

# A Frequency Based Hierarchical Fast Search Block Matching Algorithm for Fast Video Communication

Nijad Al-Najdawi  
Al-Balqa Applied University  
Al Salt, Jordan

Sara Tedmori  
Princess Sumaya University for Technology  
Amman, Jordan

Omar A. Alzubi  
Al-Balqa Applied University  
Al Salt, Jordan

Osama Dorgham  
Al-Balqa Applied University  
Al Salt, Jordan

Jafar A. Alzubi  
Al-Balqa Applied University  
Al Salt, Jordan

**Abstract**—Numerous fast-search block motion estimation algorithms have been developed to circumvent the high computational cost required by the full-search algorithm. These techniques however often converge to a local minimum, which makes them subject to noise and matching errors. Hence, many spatial domain block matching algorithms have been developed in literature. These algorithms exploit the high correlation that exists between pixels inside each frame block. However, with the block transformed frequencies, block matching can be used to test the similarities between a subset of selected frequencies that correctly identify each block uniquely; therefore fewer comparisons are performed resulting in a considerable reduction in complexity. In this work, a two-level hierarchical fast search motion estimation algorithm is proposed in the frequency domain. This algorithm incorporates a novel search pattern at the top level of the hierarchy. The proposed hierarchical method for motion estimation not only produces consistent motion vectors within each large object, but also accurately estimates the motion of small objects with a substantial reduction in complexity when compared to other benchmark algorithms.

**Keywords**—Video coding; Frequency domain; Motion estimation; Hierarchical search; Block matching; Communication.

## I. INTRODUCTION

A moving video frame (image) is captured by taking a rectangular snapshot of the natural signal at periodic time intervals. Playing back the series of frames produces the appearance of motion. A higher temporal sampling rate (frame rate) gives a smoother playback, but requires more samples to be captured and stored. Most video coding methods utilize both temporal and spatial redundancy to compress video data [1]. In the temporal domain, there is usually a high correlation between frames captured at around the same time. Temporally adjacent frames are often highly correlated, especially if the temporal sampling rate is high. In the spatial domain, there is usually a high correlation between pixels (samples) that are close to each other. Thus, the values of neighbouring samples are often very similar [3]. In video compression, intra frame and inter frame coding are applied in order to reduce the number of bits needed to represent a video. In intra-frame coding, each frame is coded without any reference to other frames. This process involves transforming the block into the frequency domain,

where the resulting coefficients are quantized and encoded. A better compression may be achieved with inter-frame coding which exploits the temporal redundancy. In inter-frame coding, motion estimation and compensation (two vital processes within video coding) have become powerful techniques to eliminate the temporal redundancy due to high correlation between consecutive frames. Successive video frames may contain the same objects. Motion estimation is the process that describes the transformation from one image to another through examining 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 obtained to achieve data compression [4]. In a video scene, motion can be a complex combination of translation and rotation. Such motion is complicated to estimate and requires huge amount of processing. However, translational motion is simply estimated and has been used successfully for motion compensated coding. Most of the motion estimation algorithms make the following assumptions: objects move in translation in a plane that is parallel to the camera plane, i.e., the effects of camera zoom, and object rotations are not considered. Illumination is spatially and temporally uniform, and occlusion of one object by another, and uncovered background are neglected [5]. Several motion estimation approaches have been proposed, two of which are the pel-recursive algorithms (PRAs) and the block-matching algorithms (BMAs). In general, BMAs are more suitable for a simple hardware realization because of their regularity and simplicity. They estimate motion on the basis of rectangular blocks and produce one motion vector for each block. These algorithms assume that all the pels within a block have the same motion activity. PRAs involve more computational complexity and less regularity, so they are difficult to realize in hardware [3].

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 matched i.e., lowest distortion, candidate block is found and its displacement

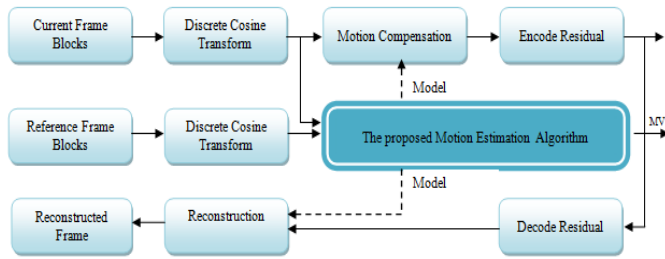


Fig. 1: Motion Estimation and Compensation in Block Diagram.

(motion vector) is recorded. In a typical inter-frame 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 inter-frame 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 required transmission bit rate is reduced [4]. Although the full-search motion estimation algorithm yields the best results, its intensive computation process limits its practical application. However, there is a trade-off between the complexity of the algorithm and the quality of the predicted frame. With this trade-off in consideration, many fast search motion estimation algorithms have been developed in literature. The fast search motion estimation algorithms can be classified mainly into two broad categories: spatial domain and frequency domain. The term spatial domain refers to the video frame plane itself, and approaches in this category are based on direct matching of pixels in successive video frames [2]. In the spatial domain, high correlation exists between pixels inside each frame block; therefore, the general block matching usually require measuring the similarities between every pair of pixels inside each block. Frequency domain motion estimation algorithms can be used to test the similarities between groups of frequencies which form a subset of the total frequencies in each block; therefore fewer comparisons can be considered for this task with a massive reduction in block matching calculations. Transforming a video frame into the frequency domain is a vital step that has to be performed in intra-frame coding. In this research, a new low complexity fast search motion estimation algorithm is proposed, as shown in Figure 1. The algorithm uses the intracoded frequency domain transformed frame in order to perform the proposed block matching technique. Section 2 provides an up to date literature review of both spatial and frequency domain motion estimation algorithms. Section 3 introduces the spatial-frequency transformation process. In section 4, the proposed matching technique is described. Section 5 provides the experimental results. Finally, section 6 concludes this research.

## II. LITERATURE REVIEW

Many sub-optimal spatial domain motion estimation algorithms have been proposed in literature such as the well-known: Cross-Search, Spiral-Search, Three-Steps-Search,

Two-Dimensional-Logarithmic-Search, Binary-Search, Four-Step-Search, Orthogonal-Search, and Diamond-Search algorithms. These algorithms are called sub-optimal because although they are computationally more efficient than the Full Search, they do not result in a quality that is as good as that of the Full Search algorithm [3]. A more recent variant of fast search motion estimation approaches may be found in [7][8][9][10][11][12][13][14]. The extensive variety of algorithms available for block-based motion estimation makes it difficult to choose between them. The choice depends on different criteria, such as: complexity, implementation, matching performance, rate-distortion performance, and scalability [15]. Motion estimation algorithms quality and performance has been a popular research area and different results have been obtained by different researchers. According to Kuhn et al. [16], the Three Step Search gives the best results. The five step diamond search performs well, but suffers in some cases from a too small search range of pixels. The hierarchical search algorithm depicted results which were not as good when compared with other algorithms. On the other hand, alternate pixel sub-sampling depicts very similar results as the original full search algorithm, where no extreme case of performance degradation occurs. According to Ghanbari [3] with regards to speed, the Two-Dimensional-Logarithmic algorithm outperforms the rest of the algorithms at the cost of quality. The Three-Steps-Search achieves a marginal improvement in terms of quality but has a high computational complexity in comparison with the Two-Dimensional-Logarithmic algorithm. The Four-Steps-Search algorithm outperforms the Three-Steps-Search algorithm in terms of complexity; however, its quality does not approach that of Full-Search as the hierarchical algorithms do. Although the complexity of the hierarchical algorithms is worst than some of other fast search algorithms, they outperform any other algorithm in terms of quality and they almost have the same quality as the Full-Search algorithms, with a significant reduction in complexity. Motion estimation in the frequency domain has been investigated by fewer researchers. Argyriou and Vlachos [17] proposed a motion estimation scheme for broadcast-quality digital video applications. The proposed scheme is based on the principle of gradient correlation in the frequency domain. The scheme involves the quad-tree decomposition of a frame. Quad-tree decompositions are obtained by using the motion compensated prediction error to control the partition of a parent block to four children quadrants. The partition criterion is applied iteratively until a target number of motion vectors or a target level of motion compensated prediction error is achieved or, until no more than a single motion component can be identified. Erdem et. al, [18] in their work model the discontinuous motion estimation problem in the frequency domain where the motion parameters are estimated using a harmonic retrieval approach. In the proposed work, the vertical and horizontal components of the motion are independently estimated from the locations of the peaks and they are paired to obtain the motion vectors using a specific procedure. L.Lucchese et al., [19] in their work introduced an alternative for 3-D motion estimation based on the Fourier transform of the 3-D intensity function described by the registered time-sequences of range and intensity data. The proposed system can lead to an unsupervised method for 3-D rigid motion estimation. This method has several advantages since it uses the total available information and not sets of features. Briassouli and Ahuja [20] in their work

analysed a video containing multiple objects in rotational and translational motion through a combination of spatial and frequency domain representations. It is argued that the combined analysis can take advantage of the strengths of both representations. Initial estimates of constant, as well as time-varying, translation and rotation velocities are obtained from frequency analysis. Improved motion estimates and motion segmentation for the case of translation are achieved by integrating spatial and Fourier domain information. For combined rotational and translational motions, the frequency representation is used for motion estimation, but only spatial information can be used to separate and extract the independently moving objects. The proposed algorithms are tested on synthetic and real videos. Tzimiropoulos et al., [21] proposed a frequency domain approach for the detection of symmetries in real images is presented. The framework is based on recent state-of-the-art research where motion estimation techniques are employed to sequentially determine all the associated parameters. In particular, the researchers introduce several modifications regarding the order of symmetry estimation and the detection of the axes of possible bilateral symmetry. Preliminary results demonstrate the efficiency of their approach. Pingault and Pellerin [22] describe a method to test motion transparency phenomena in image sequences based on an image sequence analysis in the frequency domain. It is mainly composed of a Stochastic-Expectation-Maximisation algorithm which provides a new statistical model for this problem. Young and Kingsbur [23] proposed a frequency-domain algorithm for motion estimation based on overlapped transforms of the image data. This method is developed as an alternative to block matching methods. The complex lapped transform is first defined by extending the lapped orthogonal transform to have complex basis functions. The complex lapped transform basis functions decay smoothly to zero at their end points, and overlap by 2:1 when a data sequence is transformed. A method for estimating cross-correlation functions in the complex lapped transform domain is developed. Block matching is subject to noise, therefore, researchers have attempted to use a predictor-corrector type estimator such as the Kalman Filter in order to enhance the motion vectors predictions and measurements and to obtain a better performance. The Kalman filter addresses the general problem of estimating the state of a discrete-time controlled process that is governed by the linear stochastic difference [24]. Various researches has been conducted in this field to incorporate Kalman filtering with block matching algorithms for the purpose of obtaining better motion vectors estimates such as the work in [25][26][27][28][29][30]. Although Hierarchical motion estimation algorithms (usually combines several block matching algorithms at different levels) are widely used in the spatial domain for their accuracy at extra complexity, those algorithms have not yet been investigated in the frequency domain. In this work, the authors propose a frequency based two-level hierarchical motion estimation algorithm that incorporates a novel searching method at the top-level of the hierarchy, with a matching criterion that reduces the complexity of the proposed method. The next section discusses the spatial-frequency transformation method used in this research.

### III. TRANSFORMATION FROM SPATIAL TO FREQUENCY DOMAIN

Video frames enclose high spatial and temporal correlation between adjacent pixels and consecutive frames respectively. Video compression involves reducing the spatio-temporal redundancy using intra-frame and inter-frame coding methods, in order to reduce the required number of bits that represent a video. The former process involves, transforming the block into the frequency domain, and quantizing the transformed coefficients in order to achieve compression. In the latter, further compression may be achieved by exploiting the temporal redundancy using motion estimation and compensation algorithms. In intra-frame coding the transformation process is used in order to represent the image data in another form, by switching from the spatial to the frequency domain or vice versa. The choice of transformation technique is governed by a number of criteria. However, regardless of the chosen transformation method, data in the transform domain should be separated into components with minimal inter-dependence. Moreover, any transformation method should be reversible and computationally tractable with low memory requirement and a low number of arithmetic operations [5]. Many transforms have been proposed for video coding, and the most popular transforms can be classified into two categories: block-based and frame-based transformations [31]. Although frame-based transformations are more suitable for images and give better decorrelation results, block-based methods are widely used in video coding and are more appropriate for this research, for the reason that motion estimation algorithms are based on block matching criteria which are based on matching portion of the frequency block in this research. The Discrete Cosine Transform (DCT) is chosen as the transformation method due to its accuracy and low complexity; DCT operates on  $B$ , a block of  $N \times N$  samples (pixels) and creates  $Z$ , an  $N \times N$  block of coefficients. A discrete cosine transform (DCT) expresses a sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is valuable for various applications in science and engineering. The use of cosine rather than sine functions is important in image and video applications as the sine functions lead to complex numbers and unnecessary complex computation. Specifically, a DCT is a Fourier-related transform that only uses real numbers. The most common variant of discrete cosine transform is the type-II DCT, which is often called "the DCT"; its inverse, the type-III DCT, is correspondingly often called "the inverse DCT" or "the IDCT". The action of the DCT (and its inverse, the IDCT) can be described in terms of a transform matrix  $W$  (see eq 1). The DCT of an  $N \times N$  sample block is given by: . And the inverse DCT (IDCT) is given by: , where  $B$  is a matrix of samples,  $Z$  is a matrix of coefficients, and are represented as in eq.1, and eq.2 respectively,

$$B_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x C_y Z_{xy} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2i+1)x\pi}{2N} \quad (1)$$

$$Z_{xy} = C_x C_y \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_{ij} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2i+1)x\pi}{2N} \quad (2)$$

W is an  $N \times N$  transform coefficients matrix, the elements of W are defined based on eq.3:

$$W_{ij} = C_i \cos \left[ \frac{(2j+1)i\pi}{2N} \right] \quad (3)$$

where  $c_i = \sqrt{\frac{1}{N}}$  for  $c_i = 0$  and  $c_i = \sqrt{\frac{2}{N}}$  for  $c_i \geq 0$ ;

The DCT Transformation matrix coefficients are image independent; they are always fixed for the same block size, and hence can be pre-computed and stored separately. The output of a two-dimensional DCT is a set of  $N \times N$  coefficients representing the image block data in the DCT domain and these coefficients can be considered as weights of a set of standard basis patterns [5]. The basis patterns for an 88 DCTs are composed of combinations of horizontal and vertical cosine functions. Any image block may be reconstructed by combining all  $N \times N$  basis patterns, with each basis multiplied by the appropriate weight. The result of the DCT transformation for a block in the spatial domain is a set of frequencies that are arranged in a zigzag ascending order. The frequency located at is the lowest frequency (highest wavelength) and is called the DC value. This value represents the general style of the block and is considered the most important frequency amongst all the other frequencies in the block. The rest of the frequencies range from low to high in a zigzag pattern and are called the AC values. The AC values contain the details of the block which ranges from general to fine details, as we progress forward in the zigzag order. For the purpose of this research, video frames are intra-coded using 4x4 and 8x8 DCT transformation block sizes at different levels of the proposed hierarchy. Further, selected frequencies are used in the block matching algorithm in order to obtain the best match as will be illustrated in the next sections.

#### IV. MATCHING CRITERION

The matching criterion has a huge impact on the performance of the algorithm. When comparing algorithms, different criteria should be investigated such as the well-known Mean Square Error (MSE), the Mean Absolute Difference (MAD), and the Sum of Absolute Difference (SAD). In addition to those standard criteria, other specific criteria were introduced by researchers such as: the Reduced Bit Mean Average Difference, the Min/Max-Error, and the Different Pixel Count [5]. The nonnegative matching error function (Sum of absolute differences as shown in eq.4) is normally defined over all the positions to be searched.

$$D_{m,n} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |\mu - \gamma| \quad (4)$$

where  $\mu = f_t(r+x, s+y)$  is the current frame reference block of its upper left pixel at the coordinate  $(r, s)$  and its lower right pixel at coordinate  $(r+x, s+y)$ ,  $\gamma = f_{t-1}(r+m+x, s+n+y)$  is a candidate block in the previous frame, and  $-W \leq m, n \leq W$  ( $W$  is the window size). The matching criterion has an enormous impact on the performance of the algorithm, therefore, reducing the number of required computations negatively affects the matching results when applied in the spatial domain since the pixels are highly correlated and it is impossible to

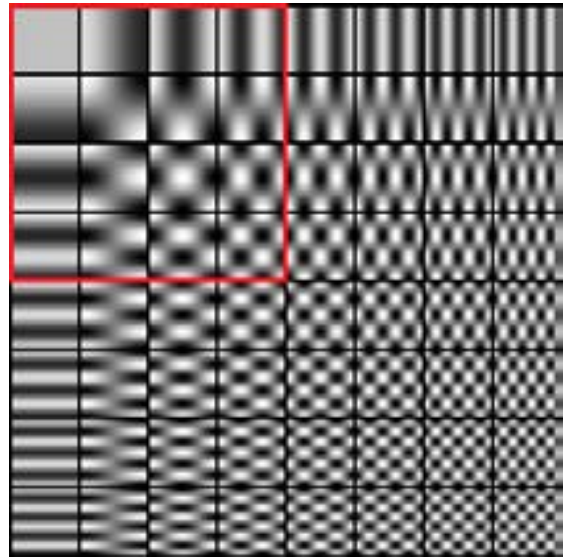


Fig. 2: Basis functions of an 8x8 DCT block, the top left quarter of the block is used in the matching criterion in the work, as frequencies in this quarter consists of a combination of low and reasonably high frequencies that represents the most important characteristics of the block.

differentiate between the significance of pixels inside a given block. However, reducing the number of required computations is possible in the frequency domain because the frequencies are highly de-correlated, making it possible to categorize frequencies based on their significance inside each transformed block. A simple method of block matching algorithm is the Full-Search Algorithm, where  $D_{m,n}$  is computed for all  $(2W+1)^2$  positions of candidate blocks in the search window. This results in  $(2W+1)^2 \times N^2$  subtractions,  $(2W+1)^2 \times (2N^2-1)$  additions and  $(2W+1)^2$  comparisons for each reference block. However, with fast search motion estimation algorithms, the SAD criterion shown in Eq. (1) requires  $N^2$  computations of subtractions with absolute values and  $N^2$  additions for each candidate block at each search position. The absence of multiplications makes this criterion computationally more attractive for real-time implementation. In this work the SAD criterion is used but with fewer numbers of computations. This approach requires  $N^2/4$  computations of subtractions with absolute values, and  $N^2/4$  additions for each candidate block at each search position. As stated earlier, the frequency coefficients produced by the DCT represent the basis functions to the source image, where the basis function increases as we move in a Zig-Zag pattern from the top-left to the bottom-right corners of the block. As shown in Fig. 2, the highest left, and the lower right corners' coefficients contain the lowest and the highest vertical and horizontal frequencies respectively. In this research, the first quarter of the transformed block is used in the matching criterion. Frequencies in this quarter consist of a combination of low and reasonably high frequencies, representing the most important characteristics of the block. As will be shown later, information in this portion of the block is sufficient to distinguish the desired block from amongst the rest of the neighbouring blocks that can be assumed as candidate locations for the search operation.

### A. THE PROPOSED HIERARCHICAL SEARCH MOTION ESTIMATION ALGORITHM

Hierarchical block matching techniques attempt to merge the advantages of large blocks with those of small blocks. The reliability of motion vectors is influenced by the selected block sizes. Larger blocks are more likely to track actual motion than smaller ones and thus are less likely to converge on local minima. Although such motion vectors are reliable, the quality of matches of large blocks is not as good as that of small blocks. Hierarchical block matching algorithms exploit the motion tracking capabilities of small blocks and use their motion vectors as starting points for searches for larger blocks. Normally, three level hierarchical searches are widely used in the spatial domain, where initially large blocks are matched and the resulting motion vector provides a starting point for a search for a smaller matching block. In this research a two-level hierarchy is used in the frequency domain, where a new search pattern is applied at the top of the hierarchy. The following summarises the steps of the proposed algorithm. This hierarchy is applied on both the previous and the current video frames:

- Step 1: the lowest level (level-1) consists of the video frame at its full resolution. This step involves sub-sampling level-1 by a factor of 2 in vertical and horizontal directions to produce level-2.
- Step 2: In this step, the frames at different levels (level-1) and (level-2) are transformed into the frequency domain using the two dimensional discrete cosine transform with different block sizes ( $4 \times 4$  block size at level-2 and  $8 \times 8$  at level-1). The search starts from the highest level (level-2) using block sizes, where the new proposed cross-diamond search pattern (described in the next section) is used to get a coarse motion vector that will be passed to level-1 (lowest level).
- Step 3: In this step, the Enhanced Three-Step-Search algorithm (described in section 3.3) is used on level-1 utilizing  $8 \times 8$  block sizes, to get the final motion vector that will be added to the previous image to get the next predicted image frame.

### B. THE PROPOSED CROSS-DIAMOND SEARCH PATTERN

The steps of the proposed algorithm are applied on the two hierarchies (current frame and previous frames hierarchies) and the search pattern is applied between corresponding levels of the hierarchies. The steps of the algorithm can be summarized as follows (Figure-2 illustrates the proposed method):

- Step 1: This step involves setting the window size to  $2^N + 1$  where  $N$  is the number of levels in the hierarchical search (i.e.,  $N = 2$  in the proposed algorithm), and setting the step size to the standard  $2^N$  (i.e., step size= 4).
- Step 2: Starting at the center point location around the obtained coarse motion vector, this step involves searching the four points forming a diamond shape pattern. The best match will be passed to step-3 as the new center of search.

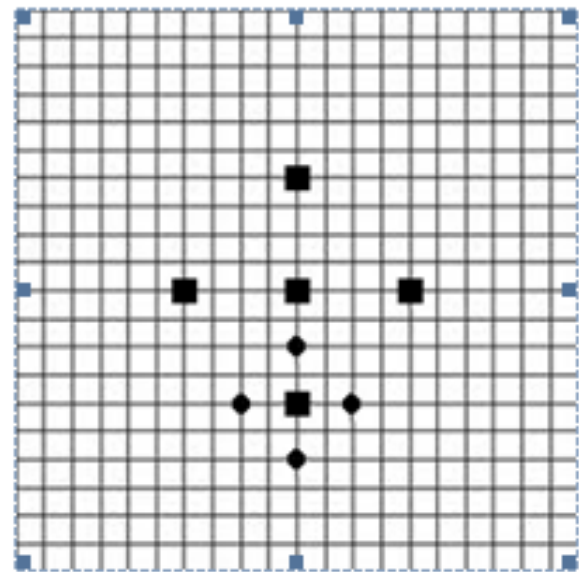


Fig. 3: The first step of the proposed algorithm involves four locations to be searched around the center forming a diamond shape pattern, the second step involves additional four locations to be searched around the best match point obtained from the first step with step size reduced to the half.

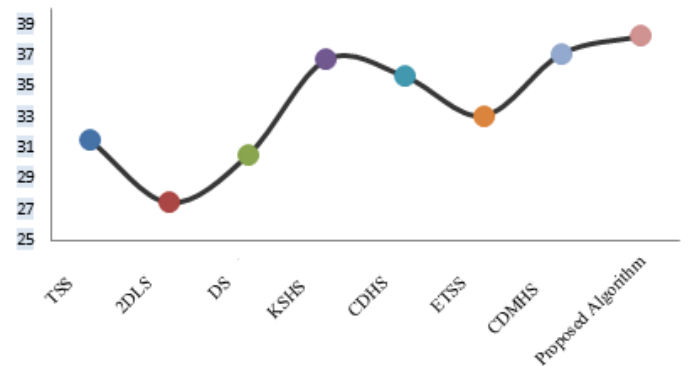


Fig. 4: Visual representation of PSNR values presented in Table-1

- Step 3: This step involves setting the step size to  $N/2$ , searching the four neighboring points around the new center obtained from step-2, and forming the diamond shape (see Fig. 3). If the step size  $> 1$ , then the step size is set to  $N/2$  and step-2 is repeated; otherwise, the best match point that is found is passed to level-1 of the hierarchy. The best matched block will be used to obtain the resulting motion vector and will be passed to the lower level (level-1), where it will be used as the centre point of search for the ETSS algorithm.

### C. THE ENHANCED THREE-STEP-SEARCH ALGORITHM (ETSS)

The obtained motion vector in the previous hierarchy is used as the centre point of the TSS algorithm (using  $8 \times 8$  block sizes). The TSS starts with 9 points to be checked (that

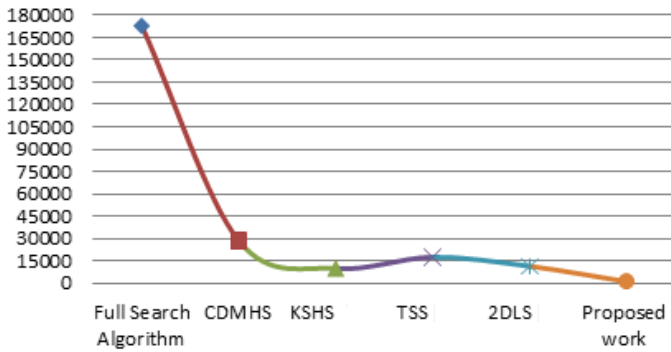


Fig. 5: Visual representation of the complexity of the proposed work compared to the rest of the standard and benchmark algorithms.

form a rectangular shape). The TSS is described as follows and is based on the following conditions:

- Condition 1: If the best match is the centre of the search window, the algorithm stops, and the same motion vector (obtained from the previous hierarchy) is considered as a final motion vector for the current block.
- Condition 2: If the best match is one of the eight rectangular neighbouring points, then the benchmark Three-Step-Search algorithm is performed based on the following criteria:
  - Search the location around the best match, and set the step size to  $S = 2^{N-1}$ .
  - Search the eight locations  $+/- S$  surrounding the location centre.
  - Reduce the step size to  $S = S/2$  and then go back to step 2.
  - Terminate when  $S = 1$

The number of comparisons required to find the best match is  $8N + 1$  for a search area of  $+/- 2N - 1$  pixels in N-Step Search algorithms. Since  $N = 3$ , the required computations are 25.

## V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this research, 13 standard Quarter Common Intermediate File (QCIF) and Common Intermediate File (CIF) video sequences of different motion contents are used to compare the performance of different algorithms. These video sequences are categorized into three classes; Class A, Class B, and Class C, with increasing motion complexity. This means that the video sequences in Class A have slow motion activities, those in Class B have medium motion activities and those in Class C have high or complex motion activities. The video sequences of Silent, Claire, Mother and Daughter belong to the Class A category. The video sequences of News, Suzie, Miss America, and Hall monitor are of moderate motion thus categorised as class B. Finally, the video sequences of Foreman, Carphone, Salesman, Flower, Coastgard, and Akiyo sequence which have fast object translation with high motion activity belong to Class C. More than 650 video frames of standard test video sequences with different formats were used

in the experiments. These comprise of the first 50 frames from each of the 13 test video sequences listed in Table-2. The results are evaluated subjectively and objectively. The PSNR (Peak signal to Noise Ratio) is used to objectively evaluate the system performance, where  $PSNR = 10 \log_{10} (L^2/MSE)$  is measured in decibel units (db units), where  $L$  is the range of pixel values (when the luminance component is only used  $L = 255$ ), and  $MSE = 1/N \sum_{i=1}^N (x_i - y_i)^2$  is the Mean Square Error, where  $N$  is the number of the pixels per frame, and  $x_i, y_i$  are the pixels within the original and predicted frames, respectively. A standard measurement states that, if the PSNR result is larger than 30db, then the difference between the original image and the resulted processed image will not be recognized through the human visual system. The higher the PSNR, the better quality it represents. Using the original and reconstructed frames, Table-2, illustrates the PSNR values for the proposed algorithm and compares the results with those of other benchmark and standard algorithms. The results in Table-2 show that, using the standard set of test videos, the proposed algorithm outperforms the standard Three-Step-Search [32] with 17% average enhancement, Two-Dimensional-Logarithmic-Search [33] with 28.6% average enhancement, and the Diamond Search algorithm [34] with 20.2% average enhancement. The average PSNR shows an enhancement of 16db units in some particular cases, signifying an enormous enhancement of quality.

In addition to the above standard algorithms, the enhanced Three-Step-Search algorithm [35], the Kalman simplified hierarchical search algorithm [36], and the Cross-Diamond Modified Hierarchical Search Algorithm [37] are chosen as the state-of-the-art benchmarks in the field of hierarchical search algorithms. When compared with the proposed work, using the same set of test videos, the average PSNR results show that the current proposed algorithm outperforms the work in [35][36], and [37] with 13.49%, 4% and 3% average enhancement respectively. The results of the PSNR values of the proposed work can be improved if the Kalman filter is applied as a stochastic predictor/ corrector estimator. Unfortunately, this will add to the complexity of the proposed work. Even without the use of an additional set of filters, the proposed algorithm has results comparable to those of the full search algorithm. Fig.5, visually illustrates the significant quality enhancement of the proposed work when compared with the rest of the benchmark and standard algorithms. In addition to the objective evaluation, a subjective evaluation of the proposed work can be seen in Fig.6-9, which illustrate visual representations of the reconstructed frames resulted from the proposed HS algorithm when applied to the set of standard videos with different class categories. The reconstructed frames in Fig.6 belong to class A with low motion complexity. The reconstructed frames in Fig.7 belong to class B with moderate motion complexity. Finally, the reconstructed frames in Fig. 8 and Fig. 9 belong to class C with high motion complexity.

The complexity of the proposed algorithm is evaluated and compared against some of the benchmark searching methods. Table-2 shows that the proposed algorithm outperforms the Full Search with a lower number of operations per block. Compared the FSA, the proposed algorithm requires 0.67% of the total number of additions, 0.7% of the total absolute differences, and 13.7% of the total number of comparisons. This can be summarized with a total Number of Operations

TABLE I: PSNR values of the proposed work, compared to standard and benchmark algorithms.

Video Sequence	TSS [32]	2DLS[33]	DS [34]	KSHS [36]	ETSS [35]	CDMHS [37]	Proposed Algorithm
Akiyo	32.74	30.74	31.23	34.92	33.31	34.89	35.61
Carphone	29.43	26.73	28.32	33.32	30.12	33.61	35.23
Claire	33.87	25.23	32.78	36.14	33.94	36.47	38.49
Coastguard	30.61	28.45	31.82	44.83	33.81	45.56	46.98
Flower	31.72	28.12	30.14	34.22	32.34	34.24	34.47
Foreman	29.26	25.80	27.96	33.79	30.04	33.91	35.71
Hall Monitor	34.81	30.11	32.19	41.81	36.81	41.73	42.34
Miss America	32.19	27.72	31.90	36.60	32.38	36.85	37.29
Mother and Daughter	31.20	28.98	28.02	38.36	34.62	38.78	39.84
News	29.50	26.29	30.74	36.31	31.71	37.64	39.21
Salesman	31.7	24.22	32.58	34.32	34.23	35.47	36.43
Silent	28.60	25.27	28.35	34.78	31.44	35.29	36.64
Suzie	33.54	29.17	30.49	37.79	35.32	37.95	38.83
<b>Average PSNR</b>	<b>31.47</b>	<b>27.45</b>	<b>30.50</b>	<b>36.71</b>	<b>33.08</b>	<b>37.11</b>	<b>38.24</b>



Fig. 6: Samples of reconstructed frames from Class A with Slow motion activities video sequences.:

- a Reconstructed Silent video, frame number 43.
- b Reconstructed Claire video, frame number 31.
- c Reconstructed Mother and Daughter video, frame number 16.



Fig. 7: Samples of reconstructed frames from Class B with moderate motion activities video sequences:

- a Reconstructed Suzie video, frame number 49.
- b Reconstructed News video, frame number 25.
- c Reconstructed Hall video, frame number 49.

Per Block (NOPB), where the proposed algorithm requires only 0.69% of the total NOPB required by the FSA (i.e., 99.31% reduction in complexity). When compared to the rest of the algorithms, the algorithm require 4.1% of the total complexity required by the Cross-Diamond Modified Hierarchical Search algorithm [37], 12% compared to Kalman Simplified HSA [36], 6.8% compared to the TSS algorithm [32], and 10.4% compared to the 2DLS algorithms [33]. This enormous reduction in complexity is due the substantial reduction in the total number of operations required in the proposed matching criterion. Fig. 4 visually illustrates the significant complexity reduction of the proposed work compared to the rest of the benchmark and standard algorithms.

## VI. CONCLUSION

Digital videos require a large amount of bandwidth for transmission or storage. Therefore, researchers have attempted to develop algorithms that compress video data whilst maintaining the highest quality possible. Motion estimation with block matching algorithms has proven to be effective in the reduction of video bit-rates while preserving the good quality. Block matching algorithms involve searching for block movements between consecutive video frames in the spatial domain. Hence, various fast searching algorithms have been investigated, each aiming at reducing the number of comparisons. However in the spatial domain, the high correlation that exists between pixels inside each frame block forces testing the similarities between every pair of pixels inside each block. In

TABLE II: The Total Number of Operations per Block (NOPB) required by the proposed algorithm, compared to the benchmark algorithms.

Algorithm	Addition	Multiplication	Absolute Difference	Comparisons	NOPB
Full Search Algorithm	$(2w + 1)^2 \times (2N^2 - 1)$ $w = 7, N = 16$ =114975	Null 0	$(2w + 1)^2 \times (N^2)$ $w = 7, N = 16$ =57600	$(2w + 1)^2$ $w = 7$ =255	<b>=172800</b>
Cross-Diamond MHS [37]	level(3) $(2w + 1)^2 \times (2N^2 - 1)$ $w = 3, N = 4$ =21421 level(2) $8 \times (2N^2 - 1), N = 8$ level(1) $23 \times (2N^2 - 1), N = 16$	Null 0	level(3) $(2w + 1)^2 \times (N^2)$ $w = 3, N = 4$ =7184 level(2) $8 \times (N^2), N = 8$ level(1) $23 \times (N^2), N = 8$	level(3) $(2w + 1)^2$ $w = 3$ level(2) $= 8$ level(1) $= 23$	<b>=28685</b>
Kalman Simplified HS [36]	level(3) $(2w + 1)^2 \times (2N^2 - 1)$ $w = 3, N = 4$ =6633 level(2) $8 \times (2N^2 - 1), N = 8$ level(1) $8 \times (2N^2 - 1), N = 16$ Kalman Filter 10	10 10	level(3) $(2w + 1)^2 \times (N^2)$ $w = 3, N = 4$ =3323 level(2) $8 \times (N^2), N = 8$ level(1) $8 \times (N^2), N = 16$ Kalman Filter 15	level(3) $(2w + 1)^2$ $w = 3$ =65 level(2) $= 8$ level(1) $= 8$	<b>=10016</b>
TSS [32]	$P \times (2N^2 - 1)$ $P = 23, N = 16$ =11753	Null 0	$P \times (N^2)$ $P = 23, N = 16$ =5888	$P = 23$ =23	<b>=17664</b>
2DLS [33]	$P \times (2N^2 - 1)$ $P = 15, N = 16$ =7665	Null 0	$P \times (N^2)$ $P = 15, N = 16$ =3840	$P = 15$ =15	<b>=11520</b>
Proposed Work	level(2) $8 \times ((N^2/2) - 1), N = 4$ =769 level(1) $23 \times ((N^2/2) - 1), N = 8$	Null 0	level(2) $8 \times (N^2/4), N = 4$ =400 level(1) $23 \times (N^2/4), N = 8$	level(2) $= 8$ =31 level(1) $= 23$	<b>=1200</b>



Fig. 8: Samples of reconstructed frames from Class C with fast motion activities video sequences with complex activities: a Reconstructed Carphone video, frame number 46. b Reconstructed Coastguard video, frame number 10. c Reconstructed Salesman video, frame number 26.



Fig. 9: Samples of reconstructed frames from Class C with fast motion activities video sequences with complex activities: a Reconstructed Flower garden video, frame number 37. b Reconstructed Foreman video, frame number 46. c Reconstructed Akiyo video, frame number 37.

this work, video frames are intra-coded and transformed into the frequency domain, where block matching can be applied to test the similarities between a subset of selected frequencies that correctly identifies each block distinctively, this yield to a fewer number of required comparisons that reduces the algorithms complexity. In this work a two-level hierarchical fast search motion estimation algorithm is proposed in the frequency domain that incorporates a novel search pattern at the top level of the hierarchy. In terms of quality and matching performance, the proposed algorithm outperforms the

other benchmark algorithms with an enormous reduction in complexity.

#### REFERENCES

- [1] O. Alzubi, *An Empirical Study of Irregular AG Block Turbo Codes over Fading Channels*, Research Journal of Applied Sciences, Engineering and Technology, 11(12), (2015) 1329-1335.
- [2] O. Alzubi, *Performance Evaluation of AG Block Turbo Codes over Fading Channels Using BPSK*, ICEMIS '15 Proceedings of the The



- International Conference on Engineering (MIS 2015), Istanbul, Turkey, (2015) 36:1-36:6.
- [3] M. Ghanbari, *Video Coding: An Introduction to Standard Codecs*, Institution of Electrical Engineers, (1999).
- [4] I. Richardson, *Video Codec Design*, John Wiley & Sons, (2002).
- [5] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley & Sons, (2003).
- [6] J. A. Alzubi, O. A. Alzubi, and T. M. Chen, *A Forward Error Correction Based On Algebraic-Geometric Theory*, 1st ed. Springer International Publishing, (2014).
- [7] J. Cai, and W. Pan, *On fast and accurate block-based motion estimation algorithms using particle swarm optimization*, 3rd ed. Elsevier Info. Sci., (2012) 53-64.
- [8] E. Cuevasa, D. Zaldivara, M. Prez-Cisnerosa, H. Sossab, and V. Osunab, *Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC)*, Elsevier Appl. Soft Comput., 13(6), (2013) 3047-3059.
- [9] E. Cuevasa, D. Zaldivara, M. Prez-Cisnerosa, and D. Olivab, *Block-matching algorithm based on differential evolution for motion estimation*, Elsevier Eng. Appl. Artif. Intel., 26(1) (2013) 488-498.
- [10] Z. Cui, G. Jiang, S. Yang, and C. Wu, *A new fast motion estimation algorithm based on the loopepipolar constraint for multiview video coding*, Elsevier Signal Process. Image Commun., 27(2), (2012) 172-179.
- [11] J. Fabrizioa, S. Dubuissonb, and D. Brziatb, *Motion compensation based on tangent distance prediction for video compression*, Elsevier Signal Process. Image Commun., 27(2), (2012) 153171.
- [12] C. Je, and H-M. Park, *Optimized hierarchical block matching for fast and accurate image registration*, Elsevier Signal Process. Image Commun., 28(7), (2013) 779-791.
- [13] K. Lai, Y. Chan, C. Fu, and W. Siu, *Hybrid motion estimation scheme for secondary SP-frame coding using inter-frame correlation and FMO*, Elsevier Signal Process. Image Commun., 27(1), (2012) 115.
- [14] F. Di Martinoa, V. Loiaa, and S. Sessab, *Fuzzy transforms for compression and decompression of color videos*, Elsevier Info. Sci., 180 (20), (2010) 3914-3931.
- [15] S.N. Basah, A. Bab-Hadiashar, and R. Hoseinnezhad, *Conditions for motion-background segmentation using fundamental matrix*, IET Comput. Vis., 3(4), (2009) 189200.
- [16] P. Kuhn, G. Diebel, S. Herrmann, A. Keil, H. Mooshofer, A. Kaup, R. Mayer, and W. Stechele, *Complexity and PSNR-Comparison of several Fast Motion Estimation Algorithms for MPEG-4*, In: Proceedings of SPIE, (1998) 486489.
- [17] V. Argyriou, and T. Vlachos, *Quad-Tree Motion Estimation in the Frequency Domain Using Gradient Correlation*, IEEE Transactions on Multimedia, 9(6), (2007) 1147-1154.
- [18] C. Erdem, G. Karabulut, E. Yanmaz, and E. Anarim, *Motion estimation in the frequency domain using fuzzy c-planes clustering*, IEEE Transactions on Image Processing, 10(12), (2001) 1873-1879.
- [19] L. Lucchese, G. Doretto, and G. Cortelazzo, *Frequency domain estimation of 3-D rigid motion based on range and intensity data*, In Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, (1997) 107-112.
- [20] A. Briassouli, and N. Ahuja, *Integrated spatial and frequency domain 2D motion segmentation and estimation*, The Tenth IEEE International Conference on Computer Vision ICCV, (2005) 244-250.
- [21] G. Tzimiropoulos, V. Argyriou, and T. Stathaki, *Symmetry detection using frequency domain motion estimation techniques*, The IEEE International Conference on Acoustics, Speech and Signal Processing, (2008) 861-864.
- [22] M. Pingault, and D. Pellerin, *Motion estimation of transparent objects in the frequency domain*, Harlow, England: Addison-Wesley, Journal of signal processing, 84(8), (2004).
- [23] R. Young, and N. Kingsbury, *Frequency-domain motion estimation using a complex lapped transform*, IEEE Transactions on Image Processing, 2(1), (1993) 2-17.
- [24] G. Torres, *3-D Kalman Filter for Image Motion Estimation*, 3rd ed. ACM Trans. Math. Softw. 37(2010) 1-16.
- [25] J. Kim, and J. Woods, *A Guide to  $\mathcal{B}T\mathcal{E}X$* , 3rd ed. IEEE Trans. Image Process. 3, (1998) 42-52.
- [26] C. Kuo, C. Chao, and C. Hsieh, *A new motion estimation algorithm for video coding using adaptive Kalman filter*, 3rd ed. J. Real-Time Imaging., 8, (2002) 387-398.
- [27] C. Kuo, C. Chao, and C. Hsieh, *An Efficient Motion Estimation Algorithm for Video Coding Using Kalman Filter*, 3rd ed. J. Real-Time Imaging. 8(2002) 253-264.
- [28] C. Kuo, S. Chung, and P. Shih, *Kalman filtering-based rate-constrained motion estimation for very low bit rate video coding*, IEEE Trans. Circuits Syst. Video Technol. 16, (2006) 3-18.
- [29] C. Kuo, C. Hsieh, and C. Chao, *Multiresolution Video Coding Based on Kalman Filtering Motion Estimation*, J. Visual Commun. Image Represent. 13, (2002) 348-362.
- [30] A. Sholiyi, J. A. Alzubi, O. A. Alzubi, O. Almomani, and T. O'Farrell, *Near Capacity Irregular Turbo Code*, Indian Journal of Science and Technology, 8 (23), (2015).
- [31] R. Gonzales, and R. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall (2008).
- [32] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, *Motion compensated interframe coding for video conferencing*, Proc. Nut. Telecommun. Conf. (1981).
- [33] J. R. Jain, and A. K. Jain, *Displacement measurement and its application in interframe image coding*, IEEE Trans. Commun. 29(12), (1981) 1799-1808.
- [34] S. Zhu, and K. K. Ma, *A new diamond search algorithm for fast block matching motion Estimation*, IEEE Trans. Image Process. 9(2), (2000) 287-290.
- [35] X. Jing, and L. P. Chau, *An efficient three-step search algorithm for block motion estimation*, IEEE Trans. Multimedia 6(3), (2004) 435-438.
- [36] S. Tedmori, and N. Al-Najdawi, *Hierarchical stochastic fast search motion estimation algorithm*, IET Comput. Vis. 6(1), (2012) 21-28.
- [37] N. Al-Najdawi, M. N. Al-Najdawi, and S. Tedmori, *Employing a novel cross-diamond search in a modified hierarchical search motion estimation algorithm for video compression*, Elsevier Inform. Sci., 268, (2014) 425435.