

HIGH-PERFORMANCE BLOCK MATCHING ALGORITHM FOR HIGH BIT-RATE REAL-TIME VIDEO COMMUNICATION

Nijad Al-Najdawi

(Received: 22-Nov.-2017, Revised: 04-Jan.-2018 and 27-Jan.-2018, Accepted: 15-Feb.-2018)

ABSTRACT

Although the advancements in hardware solutions are growing exponentially along with the communication channels capacity, high quality video encoders for real-time applications are still considered an open area of research. The majority of researchers interested in video encoders target their investigations towards motion estimation and block matching algorithms. Many algorithms that aim to reduce the total number of required mathematical operations when compared to Full Search have been proposed. However, the results often converge to local minima and a significant amount of computations is still required. Therefore, in this research, a hierarchy-based block matching method that facilitates the transmission of high bit-rate videos over standard communication methods is proposed. The proposed algorithm is based on the frequency domain, where the algorithm examines the similarities between a chosen frequency subset, which significantly reduces the total number of comparisons and the total mathematical computations required per block.

KEYWORDS

Video Communication, Video Coding, Video Conferencing, Motion Estimation, Block Matching.

1. INTRODUCTION

Digital videos consist of successive frames sampled over a period of time. Those successive frames carry high data redundancy. Therefore, eliminating bits of redundant data can be extremely helpful in reducing the size of digital video and compressing the video. Several types of compression techniques have been proposed in the past few decades. Those compression techniques are classified as being either lossless or lossy. The former type is achieved by eliminating redundant bits and reproduces exact original dataset. The latter is achieved by eliminating least important bits and reproduces a similar copy that might be indistinguishable by the human visual system from the original. Lossy compression techniques achieve better compression and are more applicable to digital videos; while, lossless techniques are in their nature more applicable for digital images. Lossless image compression allows the use of human visual system limits, by producing data that is sufficient to be classified as "good enough". The latest compression standards have set the architectures for video codecs as consisting of the following basic blocks: prediction, transform and entropy coding. Prediction includes estimates for the position of a current block inside a video frame. Transform process converts a block of pixels into frequency domain. Entropy coding involves encoding video data into a compressed bit stream.

Usually, consecutive frames have the same still or moving objects, creating a high correlation between consecutive frames. Therefore, researchers have investigated the use of methods that examine the object movements in a video sequence in order to produce motion vectors that represent the estimated motion. On the other hand, those estimated vectors are forwarded to the proper motion compensation methods that use those vectors to simulate the object movement, achieving data compression. Motion estimation and compensation methods are considered the most important techniques that eliminate temporal redundancy in successive video frames. However, those techniques are more applicable to translational motion and still have their limits when applied to rotational motion which is difficult to estimate and requires other techniques for processing. Therefore, motion estimation algorithms usually

assume the following: objects movement is translational, illumination is uniform across spatial and temporal domains, occlusions of objects by others are neglected, and finally uncovered background is not to be considered.

Various methods for coding have been proposed for video compression. Those coding techniques include intra-frame and inter-frame coding which are used to minimize the total number of bits required to transmit or store videos. In intra-frame coding, each frame is separately coded and this type of coding includes: transformation quantization and frame encoding. Inter-frame coding investigates the temporal redundancy and is usually applied in video coding in order to achieve the actual compression. In this type of coding, motion estimation and compensation algorithms are normally applied to eliminate the temporal redundancy that exists between successive frames. Various motion estimation approaches were proposed in literature; amongst those approaches, block matching algorithms were proven to be more suitable because of their reliability and simplicity. Block matching algorithms are used to estimate the object's motion in successive frames on the basis of rectangular blocks. These algorithms assume that all the pixels within a block have the same motion behaviour [8].

In block matching algorithms, frames are divided into $N \times N$ blocks; where all blocks in the current frame are matched with candidate blocks within a search area (window) on the reference frame (considering that candidate blocks have a translation movement in other frames) and the displacement motion vector is recorded for the best matched candidate. In inter-frame coding, the motion vector and the residual frame (resulting from subtracting input frame from the prediction of the reference frame) are usually transmitted. At the receiver side, the decoder builds the frame difference and adds it to the reconstructed reference frame. Therefore, data compression is achieved by eliminating inter-frame redundancy. This demonstrates the fact that better prediction methods give smaller error signals and a reduced transmission bit-rate [19].

In this paper, in addition to the introduction section, section 2 provides an up-to-date literature review of motion estimation algorithms. Section 3 introduces the transformation process. In section 4, the proposed hierarchical search algorithm is described along with the proposed matching criterion. Section 5 provides the experimental results and analysis. Finally, section 6 concludes this research.

2. LITERATURE REVIEW

A large number of block matching algorithms have been proposed over the last decades, such as the traditional methods found in [2]-[5], [7], [11], [15]-[16] and [24]. Amongst the available block matching algorithms, full search leads to the best possible match of the block in the reference frame with a block in another frame by calculating the cost function at each possible location in the search window. The resulting motion compensated frame has the highest peak signal-to-noise ratio when compared to any other block matching algorithm. However, this is the most computationally extensive block matching algorithm [8].

Optimized block matching algorithms speed up the exhaustive search required by full search algorithms based on fixed search patterns. Researchers in this domain have investigated the use of many algorithms in order to enhance the traditional search algorithms. In [33], Diaz Cortes et al. proposed a block matching algorithm that combines harmony search with a fitness approximation model. The authors considered the motion vectors in search window as potential matches. The authors applied a fitness function in order to evaluate the matching quality of each motion vector in addition to a strategy to decide which motion vectors can be estimated amongst the rest of the motion vectors. In [34], the authors proposed a hierarchy-based motion estimation algorithm using Gaussian image pyramid and unidirectional estimates of motion vectors at the top level. In their work, the authors proposed the use of five candidates for each motion vector. At the bottom level of the hierarchy, the motion vectors are corrected based on the sum of absolute difference values of the blocks. Moreover, in their work, the unidirectional motion vectors are assigned to bidirectional motion vectors.

In [35], Abdelazim et al. proposed the use of cross search algorithm in the H.265 standard that deals with high-efficiency video coding. In their work, the authors proposed a speed optimization technique in the frequency domain phase-correlation that enables compressing the videos rapidly while maintaining the video quality. In [36], Jia and Ding proposed a fast sub-pixel motion estimation algorithm. In their work, the authors proposed a scheme to skip sub-pixel search process in smooth

prediction units. Moreover, the authors proposed a fast sub-pixel search algorithm based on texture direction analysis in order to reduce the computational complexity. In [37], the authors presented a low computational complexity systolic hardware architecture for full search block matching algorithm. In their work, the proposed architecture is based on one-bit transform-based full search algorithm. The proposed motion estimation hardware architecture performs full search for four macro-blocks in parallel, where the proposed architecture was implemented in VHDL. In [38], the authors presented a three-step searching method in order to estimate the motion vectors of high-resolution image sequences using low number of computations. The searching strategy of this algorithm is carried in three steps, where the first search is performed in the large areas, the second is performed in the adaptive directional search and the last is performed in the small area search.

In [39], Arora et al. proposed a dynamic zero motion pre-judgment technique along with an adaptive diamond pattern search-based algorithm in order to enhance the search efficiency and accuracy of motion estimation. The dynamic zero motion pre-judgment is used for early identification of the stationary blocks. However, for the rest of the stationary blocks, an initial search center is used which has a high probability to be near actual motion vector. The variable size diamond pattern is used to obtain the global minima. In [40], Kovacevic et al. presented a motion estimation technique that combines recursive block-matching and customized phase plane correlation. In [41], Kamble et al. developed an approach for video coding using a modified three-step search block matching algorithm and weighted finite automata coding. In their work, the proposed block matching algorithm is based on the combination of rectangular and hexagonal search patterns and is used to compute motion vectors. The proposed weighted finite automata are used for the coding with a focus on reducing the encoding time. In order to reduce the encoding time, the authors in [47] proposed another approach for fractal coding using the weighted finite automata. The authors of [42] proposed a motion estimation method for image stabilization, integrating the speeded up robust features algorithm, modified random sample consensus and the Kalman filter. The authors achieved video stabilization with filtered motion parameters using the modified adjacent frame compensation.

In [43], the authors presented an enhanced version of the dynamic pattern search algorithm by means of reducing the search point computation. In their work, the algorithm starts by identifying the stationary blocks; then, the search points within the search area were evaluated for minimum distortion. The proposed work has been compared with other techniques like full search, diamond search and hexagon search. In [44], the authors proposed a two-step approach for enhancing the accuracy of initial search center prediction that is applied in the H.264 standard, in order to improve the motion estimation speed in video encoding. In their work, candidate blocks are identified in the first step for initial search center prediction. In the next step, the search is refined to obtain best possible initial search center.

In [45], the authors presented a hybrid approach for motion estimation. The hybrid method combines the dynamic zero motion pre-judgment technique with the initial search centers technique. In their work, calculating the initial search centers shifts according to the process of zero motion pre-judgment. In [46], the authors analyzed various tools involved in fast motion estimation algorithms. Moreover, the authors proposed a number of improvements in order to achieve a fast hybrid algorithm.

However, fewer researchers have investigated applying motion estimation algorithms in the frequency domain, such as the work of Young and Kingsbur [22] who proposed an alternate block matching method by applying a motion estimation technique based on overlapped transforms. Argyriou and Vlachos [1] in their work, proposed the use of gradient correlation in the frequency domain. Edrem et al. [6] estimated the motion parameters using a harmonic retrieval approach. Tzimiropoulos et al. [21] proposed a method for detecting symmetries in real images in the frequency domain. In their approach, the authors used motion estimation techniques to sequentially determine associated parameters. Pingault and Pellerin [17] tested motion transparency phenomena in video sequences based on the frequency domain. Their method contains an algorithm that introduces a new statistical model.

Hierarchical motion estimation algorithms are widely used for their accuracy; where in such algorithms several searching methods at different levels are applied. These types of algorithms are widely used due to their accuracy. However, applying those algorithms in the frequency domain has not yet been investigated properly. In hierarchical block matching techniques, the reliability of motion

vectors is related to the block size, where large blocks are more likely to converge on local minima. Moreover, in such algorithms, the advantages of selecting large blocks with small blocks at different levels are combined. Various research topics on the hierarchical search algorithms have been tackled in literature [25]-[29], [31].

In this work, a motion estimation algorithm based on a two-level hierarchy is proposed with a new block matching criterion to be applied at both levels of the hierarchy, as can be seen in Figure 1. The next section introduces the transformation method applied in this research.

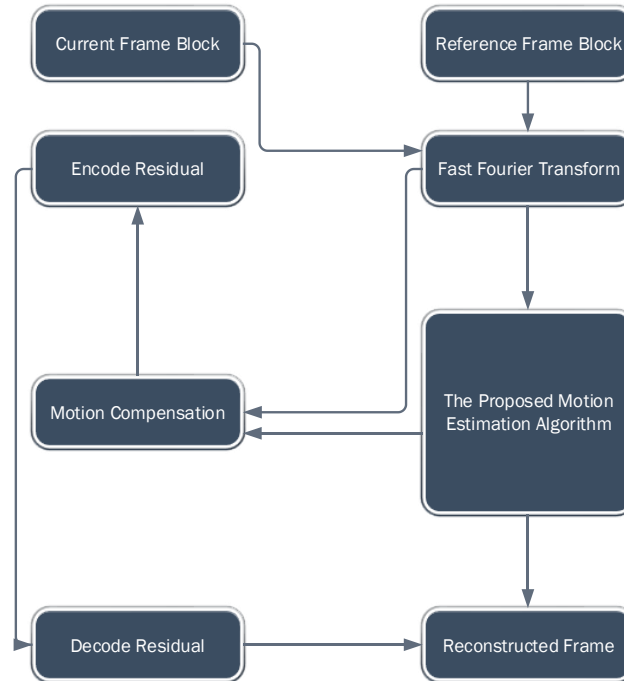


Figure 1. The architecture of general video encoders (the right side of the figure highlights the contribution in this research).

3. TRANSFORM DOMAIN

Video compression reduces the spatio-temporal redundancy that exists in the frame data and between consecutive frames using intra-frame and inter-frame coding methods. Intra-frame coding involves spatial to frequency transformation of the video frame and quantizing the frame frequencies by means of removing high frequencies that represent insignificant visual details in a given frame. Regardless of the transformation method that has been applied, it should be computationally acceptable and revertible [18]. In inter-frame coding, compression is achieved by utilizing the temporal redundancy using proper motion estimation and compensation algorithms. Various spatio-temporal transformation methods have been proposed in literature and are either image-based or block-based methods [9].

Block-based transformation methods are most applicable for use in video coding, since motion estimation algorithms are based on block matching methods. In this work, the Discrete Fourier Transform (DFT) is chosen, as it allows working in the frequency domain comparable to other transformation methods available in literature.

3.1 The Discrete Fourier Transform

Based on the Fourier theory, a complex signal can be decomposed into infinite series of cosine and sine terms and a group of coefficients that can be determined. The original function $f(t)$ can be decomposed into a series of basis states, based on (1).

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi nt}{T}\right) \quad (1)$$

where $a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(\frac{2\pi nt}{T}) dt$, $b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(\frac{2\pi nt}{T}) dt$.

The relationship above can be simplified as shown in (2)

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \exp(-i \frac{2\pi nt}{T}) \quad (2)$$

where $c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \exp(i \frac{2\pi nt}{T}) dt$, such that $c_0 = \frac{1}{2} a_0$, $c_n = \frac{1}{2} (a_n + ib_n)$ and $c_{-n} = \frac{1}{2} (a_n - ib_n)$.

In order to use Fourier transform with discrete input data such as the data available in digital videos and images, integrals are replaced by sums, T is replaced by N , $f(t)$ changes to x_n and c_n is replaced by X_n , which represents the Digital Fourier Transform shown in (3) and its inverse shown in (4). The DFT reveals periodicities in input data as well as the relative strengths of any periodic components [32].

$$X_n = \sum_{k=0}^{N-1} x_k \exp(i \frac{2\pi nk}{N}) \quad (3)$$

$$x_k = \frac{1}{N} \sum_{n=0}^{N-1} X_n \exp(-i \frac{2\pi nk}{N}) \quad (4)$$

Using (3), the N input samples (pixels) in a given block are converted into N frequency samples. The DFT is a coefficient matrix multiplication as shown in (5).

$$\begin{pmatrix} X_0 \\ \vdots \\ X_n \end{pmatrix} = \begin{pmatrix} W_{00} & \dots & W_{0n} \\ \vdots & \ddots & \vdots \\ W_{n0} & \dots & W_{nn} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix}, \quad W_{nk} = \exp\left(i \frac{2\pi nk}{N}\right) \quad (5)$$

The above calculation is of order N^2 . In order to reduce the DFT complexity, a number of researchers investigated the use of different patterns in W_{nk} . Amongst those approaches, the Fast Fourier Transform or FFT is proposed as a computational method in the order of $N \log N$. In this research, the Cooley–Tukey algorithm is used, as it is the most common FFT algorithm available in this domain [32] to transform the video frames with different block sizes at different levels of the two-level hierarchy. Moreover, only part of the frequencies is considered in the block matching criterion to get the best match as will be demonstrated in the next section.

4. THE PROPOSED HIERARCHICAL SEARCH ALGORITHM

Hierarchical block matching algorithms normally start the search process with small blocks and use their motion vectors as starting points to search for larger blocks in next hierarchies (the selected block sizes at each hierarchy affect the reliability of the produced motion vectors, where large blocks result in local minima.). Generally, in the spatial domain, three level hierarchical searches are used, as the data in its original form (pixel domain) is highly correlated. However, given the fact that data in the frequency domain is decorrelated, this facilitates the reduction of hierarchical levels needed to perform the matching process. Therefore, in this research, a two-level hierarchy in the frequency domain is used and proven to be sufficient. The proposed algorithm contains well-known algorithms in each level of the hierarchy with a new matching criterion (described in section 4.1.1) to be used at each level. The steps of the proposed algorithm are summarized in Algorithm-1 and visually represented in Figure 2.

4.1 The Proposed Matching Criterion

In order to compare algorithms in this domain, the standard Sum of Absolute Differences (SAD) shown in (6) is applied over all the possible searching positions.

$$SAD_{(m,n)} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |A - B| \quad (6)$$

Algorithm-1

Step 1: sub-sampling level-1 (the lowest level that consists of the video frame at its full resolution) by a factor of 2 in vertical and horizontal directions to produce level-2.

Step 2: transforming the frames at level-1 and level-2 into the frequency domain using the FFT with 4×4 block size at level-2 and 8×8 at level-1.

Step 3: the search process starts from level-2 with 4×4 block sizes, with the TSS search algorithm (described in section 4.1.2) to get a coarse motion vector that will be passed to level-1, based on the proposed matching criterion described in section 4.1.

Step 4: the two-dimensional logarithmic search algorithm (described in section 4.1.3) with 8×8 block sizes is applied, based on the proposed matching criterion described in section 4.1. in order to get the final motion vector.

Step 5: the resulting motion vectors from step-4 are added to the previous image in order to obtain the next predicted image frame.

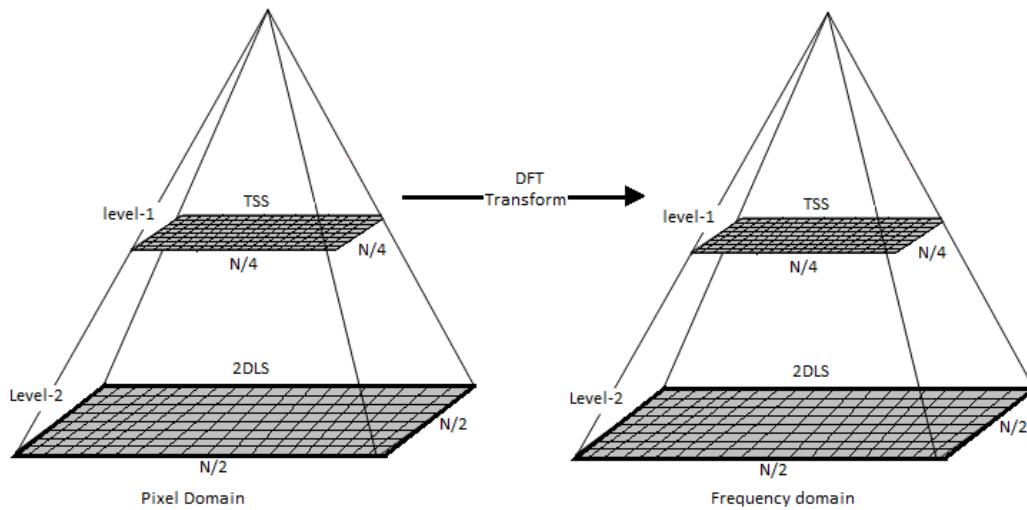


Figure 2. The proposed hierarchical search algorithm.

where $A = f_t(\lambda + x, \beta + y)$ is the block location coordinates, (λ, β) is the current block coordinates at the reference frame, $B = f_{t-1}(\lambda + m + x, \beta + n + y)$ is defined as the candidate block in the previous frame within the window size $-w \leq m, n \leq w$. The performance of the algorithm is highly dependent on the matching criterion. However, when applying the matching criterion in the spatial domain, the number of required computations cannot be reduced, as this will directly affect the matching results, since frame pixels are highly correlated. Therefore, in the frequency domain (where frame data is highly decorrelated), reducing the number of required computations is more appropriate.

In full-search algorithms, $SAD_{(m,n)}$ is computed at all $(2W+1)^2$ block positions within the search window. This results in a huge number of subtractions, additions and comparisons for each reference block. This massive number of computations can be reduced with fast search motion estimation techniques, where comparison criterion shown in (6) can be effectively reduced for search location.

The FFT produces frequency coefficients arranged in a pattern where the corners of coefficients block contain the lowest frequencies that describe the general vertical and horizontal information in the pixel block. However, the rest of the coefficients in the block include high frequencies that describe vertical and horizontal details in the pixel block. In this research, the coefficients at the four corners of the transformed block are only considered in the SAD matching criterion. Therefore, the total number of computations is reduced to a constant of 4 subtractions and 4 additions for each candidate block at each search position, instead of N^2 operations required by other algorithms in the spatial domain.

Information in these parts of the block is adequate to distinguish the desired block from the rest of the surrounding blocks as can be seen later in the experimental results section.

4.2 The TSS Algorithm

The steps of the TSS algorithm are applied at level-1 of the hierarchy in both of the current and previous frames as shown in Algorithm-2.

Algorithm-2

Step 1: Set the window size (W) to $2^N + 1$, where $N = 2$ (N= number of levels in the hierarchical search).

Step 2: Set the step size (S) to 2^N size, where $N = 2$.

Step 3: Start with search location at the center and apply the following:

- a) The eight locations at $\pm S$ around location (0, 0) are to be searched and the one with the minimum SAD is selected based on the matching criterion described in section 4.1.
- b) The search origin is set to the above selected location and the step size is reduced by a factor of 2
- c) The search repeats until $S = 1$ and the location with minimum SAD is considered as the best match in level-1.

Step 4: Pass the obtained coarse motion vector to level-2.

In the TSS algorithm, the total number of computations is reduced when compared to the full search algorithm by a factor of 9. Instead of evaluating 225 blocks, the TSS only evaluates 25 blocks.

4.3 The Two-Dimensional Logarithmic Search Algorithm

The two-dimensional logarithmic search algorithm is closely related to the three-step search algorithm. This algorithm requires more steps than the three-step search; however, it has a better accuracy. The two-dimensional logarithmic search algorithm is described in Algorithm-3:

Algorithm-3

Step 1: Set the window size to $4^N + 1$, $N = 2$ (N= number of levels in the hierarchical search).

Step 2: Set the step size to 4^N , where $N = 2$.

Step 3: Start with search location at the center and apply the following:

- a) Search the 4 locations at S distance from the center on the vertical and horizontal directions.
 - b) Amongst the searched locations, select the one with minimum SAD based on the matching criterion described in section 4.1.
 - c) If a point other than center has the minimum SAD, then this point is considered as the new center.
 - d) Repeat steps b and c
 - e) If the minimum SAD is given by the center point, then set $S = S/2$.
 - f) If $S = 1$, search the eight locations surrounding the center at S distance.
 - g) Set the motion vector as the point with the minimum SAD.
-

The resulting motion vectors from this step will be added to the previous frame in order to obtain the next predicted image frame. The TDLS algorithm is related to the TSS algorithm; however, it is used for a large search window size.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In order to test the efficiency of the proposed methods, two sets of standard testing videos are used in this research (shown in Table 1 and Table 2). The first set comprises a total of six standard videos of type CIF with an aspect ratio of 4:3. The CIF (Common Interchange Format) is a video format initially proposed in the H.261 standard. This video format is used to standardize the horizontal and vertical resolutions of YCbCr sequences in video signals. This type of video is used in standard video teleconferencing systems. CIF defines a video sequence with a resolution of 352×288 and a frame rate of 30 frames per second with color encoding using the YCbCr 4:2:0 standard, where the selected video sequences consist of 300 frames in each sequence. The second set of videos comprises a total of 3 High Definition (HD) videos (1080p) with an aspect ratio of 16:9 and color encoding using the YCbCr 4:4:4 standard, where the selected video sequences consist of 500 frames in each sequence.

The selected videos from both sets are well-known standard videos that are used to test the efficiency and compare the work with other benchmark algorithms. These video sequences from both sets are selected with increasing motion complexity ranging from slow to high motion complexity. More than 3300 video frames from the different sequences were used in the experiments and are listed in Table 1 and Table 2. The well-known PSNR is used to evaluate the proposed motion estimation algorithm performance. The $PSNR = 10 \log_{10}(L^2 / MSE)$, where L represents the range of pixel values. The Mean Square Error comparison criterion measures the similarity between the frame pixels and is measured as follows: $MSE = 1/N \sum_{i=1}^N (x_i - y_i)^2$, where N represents the number of pixels inside the frame, x_i, y_i represent the pixel value inside the original and predicted frames, respectively. High PSNR values indicate better quality. A PSNR result above 30dB means that changes caused by compression algorithms cannot be visually recognized.

Applying the PSNR between the original and reconstructed frames measures the efficiency of the proposed work. Therefore, Table 1 compares the obtained PSNR values of the proposed algorithm with those from other state-of-the-art algorithms in this domain using videos from the first set. Figure 3 and Figure 4 visually represent the results in Table 1. Table 2 shows and compares the PSNR results of the proposed work with those of state-of-the-art algorithms using HD videos (1080p) in the second set. Using the first set of test videos (CIF), and as can be seen in Table 1, the proposed work outperforms the standard three-step search [13], two-dimensional logarithmic search [10] and the diamond search algorithm [23] with 22%, 28% and 23% average enhancement, respectively. Moreover, when compared to the well-know KSHS algorithm [20], ETSS [12], CDMHS [14] and FHFS [30], the proposed algorithm outperforms those algorithms with 12%, 16%, 7% and 2%, respectively.

Using the HD (1080p) videos set, the proposed work outperforms the standard three-step search [13], two-dimensional logarithmic search [10] and the diamond search algorithm [23] with 12%, 26% and 24% average enhancement, respectively. Moreover, when compared to the well-know KSHS algorithm [20], ETSS [12], CDMHS [14] and FHFS [30], the proposed algorithm outperforms those algorithms with 11%, 10%, 9% and 5%, respectively. Figure 5 provides a visual representation of the average PSNR values shown in Table 1 and Table 2 and represents the results of the proposed algorithm compared to the rest of the state-of-the-art algorithms when applied to HD (1080p) high-resolution videos and normal CIF standard videos. Figure 6 shows samples of the reconstructed frames from the HD (1080p) videos listed in set 2. Figure 7, Figure 8 and Figure 9 provide samples of the reconstructed frames listed in set 1 that contains the standard CIF well-known videos.

The complexity of the proposed work is compared against the benchmark block-based motion estimation algorithms. Let w be the search window size, N represents the block size (hierarchy-based algorithms use various block sizes at each level of the hierarchy), the full search algorithm requires $(2w+1)^2 \times (2N^2 - 1)$ additions and $(2w+1)^2 \times N^2$ absolute differences, a. The cross diamond modified hierarchical search requires $(2w+1)^2 \times (2N^2 - 1)$ additions at level-1, $8 \times (2N^2 - 1)$ additions at level-2 and $23 \times (2N^2 - 1)$ additions at level-3; in addition to $(2w+1)^2 \times N^2$ absolute differences at level-1, $8 \times N^2$ and $8 \times N^2$ at level-2 and level-3, respectively. The Kalman simplified hierarchical search requires $(2w+1)^2 \times (2N^2 - 1)$, $8 \times (2N^2 - 1)$ and $8 \times (2N^2 - 1)$ at level-1, level-2 and level-3,

respectively. In terms of the number of absolute differences, the algorithm requires $(2w+1)^2 \times N^2$, $8 \times N^2$ and $8 \times N^2$ at each level, respectively. The proposed algorithm requires $15 \times ((N^2/2)-1)$ and $23 \times ((N^2/2)-1)$ additions at level-1 and level-2, respectively and $8 \times (N^2/4)$, $23 \times (N^2/4)$ comparisons at both levels. Using an appropriately sized search window (w) and an appropriate block size (N) for each algorithm, the proposed algorithm requires less than 1% of the total number of additions and the total number of absolute differences compared to the full search algorithm. When compared to the rest of the algorithms, the algorithm requires less than 5% of the total complexity required by the cross-diamond modified hierarchical search algorithm and less than 15% of the total complexity required by the Kalman simplified hierarchical search. This complexity reduction can be attributed to the substantial reduction in the total number of operations required in the proposed matching criterion.

Generally, in order to evaluate the use of motion estimation algorithms in the frequency domain, a performance comparison is conducted that evaluates the standard full search algorithm when applied in both pixel and frequency domains. Table 3 presents the resulted PSNR values of the full search algorithm implemented in both domains based on the standard set of HD (1080p) test videos (first 50 frames of each video are included in the test). As shown in this table (Table 3), the resulting average PSNR in the frequency domain is slightly better than that in the pixel domain. This small enhancement does not cover the cost of the extra complexity caused by the transformation process. However, it can achieve far better results (in terms of complexity reduction) when accompanied with proper search and matching techniques.

The direct implementation of Discrete Fourier Transform (DFT) requires $O(N^2)$ operations. However, when using Fast Fourier Transform (FFT), this can be reduced to $O(N \log(N))$, resulting in a substantial difference in the tractability of the DFT. The fact that the transition between the domains can be computed efficiently allows for more efficient implementations of the DFT.

Table 1. The resulting PSNR values of the proposed algorithm and the rest of the standard algorithms when applied to the standard set of CIF test video sequences from set-1.

Standard Videos (CIF 352x288)	TSS [13]	2DLS [10]	DS [23]	KSHS [2]	ETSS [12]	CDMHS [14]	FHFS [30]	Proposed work
<i>Akiyo</i>	31.9	29.5	30.8	34.2	32.6	36.3	35.2	36.4
<i>Mother and Daughter</i>	32.0	30.8	31.7	34.7	31.2	36.2	37.5	36.5
<i>News</i>	30.2	29.4	30.2	33.9	32.6	34.3	36.2	37.2
<i>Hall</i>	30.9	28.8	30.0	32.4	32.9	33.8	35.9	37.7
<i>Flower Garden</i>	28.3	28.0	29.2	30.8	30.5	32.4	36.1	36.8
<i>Football</i>	27.1	25.7	27.5	31.9	30.7	33.0	35.7	36.0
Average PSNR	30.1	28.7	29.9	33.0	31.8	34.3	36.1	36.8

Table 2. The resulting PSNR values of the proposed algorithm and the rest of the standard algorithms when applied to the standard set of HD test video sequences from set-2.

Standard Videos HD (1080p)	TSS [13]	2DLS [10]	DS [23]	KSHS [2]	ETSS [12]	CDMHS [14]	FHFS [30]	Proposed work
<i>Park_Joy</i>	42.8	39.1	39.8	40.5	43.2	43.4	42.1	43.8
<i>In-To-Tree</i>	38.6	35.9	36.1	42.8	39.6	32.9	44.6	45.2
<i>Station</i>	36.7	32.4	33.3	38.4	36.9	42.3	43.7	45.7
<i>Blue_Sky</i>	41.2	34.7	35.2	39.0	42.2	45.2	40.4	44.1
Average PSNR	39.8	35.5	36.1	40.2	40.1	41.0	42.7	44.7

Table 3. The PSNR values of the full search algorithm implemented in pixel and frequency domains based on the standard set of HD (1080p) test videos.

Full Search Algorithm Implementation	<i>Park_Joy</i>	<i>In-To-Tree</i>	<i>Station</i>	<i>Blue_Sky</i>	Average PSNR
<i>Pixel domain</i>	45.19	47.78	46.31	46.46	46.44
<i>Frequency domain</i>	46.24	46.93	45.70	47.81	46.67

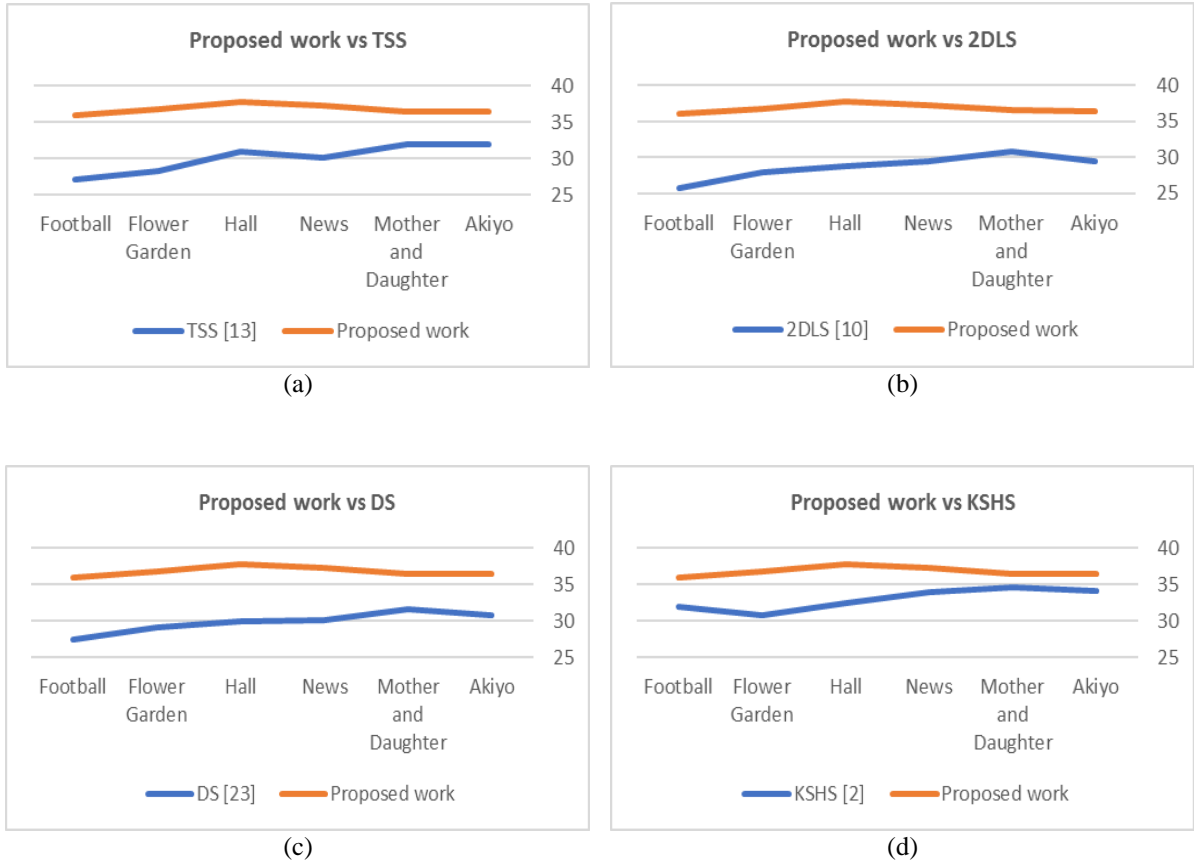
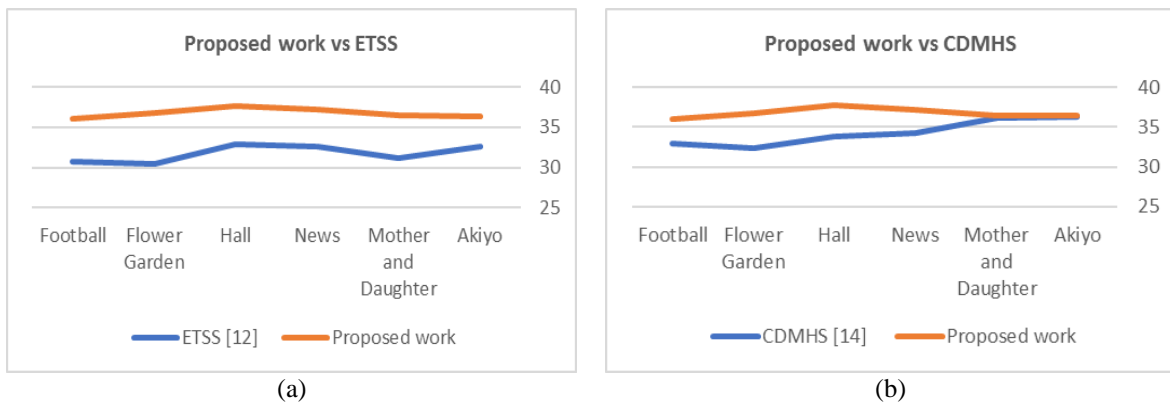
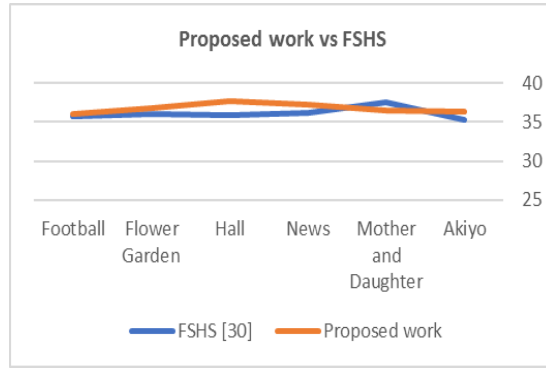


Figure 3. Visual representation that compares the proposed work with well-known algorithms as follows: (a) three-step search, (b) two-dimensional logarithmic search, (c) diamond search, (d) Kalman simplified hierarchical search.





(c)

Figure 4. Visual representation that compares the proposed work with well-known algorithms as follows: (a) enhanced three-step search, (b) cross diamond modified hierarchical search, (c) frequency-based fast hierarchical search.

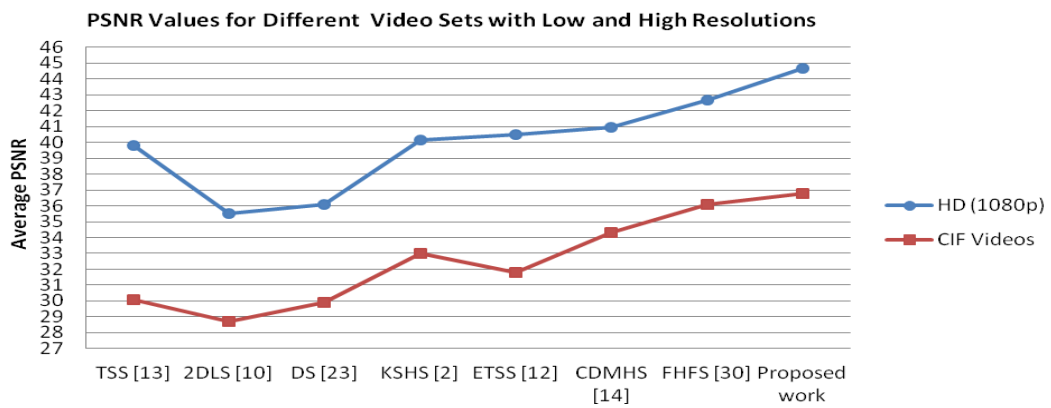


Figure 5. Visual representation of the average PSNR values shown in Table 1 and Table 2, that represents the results of the proposed algorithm compared to the rest of the state-of-the-art algorithms when applied to HD (1080p) high-resolution videos and normal CIF standard videos.



Figure 6. From top to bottom and from left to right, the reconstructed frames from "Park_Joy", "In-To-Tree", "Station" and "Blue_Sky" video sequences.



Figure 7. Sample of standard videos consisting of low motion activities, from left to right, reconstructed “Akiyo” and reconstructed “Mother and Daughter” video frames.



Figure 8. Sample of standard videos consisting of moderate motion activities, from left to right, reconstructed “News” and reconstructed “Hall” video frames.



Figure 9. Sample of standard videos consisting of high complex motion activities, from left to right, reconstructed “Flower Garden” and the reconstructed “Football” video, frames.

6. CONCLUSIONS

Digital videos consist of successive frames sampled over a period of time and carry high data redundancy. Digital video sizes can be massively reduced by eliminating redundant bits which can be achieved by proper compression methods. Various types of compression methods have been proposed in literature and during the last few years, many algorithms have been proposed to compress the massive amount of data available in digital videos while maintaining as much of the visual quality as possible. Motion estimation techniques based on block matching algorithms have been widely used for this purpose. In block matching techniques, each video frame is divided into blocks of similar sizes that contain frame pixels. Object movements successive video frames are searched and investigated on

block basis. In this work, block matching is applied in the frequency domain, where a group of carefully chosen frequencies that correctly identify each block distinctively is tested. The algorithm proposed in this research has reduced the total number of required operations, significantly reducing the algorithm's complexity. The proposed algorithm has been tested using standard test videos and has proven to outperform other state-of-the-art algorithms. Two sets of standard test videos were used in this work, the first set is comprised of the standard CIF videos and the other set is comprised of the standard HD (1080p) videos. Using the standard set of CIF videos, the proposed work outperforms the standard three-step search, two-dimensional logarithmic search and the diamond search algorithms with 22%, 28% and 23% average enhancement, respectively. Moreover, when compared to the well-know Kalman simplified hierarchical search algorithm, the enhanced three-step search algorithm, the cross diamond modified hierarchical search and the frequency-based Hierarchical fast search, the proposed algorithm outperforms those algorithms with 12%, 16%, 7% and 2%, respectively. Moreover, using the standard HD (1080p) videos set, the proposed work outperforms the standard three-step search, two-dimensional logarithmic search and the diamond search algorithms with 12%, 26% and 24% average enhancement, respectively. When compared to the well-know Kalman simplified hierarchical search algorithm, the enhanced three-step search algorithm, the cross diamond modified hierarchical search and the frequency-based hierarchical fast search, the proposed algorithm outperforms those algorithms with 11%, 10%, 9% and 5%, respectively. The complexity of the proposed work is compared against the benchmark block-based motion estimation algorithms. Results show that the proposed algorithm requires less than 1% of the total number of additions and the total number of absolute differences when compared to the full search algorithm. Moreover, when compared to the rest of the algorithms, the proposed work requires less than 5% of the total complexity required by the cross-diamond modified hierarchical search algorithm and less than 15% of the total complexity required by the Kalman simplified hierarchical search.

REFERENCES

- [1] V. Argyriou and T. Vlachos, "Quad-Tree Motion Estimation in the Frequency Domain Using Gradient Correlation," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1147-1154, 2007.
- [2] J. Cai and W. Pan, "On Fast and Accurate Block-based Motion Estimation Algorithms Using Particle Swarm Optimization," *Elsevier Information Sciences*, vol. 197, pp. 53-64, 2012.
- [3] E. Cuevasa, D. Zaldívar, M. Pérez-Cisneros and D. Oliva, "Block-matching Algorithm Based on Differential Evolution for Motion Estimation," *Elsevier Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 488-498, 2013.
- [4] E. Cuevasa, D. Zaldívar, M. Pérez-Cisneros, H. Sossa and V. Osuna, "Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC)," *Elsevier Applied Soft Computing*, vol. 3, no. 6, pp.3047-3059, 2013.
- [5] Z. Cui, G. Jiang, S. Yang and C. Wu, "A New Fast Motion Estimation Algorithm Based on the Loop-epipolar Constraint for Multi-View Video Coding," *Elsevier Signal Processing: Image Communication*, vol. 27, no. 2, pp. 172-179, 2012.
- [6] 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*, vol. 10, no. 2, pp. 1873-1879, 2001.
- [7] J. Fabrizio, S. Dubuisson and D. Béréziat, "Motion Compensation Based on Tangent Distance Prediction for Video Compression," *Elsevier Signal Processing: Image Communication*, vol. 27. no. 2, pp. 153-171, 2012.
- [8] M. Ghanbari, "Video Coding: An Introduction to Standard Codecs," *Institution of Electrical Engineers*, 1999.
- [9] R. Gonzales and R. Woods, *Digital Image Processing*, 3rd Edition, Prentice Hall, 2008.
- [10] J. R. Jain and A. K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Transactions on Communincations*, vol. 29, no. 12, pp. 1799-1808, 1981.
- [11] C. Je and H-M. Park, "Optimized Hierarchical Block Matching for Fast and Accurate Image Registration," *Elsevier Signal Processing: Image Communication*, vol. 28, no. 7, pp. 779-791, 2013.

- [12] X. Jing and L. P. Chau, "An Efficient Three-step Search Algorithm for Block Motion Estimation," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 435-438, 2004.
- [13] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conferencing," in: *Proc. Nat. Telecommunications Conference*, pp. G5.3.1-5.3.5, 1981.
- [14] 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 Information Sciences*, vol. 268, pp. 425-435, 2014.
- [15] 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 Processing: Image Communications*, vol. 27, no. 1, pp. 1-15, 2012.
- [16] F. Di Martino, V. Loia and S. Sessa, "Fuzzy Transforms for Compression and Decompression of Color Videos," *Elsevier Information Sciences*, vol. 180, no. 20, pp. 3914-3931, 2010.
- [17] M. Pingault and D. Pellerin, "Motion Estimation of Transparent Objects in the Frequency Domain," *Journal of Signal Processing*, vol. 4, no. 8, 2004.
- [18] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley & Sons, 2003.
- [19] I. Richardson, *Video Codec Design*, John Wiley & Sons, 2002.
- [20] S. Tedmori and N. Al-Najdawi, "Hierarchical Stochastic Fast Search Motion Estimation Algorithm," *IET Computer Vision*, vol. 6, no. 1, pp. 21-28, 2012.
- [21] G. Tzimiropoulos, V. Argyriou and T. Stathaki, "Symmetry Detection Using Frequency Domain Motion Estimation Techniques," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 861-864, 2008.
- [22] R. Young and N. Kingsbury, "Frequency-domain Motion Estimation Using a Complex Lapped Transform," *IEEE Transactions on Image Processing*, vol. 2, no. 1, pp. 2-17, 1993.
- [23] S. Zhu and K. K. Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287-290, 2000.
- [24] H. Alzoubi and W. Pan, "Fast and Accurate Global Motion Estimation Algorithm Using Pixel Subsampling," *Elsevier Information Sciences*, vol. 178, no. 17, pp. 3415-3425, 2008.
- [25] K. M. Nam, J. S. Kim, R. H. Park and Y. S. Shim, "A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 4, pp. 344-351, 1995.
- [26] C. Kuo, C. Hsieh and C. Chao, "Multiresolution Video Coding Based on Kalman Filtering Motion Estimation," *Elsevier Journal of Visual Communication and Image Representation*, vol. 13, no. 1, pp. 348-362, 2002.
- [27] B. C. Song and J. B. Ra, "A Hierarchical Block Matching Algorithm Using Partial Distortion Criterion," in: *Proc. SPIE 3309, Visual Communications and Image Processing*, pp. 88-95, 1998.
- [28] C. K. Cheung and L. M. Po, "A Hierarchical Block Motion Estimation Algorithm Using Partial Distortion Measure," in: *Proc. International Conference on Image Processing*, pp. 606-609, 1997.
- [29] S. Basah, A. Bab-Hadiashar and R. Hoseinnezhad, "Conditions for Motion-background Segmentation Using Fundamental Matrix," *IET Computer Vision*, vol. 3, no. 4, pp. 189-200, 2009.
- [30] N. Al-Najdawi, S. Tedmori, O. Alzubi, O. Dorgham and J. Alzubi, "A Frequency-Based Hierarchical Fast Search Block Matching Algorithm for Fast Video Communication," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 447-455, 2016.
- [31] Z. Weng, G. Chen, L. Shieh and J. Larsson, "Evolutionary Programming Kalman Filter," *Elsevier Information Sciences*, vol. 129, no 1-4, pp. 197-210, 2000.
- [32] R. Tolimieri, M. An and C. Lu, "Algorithms for Discrete Fourier Transform and Convolution," *Springer Science and Business Media*, New York, pp. 55-70, 1997.
- [33] MA. Diaz-Cortes, E. Cuevas and R. Rojas, "Motion Estimation Algorithm Using Block-Matching and Harmony Search Optimization," *Engineering Applications of Soft Computing, Intelligent Systems Reference Library*, vol. 129, 2017.
- [34] S. Yu and J. Jeong, "Multidirectional Motion Estimation Algorithm for Frame Rate Up-conversion,"

- 21st International Conference on Circuits, Systems, Communications and Computers, 2017.
- [35] A. Abdelazim, A. Hamza and D. Ait-Boudaoud, "Cross Search Frequency Domain Motion Estimation Algorithm for the High Efficiency Video Coding Standard," 9th IEEE-GCC Conference and Exhibition, 2017.
- [36] S. Jia and W. Ding, "A Fast Sub-Pixel Motion Estimation Algorithm for HEVC," IEEE International Symposium on Circuits and Systems, 2016.
- [37] P. Muralidhar and C. Rao, "High Performance Architecture of Motion Estimation Algorithm for Video Compression," Journal of Circuits, Systems and Computers, vol. 25, no. 8, 2016.
- [38] S. Chang and R. Wang, "Novel Motion Estimation Algorithm for Image Stabilizer," Engineering Computations, vol. 34, no. 1, pp. 77-89, 2017.
- [39] S. Arora, N. Rajpal and R. Purwar, "A New Fast Motion Estimation Algorithm Using Adaptive Size Diamond Pattern Search with Early Search Termination," International Journal of Computational Vision and Robotics, vol. 7, no. 6, pp. 623-643, 2017.
- [40] V. Kovacevic, Z. Pantic, A. Beric and R. Jakovljevic, "Block-Matching Correlation Motion Estimation for Frame-Rate Up-Conversion," Journal of Signal Processing Systems, vol. 84, no. 2, pp. 283-292, 2016.
- [41] S. Kamble, N. Thakur and P. Bajaj, "Modified Three-Step Search Block Matching Motion Estimation and Weighted Finite Automata-based Fractal Video Compression," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 4, no. 4, pp. 27-39, 2017.
- [42] X. Cheng, Q. Hao and M. Xie, "A Comprehensive Motion Estimation Technique for the Improvement of EIS Methods Based on the SURF Algorithm and Kalman Filter," Sensors Journal, vol. 16, no. 4, pp. 1-15, 2016.
- [43] R. Purwar, "Enhanced Dynamic Pattern Search Algorithm with Weighted Search Points for Fast Motion Estimation," Signal, Image and Video Processing, vol. 11, no. 6, pp. 1001-1007, 2017.
- [44] S. Arora, N. Rajpal, K. Khanna and R. Purwar, "Improved Accuracy in Initial Search Center Prediction to Fasten Motion Estimation in H.264/AVC," IETE Journal of Research, vol. 62, no. 6, pp. 842-851, 2016.
- [45] S. Arora, K. Khanna and N. Rajpal, "A Novel Hybrid Approach for Fast Block-based Motion Estimation," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 4, no. 6, pp. 24-30, 2017.
- [46] P. Nalluri, L. Alves and A. Navarro, "Complexity Reduction Methods for Fast Motion Estimation in HEVC," Image Communication Journal, vol. 39, pp. 280-292, 2015.
- [47] S. Kamble, N. Thakur and P. Bajaj, "Fractal Coding Based Video Compression Using Weighted Finite Automata," International Journal of Ambient Computing and Intelligence, vol. 9, no. 1, 2018.

ملخص البحث:

على الرغم من التطورات التي حدثت في مجال حلول معدات الحاسوب، ما زالت مُرَمِّزات الفيديو عالية الجودة بتطبيقات الزمن الحقيقي مجالاً مفتوحاً للبحث. فقد تم اقتراح العديد من الخوارزميات التي تهدف إلى التقليل من عدد العمليات الرياضية المطلوبة. وكانت النتائج تلقت عند قيم صغيرة محلية وتستمر الحاجة إلى إجراء عدد كبير من العمليات الحسابية. لهذا تقترح هذه الورقة طريقة مواءمة هرمية تسهل نقل الفيديوهات عالية التفاصيل بالمقارنة مع طرق الاتصال المعيارية. والجدير بالذكر أن الخوارزمية المقترحة مبنية على الحقل الترددي؛ إذ تفحص الخوارزمية التشابهات بين مجموعة فرعية مختارة من الترددات، مما يقلل إلى حد كبير من العدد الكلي من المقارنات والحسابات الرياضية الإجمالية المطلوبة لكل كتلة.