}(1, x_s', y_s')\right| \quad (13)$$

where, M_{pool} represents the moment of the investigated block listed in the 4th sub-pool. Then, compute the sum of differences *Sum Dif*(1), as follows:

$$SumDif(i) = \sum_{j=0}^{i} Dif_j \quad (14)$$

If the value of Dif_1 is greater than a predefined threshold value (V_4) or the value of *SumDif*(1) exceed a threshold value (V_{sum}) then the investigated block is shuffled for the 4th sub-pool; and then the next block in the sub pool is investigated.

While in case the $Dif_1$and *SumDif*(1) passed the comparison tests then the procedure of testing the absolute differences of other moments (i.e., *M*(2), *M*(3), *M*(4)) with the corresponding sum of differences {i.e., *SumDif*(*i*)|i = 2, 3, 4} satisfy the matching conditions, where:

$$Dif_i = \left|M_{tested}(i, x_s, y_s) - M_{pool}(i, x_s', y_s')\right| \quad (15)$$

The Matching Conditions are: $Dif_i< V_4$
SumDif(i) <V_{sum}

- For the non-excluded blocks in 4th sub-pool select a number of blocks which show lowest *SumDif(4)* values(number of blocks matching trials ≥ N_3). These blocks are considered as the most nominated blocks which they have the first priority to be matched using the pixel-wise similarity measure (MAD):

$$MAE(x_s, y_s|x_s', y_s') = \sum_{y=0}^{L-1}\sum_{x=0}^{L-1} |P_{tested}(x + xs,y+ys)-Ppool(x+xs',y+ys')| \quad (16)$$

The best matched block (i.e., with lowest MAE) is considered as the best matched block. When the lowest found MAE is high, although it is occurred very rarely, then the remaining blocks in the 2rd and 3rd sub pools are tested using the steps given in (4).

As matching output for each tested block the displacements (Δx, Δy) or the index value (i) of best matched block in the corresponding main pool is registered.

Intelligent-Search Scheme (IS): This algorithm depends on the prediction mechanism. The main idea of the proposed predictive based search system is to use the determined motion vectors of the already searched neighbor blocks to predict the best search region of the currently tested block. The prediction depends on the number and positions of the previously tested neighbor blocks taken into consideration, according to this principle, two prediction cases have been considered; these cases must be used in same order to ensure the success of prediction process. Figure 4 shows the three parts of frame covered by these two prediction cases.

The parts covered by each predictive case are handled according to the following.

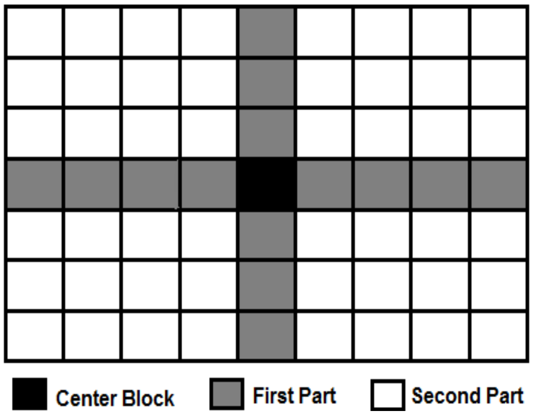

Fig. 4: The two parts of a P frame handled by the 2 prediction cases

For 1st Frame Part: Each block belong to this part have only one neighbor (i.e., top, down, left and right block) that has predetermined motion vector. The initial motion vectors (Δx, Δy) are calculated according to the following:

$$\Delta x(i,j) = \Delta x(i+p, j+k) \quad (17)$$

$$\Delta y(i,j) = \Delta y(i+p, j+k) \quad (18)$$

where, |p|+|k| = 1. The motion vector of neighbor block is used to calculate the initial motion vector of tested blocks.

For 2nd Frame Part: Each block belong to this part is treated as a block has three or numerous surrounding neighbor blocks that have predetermined motion vectors. So, the initial motion vector of the block will be taken as the average motion vector (Δx, Δy) to the three predetermined motion vectors of the neighbor blocks:

$$\Delta x(i,j) = \frac{1}{3}\big(\Delta x(i,j+k) + \Delta x(i+p,j) + \Delta xi{+}k{,}j{+}p \quad (16a)\Delta yi{,}j{=}13\Delta yi{,}j{+}k{+}\Delta yi{+}p{,}j{+} \Delta yi{+}p{,}j{+}k \quad (19)$$

Where, the pair (p, k) will be one of the followings: (1, 1), (-1, 1), (1, -1), (-1, -1).

The key point to the success of such scheme is the ability to predict the tested frame based on the previous ones. A good predictor should produce a motion vector prediction which is very close to the true motion. In ideal case, the prediction results should give the exact motion vectors which render the transmission of unnecessary error.

At first, the descriptor of center block (i.e., block in the core of frame) must be checked to ensure it is a proper initial reference block (i.e., not significantly changed) to start the prediction process. If Dif_0 (using equation 10) is less than a predefined threshold V_5 value then this center block is taken as the initial block to start the prediction process of other blocks belong to the frame. Otherwise, a search window (5×5 blocks) around this block is opened and the search process started, from nearest to farthest, to find the best block show minimum Dif_0 and take it as initial reference center blocks in the frame. The steps of this algorithm are similar to the proposed blind search except the prediction steps which are used to assess the initial motion vector. Depending on motion vectors (Δx, Δy) of the tested neighbor blocks, which lay within certain search window allocated around the tested block, the initial motion vector could be determined as the mean vector of the motion vectors of the neighbor.

With the existence of initial motion, the absolute difference between the mean descriptors of the tested block with the block belongs to reference, I, frame is taken:

$$Dif_0 = \left|M_{tested}(0, x_s, y_s) - M_{pool}(0, x_s' + \Delta x, ys' + \Delta y \quad (20)\right.$$

Note that when the location of tested block is in first part region (Fig. 4) then the opened search window is of type S_1. Here the used search window size must be similar to that used in TSS in order to produce more accurate prediction. For blocks in second part region (Fig. 4) the search window is of type S_2 (a block has three surrounding neighbor blocks that have predetermined motion vectors); and the search window is of type S_3 (a block has numerous surrounding neighbor blocks that have predetermined motion vectors) when the tested block in the second part.

TESTS RESULTS

The significance of the proposed work lays in its flexibility to achieve a compromise between making good reduction in coding time while the video quality remains acceptable.

Numerous sets of tests have been conducted to examine and evaluate the performance of the proposed descriptor block matching algorithm. The used video test samples are Family and Foreman (with frame size specifications = 320×240 pixels and 352×288 pixels, respectively and the pixel color depth = 24 bit). The programs have been developed using C# programming language. The adopted coding parameters are: (i) block size 8x8 and (ii) the size of Group of Pictures (GOP) are set 10 frames for all tested blocks. In all tests the first GOP was taken. The used performance measures for the proposed algorithms are: (i) the fidelity level of the reconstructed frames using the metric (MAE) and (ii) the overall search time (second) for the first GOP.

Table 1: The default values (with prediction and without prediction) of the control parameters

Parameter	Default value with prediction (IS)	Default value without prediction (BS)	Range
V_1	8	7	[1, 20]
V_2	8	5	[1, 20]
N_3	4	6	[1, 15]

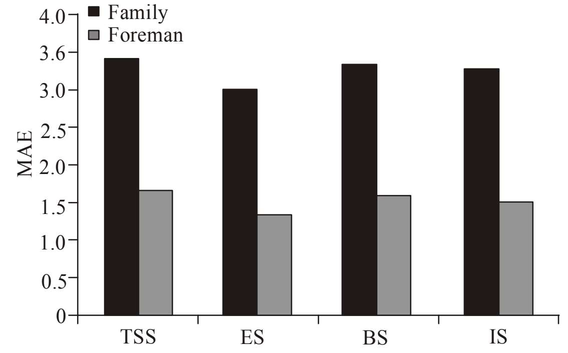

Fig. 5: The test results for comparing the performance of different search methods in term of error

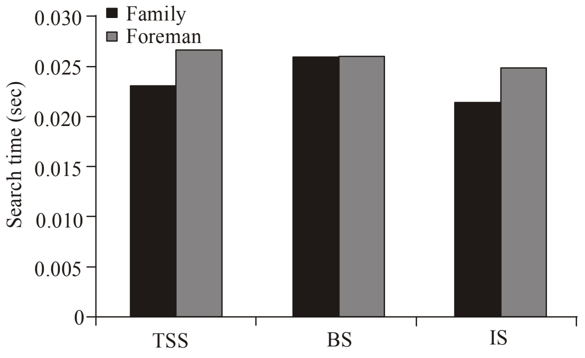

Fig. 6: The test results for comparing the performance of different search methods in terms of search time (sec)

Figure 5 and 6 present a comparison between the performance parameters (i.e., mean absolute error and the search time) of the Three Step Search (TSS), Exhaustive Search (ES), Blind-Search (BS), Intelligent-Search (IS). Figure 6 the search time for ES method was excluded because it is too high; it is 0.2468 sec (for Family) and 0.3106 sec (for Foreman). It is obvious that BS and IS methods perform better than the traditional methods in terms of reducing the error and search time. Figure 7 illustrates the results shows performance curve (i.e., MAE versus search time) for Intelligent-Search and Blind-Search methods. The results indicate that Intelligent-Search performs better than Blind-Search. So, the use of prediction in block descriptor is vital to give better results in MAE and search time.

Several parameters have been taken into consideration to study the performance of the suggested two algorithms. Some of these parameters have significant effect; for this reason they called "control parameters". The effects of the following control parameters have been investigated: (1) V_1, (2) V_2 and (3) N_3.

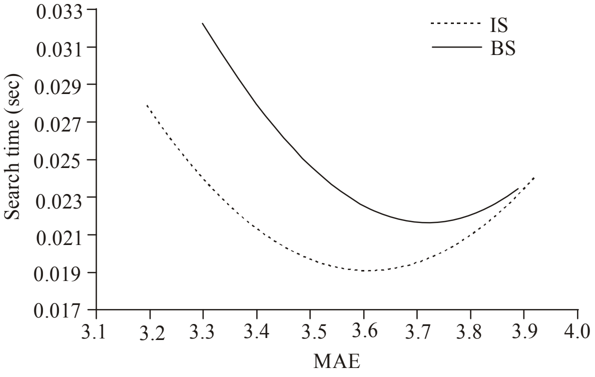

Fig. 7: Comparison between intelligent-search and blind-search

Table 1 presents the adopted default values of the considered control parameters, these values are selected after making a comprehensive tests and choosing the best value for each parameter. The effects of each parameter are examined by varying its value while setting other parameters fixed at their default values. The tests target is “low Search time and accepted quality” and the choice of default values was based on satisfying this test target.

Figure 8 presents the effects of V_1 on Time and MAE, Fig. 9 shows the effects of V_2 on Time and MAE and Fig. 10 presents the effects of N_3 parameter on Time and MAE. The tests of these parameters (V_1, V_2, N_3) have been conducted on Family video. It is apparent that these parameters cause significant effects. The results of all tests indicated that IS algorithm show better performance in terms of time and error reduction in comparison with BS.

The tests results indicated that the effects of other parameter are insignificant, for this reason these values are set fixed as listed in Table 2.

The parameter V_{sum} depends on following equation:

$$V_{sum} = (V_1 + 4V_4) * 1.1 \quad (20)$$

CONCLUSION

From the results of the proposed algorithm, the following remarks are stimulated:

- An improvement was introduced to the scheme of block matching by making use of blocks descriptors. The introduced mechanism replaces many of the pixel-wise block matching steps with hierarchal multi single value comparisons.

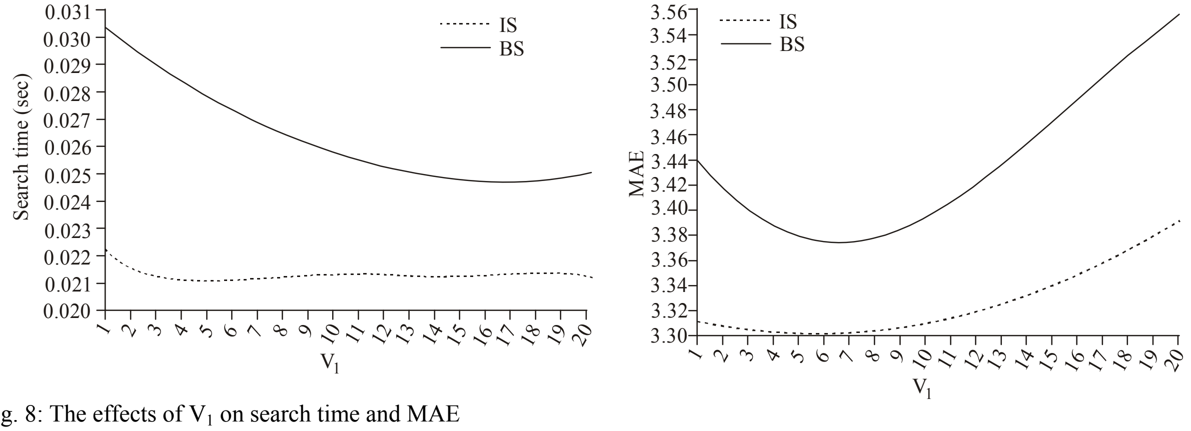

Fig. 8: The effects of V_1 on search time and MAE

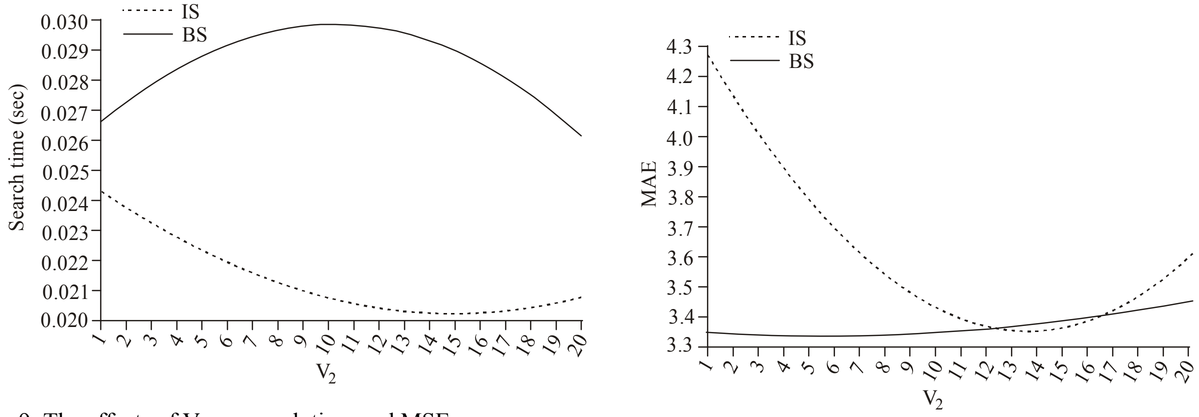

Fig. 9: The effects of V_2 on search time and MSE

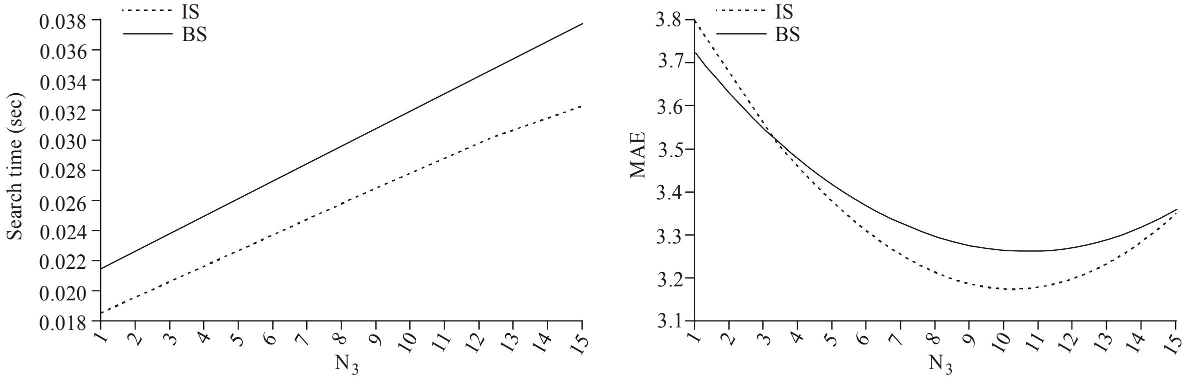

Fig.10: The effects of N_3 on search time and MAE

Table 2: The default values (with prediction and without prediction) of the fixed parameters

Parameter	Default value in IS	Default value in BS
V_3	21	21
V_4	24	24
V_5	10	
N_1	22	21
N_2	20	20
S_1	7	
S_2	5	
S_3	6	
S		8

- The use of two criteria mean and low order moments to derive descriptors values make the matching process faster e special when using prediction (Intelligent-Search).
- The conducted tests results show that the attained error is similar or close to the standard Exhaustive Search and the required time is close to that of TSS but with less number of blocks matching.
- In case of using Intelligent Search the relative search window can be reduced while keeping the fidelity level preserved.

As future work, the following steps could be taken:

- Using standard deviation as criteria to classify the blocks.

- A new method of predictive search can be applied.

REFERENCES

Ahmed, Z., A.J. Hussain and D. Al-Jumeily, 2011. Fast computations of full search block matching motion estimation (FCFS). IEEE T. Commun., pp: 1-6.

Aziz, T. and D.R.J. Dolly, 2012. Motion estimation and motion compensated video compression using DCT and DWT. Int. J. Emerg. Technol. Adv. Eng., 2(12): 667-671. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.414.5587&rep=rep1&type=pdf.

Bhavsar, D.D. and R.N. Gonawala, 2014. Three step search method for block matching algorithm. Proceeding of IRF International Conference. Pune, India, pp: 101-104.

Jagiwala, D.D. and S.N. Shah, 2012. Analysis of block matching algorithms for motion estimation in H.264 Video CODEC. Int. J. Eng. Res. Appl. (IJERA), 2(6): 1396-1401. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.415.4744&rep=rep1&type=pdf.

Jie-Rong, G. and L. Chang-Qing, 2011. Application of an improved motion estimated three step search algorithm in infrared video. Proc. Eng., 15: 2624-2628. http://www.sciencedirect.com/science/article/pii/S1877705811019941.

Khammar, M.R., 2012. Evaluation of different block matching algorithms to motion estimation. Int. J. VLSI Embed. Syst. (IJVES), 03(03): 148-153.

Kilthau, S.L., M.S. Drew and T. Möller, 2002. Full search content independent block matching based on the fast fourier transform. Proceeding of the IEEE International Conference on Image Processing, 1: 669-672.

Kiran, K.K. Sharma and S. Chauhan, 2014. A review on variants of block matching for motion detection in videos. Int. J. Res. Aspects Eng. Manage., 1(2): 73-75. http://emax.edu.in/journal/vol1-02/IJRAEM-01-02-21.pdf.

Kulkarni, S.M., D.S. Bormane and S.L. Nalbalwar, 2014. Coding of video sequences using block matching motion estimation three steps search algorithms. Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng., 2(6): 1651-1655. http://ijireeice.com/upload/2014/june/IJIREEICE3E%20s%20shamsundar_Coding_of_video.pdf.

Love, N.S. and C. Kamath, 2006. An empirical study of block matching techniques for the detection of moving objects. UCRL-TR-218038 Lawrence Livermore National Laboratory. https://e-reports-ext.llnl.gov/pdf/329054.pdf.

Manikandan, L.C. and R.K. Selvakumar, 2014. A new survey on block matching algorithms in video coding. Int. J. Eng. Res., 3(2): 121-125. http://www.ijer.in/ijer/publication/v3s2/IJER_2014_218.pdf.

Manjunatha, D.V. and Sainarayanan, 2011. Comparison and implementation of fast block matching motion estimation algorithms for video compresssion. Int. J. Eng. Sci. Technol. (IJEST), 3(10): 7608-7613. https: //www.idc-online.com/technical_references/pdfs/electronic_engineering/COMPARISON%20AND%20IMPLEMENTATION%20OF%20FAST%20BLOCK%20MATCHING%20MOTION%20ESTIMATION%20ALGORITHMS%20FOR%20VIDEO%20COMPRESSSION.pdf.

Reddy, V.S. and S. Sengupta, 2008. A fast predictive algorithm and architecture for block matching motion estimation. ICGST-GVIP J., 8(1): 9-16.

Vijay, B., G. Hegde and S. Sanju, 2014. Fast block-matching motion estimation using modified diamond search algorithm. Int. J. Adv. Comput. Eng. Communi. Technol. (IJACECT), 3(1): 1-6. http://www.irdindia.co.in/journal/journal_ijacect/pdf/vol3_iss1/1.pdf.