Adaptive Order Cross–Square–Hexagonal search and fuzzy tangential-weighted trade-off for H.264 in motion estimation

BACHU SRINIVAS^{1,*} and K MANJUNATHACHARI²

¹Department of Electronics and Communication Engineering, Guru Nanak Institutions Technical Campus, Hyderabad 501506, India

²Department of Electronics and Communication Engineering, GITAM Deemed To Be University, Hyderabad 502329, India
e-mail: srinivasbachu14@gmail.com

MS received 11 May 2017; revised 5 August 2017; accepted 7 August 2017; published online 10 March 2018

Abstract. In recent years, various algorithms have been proposed to attain low computational complexity in motion estimation of the image sequence coding based on block matching. This paper presents an Adaptive Order Cross–Square–Hexagon (AOCSH) search algorithm, which employs a smaller cross-shaped pattern before the first step of a square pattern and replaces the square-shaped pattern with the hexagon search patterns in subsequent steps. The proposed search patterns aid in finding the best matching block, without much consideration of the vast number of search points. Here, fuzzy tangent-weighted function is also proposed to evaluate the matching points using the rate and the distortion parameters. The proposed methods are effectively applied to the block estimation process to handle the objectives of visual quality and distortion. The performance of the proposed AOCSH approach is compared to the existing methods, such as AOSH, H.264 and elastic models, using the structural similarity index (SSIM) and the peak signal to noise ratio (PSNR). From the analysis, it can be seen that the proposed approach attains the maximum SSIM of 0.99 and maximum PSNR of 40.92 dB with reduced computation time of 3.28 s.

Keywords. Block motion estimation; square search; hexagon search; fuzzy tangent-weighted function.

1. Introduction

Mobile communication technologies are innovating day to day. The communication devices like, mobile phone, notebook, and many more devices have the ability to execute the complex multimedia applications. In multimedia [1, 2], video compression is one among the significant entities to be considered. For the portable devices to transfer the video with the ease of transmission as well as reception and less storage, the bandwidth of the selected video must be reduced. Video compression or video coding attains this objective. The video is a sequence of image frames that are related along the temporal and the spatial dimensions. When the camera or an object in the video sequence moves, the captured images also get altered. Therefore, a part of an image may appear at different, but close, locations, in the multiple consecutive video frames. Such an option will provide the leverage for compressing the video sequence using the motion estimation techniques. In motion estimation, the movement in the blocks of a frame is identified with a match to the reference frame at different and close locations. The video compression

reduces the data, which represent the video frames. Hence, video transmission with redundancy reduction of the data is a challenge. Later, its predictable properties are defined to predict the real data to be transmitted.

Algorithms for video compression are of several types. H.264 is a standard video coding algorithm, also called as the MPEG-4 Advanced Video Coding [3, 4]. The H.264 coding provides enough flexibility so that it can be applied to a wide variety of applications in several networks and system, which involve low or high bit rates, etc. The significance of the H.264 standard, when compared with the previous video coding algorithms [5–7], is the considerable advancement in the coding tools. Specifically, the deblocking filter and the intra-frame prediction are the added advancements lacking in the previous coding standards. The motion estimation process becomes computationally complex depending on the blocks used. When the blocks used for matching are of variable size and shape with respect to the reference frame, the motion movement prediction will be complex. The matching procedure is majorly concerned with the blocks' characteristics, matching the precision and the motion model.

Researchers have put efforts to predict the motion block that ends up in low computational complexity. The previous algorithms for motion block estimation are full search

*For correspondence

algorithm [8], three-step search algorithm, 2D-logarithmic search algorithm, orthogonal search algorithm, four-step search algorithm, block-based gradient descent search algorithm, etc. These search algorithms competently affect the block matching strategy with increased search speed and accuracy, while estimating the motion movement in the macro blocks. However, the video having high motion content suffers from reduced video quality [9]. The searching time increases with the number of search points, which is a considerable factor in the existing searching algorithm because of the increased number of searching points.

In this paper, an adaptive search algorithm that uses the cross search pattern, the square search pattern and the hexagon search pattern for block motion estimation is proposed. The cross search ascertains the direction for motion estimation. The square search utilizes the direction that is identified out of the cross search and later the hexagon search follows in relation to this order. Utilization of integrated search pattern reduces the search points, leading to an increase in the searching speed. Also, motion predictability can be attained by the proposed search algorithm. A fuzzy tangent-weighted function is proposed to evaluate the matching point from the search points. The rate, as well as the distortion parameters, is considered for the trade-off selection of the matching point with the minimum searching points. After the selection of the search point, the motion vectors are identified and subjected to transformation and quantization, and finally, Context-Adaptive Variable Length Coding (CAVLC) is applied to the bit stream.

The main contributions of the paper are as follows. (a) Adaptive Order Cross–Square–Hexagon (AOCSH) algorithm that integrates cross, square and hexagon searches. The cross search pattern allows the direction of the search to be identified through avoiding the search points in the wrong direction, while the square or the hexagon search pattern performs motion estimation. The order of search is changed to find the motion point with less searching time. (b) The fuzzy tangent-weighted function is proposed to evaluate the matching points using the rate and the distortion parameters, which are weighted using the fuzzy tangential function.

The organization of the paper is as follows. Section 2 reviews the literature based on block motion estimation. In section 3, the motivation for the proposal of the adaptive search algorithm is deliberated. Section 4 presents the proposed adaptive search algorithm and the fuzzy tangent-weighted objective function in detail. The experimentation results and analysis based on the performance evaluation metrics are discussed in section 5. Finally, section 6 concludes the paper.

2. Literature review

Reviews of the recent works that are related to block motion estimation are listed in table 1. Accordingly, 13 research papers are reviewed in this section. The papers [5, 10–21]

are based on the block motion estimation scheme. The intention of motion estimation is to attain the motion blocks, without compromising on the visual quality and compression. The computational overhead of motion estimation is yet another norm, which needs utmost consideration, in the case of a good motion estimation algorithm.

In [5, 17], pixel-based motion estimation algorithm has been employed. The motion vectors in the proposed systems have been adversely affected due to the unclear search classification and the refining of the search. The prediction quality of the pixel-based motion estimation algorithm has also been poor. Motion estimation [10, 11] based on multi-hexagon grid search has perilous criteria regarding the computation time, even though the accuracy of the predictability is good. Normally, the size of the macro blocks has large influence on the results obtained from the motion estimation scheme. The macro blocks can have sizes of 4×4 , 8×8 , 16×16 , etc. The larger block motion was tedious to be estimated using the elastic motion model [12, 13]. After the size of the block, the search pattern selection is the criterion that is considered.

The balance in the visual quality, as well as the searching time, is a difficult factor to be attained while adapting the new search patterns. The octagon- and the triangle-based searches have found difficulty in handling the discontinuities in the frame of the video sequences [14, 16]. The number of search points has been increased with the hexagon search in [18]. In the case of the 2D motion estimation algorithm [19], the computation time for searching has increased because of its improper motion parameter selection. Use of mean squared error in the direction selection of the fast search algorithm [20] has made the motion estimation more problematic. The modified full search algorithm with improvement in speed has resulted in increased computation time when compared with the normal full search algorithm. Table 1 presents the works in literature in detail.

Chen and Tsaur [22] have proposed a function for weighting possibilistic mean and variance, and they have derived a weighted fuzzy portfolio model for different investment conditions. These models attain the finest proportion of an investment all over the business cycle to meet the expectations of investors. However, this method investigates only a few values in the weighted function and specifies stages with various weighted possibilistic means and variances.

Igoulalene *et al* [23] have developed two fuzzy hybrid techniques to solve the strategic selection problem for providing chain coordination. The first technique integrates the fuzzy TOPSIS technique and fuzzy consensus-based possibility measure. The second technique integrates goal programming model and the fuzzy consensus-based neat OWA. Here, the Levenshtein distance was used to perform the sensitivity analysis, which was utilized for the first time in multi-criteria group decision making. The drawback of this method is that it did not offer more flexibility to the stakeholders.

Table 1. Literature review.

Author	Contributions	Issues
Chen <i>et al</i> [5]	Adaptive pixel-based motion estimation	Refining the search classification affects the motion vectors severely
Liu <i>et al</i> [10]	Motion estimation based on unsymmetrical-cross multi-hexagon-grid search	Requires more computational effort to search over the partitions
Pan and Kwong [11]	Motion estimation based on direction-based unsymmetrical-cross multi-hexagon-grid search	Recursive search process takes too long time to find the search patterns
Muhit <i>et al</i> [12]	Elastic motion model with larger blocks	Smaller level motion preservation is tough with larger blocks
Muhit <i>et al</i> [13]	Geometry-adaptive partitioning-based elastic motion model	Limited number of searches over the partitions
Duanmu and Zhang [14]	Motion estimation based on octagon and triangle search	Finds difficulty in handling tricky motion, such as zoom and rotation
Bosch <i>et al</i> [15]	Motion model based on visual motion perception	Chance of missing the smaller level motion due to the visual motion perception model
Fabrizio <i>et al</i> [16]	Motion compensation based on tangent distance prediction	Fails to preserve the motion discontinuities
Wei and Xin [20]	Fractional pixel motion estimation algorithm	Lower prediction quality
Thambidurai <i>et al</i> [17]	Hexagon search algorithm	Increased number of search points
Kalivas and Sawchuk [18]	2D motion estimation algorithm	Increased computation time with improper initialization of motion parameters
Alvar <i>et al</i> [19]	Fast search algorithm	Inaccurate, since it uses MSE
Kim <i>et al</i> [21]	Fast full search motion estimation algorithm	Increased computation time
Chen and Tsaur [22]	A weighted fuzzy portfolio model	This method investigates only a few values in the weighted function
Igoulalene <i>et al</i> [23]	Fuzzy hybrid technique	The drawback of this method is that it did not offer more flexibility to the stakeholders
Aymerich <i>et al</i> [24]	Algorithm based on a Mamdani-type fuzzy rule-based system	This method had rate errors during segmentation
Wang <i>et al</i> [25]	Integrated the cuckoo search with harmony search algorithm	This method did not address the real world problems
Mirjalili and Gandomi [26]	Chaotic Gravitational Constants for the Gravitational search algorithm	This method did not solve the real world engineering problems
Verdoliva <i>et al</i> [27]	Fast approximate nearest neighbour search algorithm	This method needs a great deal for further improvements

Aymerich *et al* [24] have introduced an algorithm to fragment veins in the periventricular region of the brain in susceptibility-weighted magnetic resonance images. The performance of this algorithm in susceptibility-weighted images shows very good segmentation, with a minimum number of false positives. The results of this algorithm were not exaggerated by minute changes in the location and size of the regions of interest. This method had some rate errors during the segmentation process.

Wang *et al* [25] have integrated the cuckoo search with harmony search algorithm to enhance the cuckoo search algorithm. They evaluate the performance of this algorithm on different benchmarks. In the cuckoo updating process, the improvement was applied among cuckoos by the harmony search algorithm. This method did not address the real world problems.

Mirjalili and Gandomi [26] have proposed Chaotic Gravitational Constants for the Gravitational search algorithm. They have also proposed an adaptive normalization

technique to move from the exploration phase to the exploitation phase. This method did not solve the real world engineering problems.

Verdoliva *et al* [27] have suggested an algorithm for fast approximate nearest neighbour search depending on the properties of ordered vectors. This algorithm was correlated with the locality-sensitive hashing in the space of directions in which hashing was based on the order of components. This method provides the state-of-the-art performance on unstructured data, but there is surely a great deal for further improvements.

3. Motivation

Most of the research work on video compression or video coding is completely based on the H.264 or MPEG video standard. The reason is that it provides a significant

application for video compression, owing to its adaptable characteristics. In motion estimation, the task is to find the motion vector from the frame block through making a comparison against the reference frame. Several search algorithms exist to achieve motion estimation, and some of them can be seen in section 2. The computational time of the search algorithm depends on the number of search points. The search algorithm checks the macro blocks for matching the motion block that is as same as that of the reference frame under consideration. The block with motion, which is obtained using the search algorithm, is evaluated based on the trade-off criterion to choose the best matching block. Hence, a search algorithm with reduced search point and a better trade-off criterion is necessary for the optimal performance of video compression.

In order to accomplish this objective, an adaptive search algorithm that incorporates the cross search, the square search and the hexagon search is proposed. Normally, the search patterns search the motion in the block in every direction. In every possible direction, the number of search points gets increased because of the uncertainty. Hence, direction selection is important to reduce the number of search points prior to searching. Here, the cross search will be initially used to find the direction of motion in the block. Once the BDM is chosen by the cross search, the square search starts searching. However, the motion discontinuities are difficult to be preserved. To attain the discontinuity like zooming, rotation and fast moving objects in the macro blocks, the hexagon or orthogonal search is introduced. It enhances the search by overcoming the discontinuity resulting from the search using the edge points in the shape.

It is clear that the shape and the size of the search patterns are the major factors concerned with the achievement of higher video quality (error performance), other than the computational complexity of motion estimation. The intrinsic goal here is to find the matching point. If more search points are used to find the best matching point, the complexity would be high. However, increasing the search points will further increase the gain in video compression. Hence, finding the better matching point to estimate the motion field is a reasonable progression in the field of video coding research. In a searching algorithm, the selection of better search points is always evaluated by an objective function. Here, the rate and the distortion parameters that are based on the fuzzy tangent-weighted function are considered for trade-off selection.

4. Proposed AOCSH search and fuzzy tangent-weighted trade-off for H.264 in motion estimation

This section presents the proposed search algorithm, AOCSH, and fuzzy tangent-weighted trade-off for H.624 motion estimation. The overall block diagram that portrays

the procedure for the proposed AOCSH algorithm is shown in figure 1.

Initially, the video that is to be compressed is taken as the input for the motion estimation algorithm. The frames are then extracted from the video sequences. Individual frames are then split into macro blocks. The macro blocks are the input for the motion estimation algorithm. Later, the blocks are subjected to the hybrid search procedure. The hybrid search integrates the cross search, the square search and the hexagon search. The cross search is not based on the order of the rate and distortion trade-off, but it identifies the direction of the search for motion estimation. Once the direction is identified, the search point in the unwanted direction gets reduced. The cross search enables searching in the horizontal or the vertical direction. The search point in the considered direction is then searched for motion using the square or the hexagon search. The availability of the corresponding search pattern is based on the characteristics of the proposed rate-distortion trade-off. The rate-distortion trade-off is evaluated using the fuzzy tangent-weighted function. Based on the trade-off value, the block with the maximum motion is estimated. Then, the search point is selected for motion vector generation. The generated vector is subjected to transformation and quantization, by which the coefficients for the coding process are generated. Then, using the entropy coding process, CAVLC, the encoding is done. The corresponding video sequences are stored as bit streams so that the video transmission with less storage space is made possible. Then, the decompression is done to retrieve the video sequence from the encoded bit streams. This paper is solely related to motion estimation algorithm.

4.1 Search patterns used in AOCSH

In this section, the search patterns that are used in the proposed motion estimation algorithm are discussed. The selection of search patterns is done in such a way to stabilize the effect of one from another. The search patterns used here are (i) cross search, (ii) square search and (iii) hexagon search.

4.1a Cross search: The cross search is the initial search pattern that is used in the proposed algorithm. The cross search is organized with a total of five search points. Even if the size of the pattern alters in accordance with the size of the macro block, the number of search points will not get amended. The small or the large cross search pattern is used to find the direction of motion in the macro blocks.

Directional cross search

The direction of motion that aids in performing motion estimation is initialized by the cross search. In cross search, the prediction of initial search centre is the first step. The

Figure 1. Block diagram.

neighbouring motion blocks are well correlated to their motion vectors so that they allow the prediction of the initial search centre [28]. For reducing the computational complexity, the current block is considered with zero motion vectors. Early termination of the search for accomplishing the optimal direction search is performed through comparing the Sum of Absolute Differences (SAD) values, and it tends to reduce the cost of block matching, without degrading the visual quality. Based on the cross pattern utilized, the size of the cross search varies. If the cross pattern is 2×2 , the cross search involves a small cross search pattern. If the cross pattern is 4×4 , the cross search will be a large cross search pattern. The cross search finds the direction by making use of the minimum SAD value. The size of the cross pattern depends on the depth value (d).

Assume that the initial search centre, which is predicted for the cross search, is (x, y) . The searching of the direction using the cross search involves the

following four search points: the search points in horizontal right, horizontal left, vertical top and vertical bottom direction. The search point in the four directions is represented as

$$SP_i^C = (x \pm p, y \pm q); \quad p = 0, \dots, d; \quad q = 0, \dots, d; \quad i = 1, 2, 3, 4 \quad (1)$$

where SP_1^C is the search point in the horizontal right direction, SP_2^C is the search point in the horizontal left direction, SP_3^C is the search point in the vertical top and SP_4^C is the search point in the vertical bottom direction that are defined in the cross search region of $d \times d$.

Figure 2 shows the cross search with a depth size of 2, 1 and 4 (i.e., 2×2 , 1×1 and 4×4 patterns, respectively).

4.1b Square search: Square search is a centre-based search [29] and it has a half-way stop provision. It fixes a pattern size, no matter what the search parameter value (p) is.

Figure 2. Cross search.

Adaptive Order Square Search

Square search of the proposed AOCSS algorithm is modified with the adaptive order square process. The adaptive order is used to choose the square search as the search pattern and to select the size of the search. The adaptive order is based on the fuzzy tangential trade-off criterion. In motion estimation, the square search uses five search points. The pattern is generated from four corners and one centre of the search point. The ease of the square search is initiated by the cross search itself. Since the direction of motion is waved up by the cross search, the square searching is done only in the concerned direction. Basically, the square search can quickly detect the motion in any direction because of its symmetrical searching capability. In the case of the motion block, the motion is usually in the upward or the downward direction. If vertical-up is the direction mentioned by the cross search, the square search is done along the vertical-up direction. Hence, the square search is beneficial and robust. In Adaptive Order Square Search, the movement of the search point is obtained through five points and the order, which can be either 0 or 1. Order denotes the parameter that is introduced in the proposed algorithm to resemble the size of the square. The depth of the square search can be easily found using the following equation:

$$d = O + C \quad (2)$$

where d indicates the depth or the size of the square, O specifies the order of the search and C represents the constant parameter that is to be fixed manually. Since the nature of the order value is 0 or 1, two square-sized searches are possible in the proposed algorithm.

Let (x, y) be the centre location of the square search point. The searching of the best matching block is possibly

done using the following four search points, along with the centre points. The four search points are obtained by crossing the centre search point with the corner search points in the upper-right, upper-left, down-left and down-right directions, which is given by a general representation as

$$SP_j^S = (x \pm p, y \pm q); \quad p = 0, \dots, d; \quad q = 0, \dots, d; \quad (3) \\ j = 1, 2, 3, 4$$

where SP_1^S is the search point in the upper-right direction, SP_2^S is the search point in the upper-left direction, SP_3^S is the search point in the down-left direction and SP_4^S denotes the search point in the down-right direction, defined in the square region of $d \times d$.

In figure 3, the square search with depth size of 1 and 2 is shown. The variation in the depth is identified from the size of the square search pattern. The size does not alter the searching points in number.

Let the order be 0 and the constant be 1. Then

$$d = 0 + 1 = 1. \quad (4)$$

If the order is 1 with constant value 1, then

$$d = 1 + 1 = 2. \quad (5)$$

4.1c Hexagon search: Hexagon search is an important search pattern, which is also a centre-based one. Translating, zooming and tilting motion in the macro blocks can be accomplished with the hexagonal search and it is a modified form of the diamond search. Hexagon search reduces the two searching points of the diamond search and provides an efficient estimation scheme. When compared with the other search patterns, the hexagonal search has undergone a considerable development of the other motion

Figure 3. Adaptive Order Square Search.

estimation algorithms because of the hexagonal structure with fewer search points.

Adaptive Order Hexagonal Search

Hexagon search is the last search pattern that is used in the proposed AOCSS algorithm. The discontinuities in the motion movement, which cannot be detected by the square search, can be easily estimated using the hexagon search pattern. Here, the discontinuities are easily detected because the search point is defined through the hexagonal corner, i.e., with wide acceptance. It finds the best matching block because of the hexagonal centre pixel. In a hexagon, seven search points are present. Six points are from the corners of the hexagon with one search point as the centre. The best matching block is detected based on the rate distortion condition.

The adaptive hexagon search pattern that is proposed is modified by the incorporation of the order concept and trade-off criterion. The depth of the hexagonal shape is described by the order. The centre points move through the hexagonal corner points to find the motion. The order defined for the hexagonal shape is either 0 or 1 and the depth of the hexagon is the summation of the order of the search and a constant parameter C . The constant value is fixed manually. The depth because of the two orders is 1 and 2; hence, two sets of hexagons are introduced for finding the best matching block.

Assume (x, y) as the centre location of the hexagon centre search point. The centre point is moved to the six corner points to detect the matching block. The centre point traverses on the hexagonal corner points in every possible direction to find the motion, and it is given as

$$\begin{aligned} SP_k^H &= (x \pm gp, y \pm q); \quad p = 0, \dots, d; \quad q = 0, \dots, d; \\ k &= 1, 2, \dots, 6 \end{aligned} \quad (6)$$

where SP_1^H , SP_2^H , SP_3^H , SP_4^H , SP_5^H and SP_6^H specify the six search points that are defined in the hexagonal corners. The first four search points in the directions upper-right, upper-

left, down-left and down-right directions assume a value 1 for the factor g , i.e., $g = 1$. For the search points SP_5^H and SP_6^H , the value is $g = 2$.

The hexagonal structure with different depth sizes can be viewed from figure 4. The size of the hexagon varies because of the change in order only.

4.2 Fuzzy tangent-weighted trade-off criterion

The important step to find the best optimal block using the proposed algorithm is the fuzzy tangent-weighted trade-off criterion. The centre pixels are moved through the cross shape, the square shape and the hexagon shape searches, and the selection criterion after searching different blocks is obtained based on the weighted function. The weight for the block with motion is calculated based on the fuzzy tangential function. The fuzzy tangent-weighted function gets advantage because it is fuzzy. Since fuzziness is a natural way to express uncertain information, it helps in finding the optimal block from the macro blocks with uncertain motion. The fuzzy function is more flexible and linear in parameter system, and hence the best possible block selection can be made available through considering the normal tangent-weighted function [30]. The fuzzy membership function converts the motion parameter of certain range into linguistic variables, and thereby the possibility of selecting the best optimal matching block is improved. The reason for the enhancement is that the restricted range selection is overcome by the linguistic variables. The selection of blocks in a match with the reference must have a good correlation among one another. The blocks with minimum motion are stored using lower bits, and hence the compression performance is improved.

The formula for best matching block selection is based on two parameters, called rate and distortion. The rate is allied with the minimization of the storage bits. Distortion is allied with performance optimization, which is achieved

Figure 4. Adaptive Order Hexagon Search.

by finding the minimum motion block. The rate parameter is designed in such a way that it brings the best performance through the minimization of the storage bits needed during compression, whereas the distortion parameter is designed to bring the best performance through the minimum motion block. Based on the rate and the distortion parameters, the maximization function is formulated as follows:

$$T(b) = \eta T_d(b) + \tau T_r(b) \quad (7)$$

where η and τ denote the weighted constants that correspond to both the parameters T_d and T_r . The first parameter belonging to distortion ($T_d(b)$) considers the motion-dependent parameter ($M(b)$) and weight function ($W_d(b)$).

The parameter, $T_d(b)$ should be maximum for better performance. The motion-dependent parameter can be expressed as

$$T_d(b) = f(M(b))W_d(b). \quad (8)$$

The function $f(M(b))$ indicates the fuzzy membership function. The fuzzy membership function converts the motion-dependent parameter from its range of 0–1 into the linguistic function.

The motion-dependent parameter $f(M(b))$ is computed by taking the difference ratio of the size of the block, with respect to the pixels having the static values. Hence, this parameter will attain the maximum value if there is an exact equivalent block like the reference block. Otherwise, the values will be towards zero. The maximum value refers to the value of 1. The motion-dependent parameter can be stated as

$$f(M(b)) = f\left(\frac{S(b) - A(b)}{S(b)}\right) \quad (9)$$

where $S(b)$ points to the number of pixels in the block and $A(b)$ denotes the count of pixels having dynamic values similar to those of the previous frame.

The weightage for the motion-dependent parameter $M(b)$ is based on the size of the blocks and fuzzy tangential function, which always lie on the linguistic variables. The weightage function is calculated using the formula

$$W_d(b) = \left[\frac{1}{1 + \exp[-S(b)]} \right]. \quad (10)$$

The parameter that is considered for the trade-off criterion, other than distortion, is the rate. The rate parameter $T_r(b)$ considers the rate-dependent parameter $R(b)$ and the weight function. The rate-dependent parameter considers the bits that are required to store the block before and after CAVLC coding. This parameter should have maximum value for better performance. The rate parameter is given by

$$T_r(b) = f(R(b))W_r(b) \quad (11)$$

where $f(R(b))$ is the fuzzy membership function that corresponds to the rate distortion parameter. The membership function converts the rate distortion within the range of 0–1 into the linguistic variables. Rate-dependent parameter $R(b)$ is calculated by finding the difference ratio of the bits that are required to store the block before and after CAVLC coding. If this parameter obtains the maximum value, then the block is better regarding compression performance. Otherwise, the values will be near zero.

$$f(R(b)) = f\left(\frac{R_B(b) - R_A(b)}{R_A(b)}\right) \quad (12)$$

where $R_A(b)$ denotes the number of bits required for storing the block after compression and $R_B(b)$ specifies the number

of bits required for storing the block before compression. The weightage for the rate-dependent parameter $T_r(b)$ is based on the bits that are required for storing the block after compression and fuzzy tangential function, which always lies on the linguistic variables. The weightage in accordance with the rate-dependent parameter can be calculated as

$$W_r(b) = \left[\frac{1}{1 + \exp[-R_A(b)]} \right]. \quad (13)$$

4.3 Design flow of the proposed AOCSH algorithm

The design flow of the proposed AOCSH search algorithm is discussed in this section.

Initially, the cross search pattern finds the direction of motion that is extant in the block. Once the direction of motion is known, the square search pattern resumes the job of motion estimation. Based on the depth of the search, the size of the square is selected. The order value emphasizes the depth of the square search. At first, for the order value of zero and the constant value, the first five search points are visited in the motion block. Alongside of order 1, the search points with different sizes of the square search visit the block in search of motion. Hence, the adaptive order will do two square searches for every searching process. Once the square search is completed, the hexagon search takes place. In Adaptive Order Hexagon Search, two sets of hexagon search points are visited for order 0 and order 1 to find the best matching block. The best matching block resulting from the proposed search algorithm is selected based on the rate-distortion trade-off criterion. It is calculated using the fuzzy tangent-weighted function. Figure 5 shows the overall search that takes place in the search algorithm. In figure 5, the depth of the square search or the hexagon search is not mentioned. Usually, the search takes place in the horizontal right or left, vertical top or bottom direction as estimated by the cross search. Figure 5 shows the search in the horizontal right direction.

The proposed AOCSH algorithm is deliberated in four steps. The algorithmic description of the AOCSH algorithm is given in figure 6.

Step 1. Starting: In the current frame, the centre point (x, y) is located at the same location of the reference frame. Then, a block of size $B \times B$ is formed by fixing the centre point as the centre pixel.

Step 2. Cross search: The cross search is performed to find the direction of motion in the block. The search point with the direction is initiated as the centre for the successive steps.

Step 3. Adaptive Order Square Search: The adaptive square search is performed for the given S value, with respect to the depth and the order, by traversing through the four corner points in the right, left, top and bottom direction. For all the five points visited in the square shape for order 0, the block is formed by fixing the search point of the cross search as the centre pixel. Similarly, for order 1, again five points will be searched to find the best matching block.

Step 4. Adaptive Order Hexagon Search: The search point from the square search is used as the centre pixel for the hexagon search. After the completion of the square search, the hexagon search is performed to find the best matching blocks. The centre point of the hexagon transverses through the six corner points to find the optimal block. For all the pixels visited through the hexagon and square searches, fuzzy tangent-weighted trade-off criterion is calculated. The block having the maximum value is taken as the best matching block.

4.4 Encoding issues with the AOCSH algorithm

After the best matching block is estimated for redundancy reduction, the motion vectors are generated based on the motion-estimated value. After finding the motion vectors, the encoded bits are generated for video compression. It is almost similar to the H.264 estimation. Initially, the video

Figure 5. AOCSH search pattern.

	AOCSH Algorithm
1	Input: Initial Point, (x, y) , Constant, C
2	Output: Best matching point (x_b, y_b)
3	Procedure
4	Start
5	Read initial point, (x, y)
6	Draw block for the initial point
	Find the trade-off criterion, $T(b)$
7	Directional cross search
	Get search points $SP^C_1, SP^C_2, SP^C_3, SP^C_4$ using Cross search
8	Initiate directional search points as the center points
7	For order 0 and 1
8	Find depth d
9	Get search points SP^S_1, SP^S_2, SP^S_3 and SP^S_4 using square search
10	Get search points SP^H_1, SP^H_2, SP^H_3 and SP^H_4, SP^H_5 and SP^H_6 using hexagon search
11	Find the trade-off criterion, $T(b)$ for all the search points
12	EndFor
13	Select the best point (x_b, y_b)
14	Return the best point (x_b, y_b)
26	End

Figure 6. AOCSH algorithm.

input for compression encodes the height of the frame, the width of the frame, number of frames, frame rate and the scalable quantization parameters as 8 or 16 bits. Once the parameters are encoded, the inter-frame is encoded by CAVLC. Before entropy coding, the I frame is subjected to blocking, integer transformation and quantization. Only after the completion of the quantization process, they are encoded into bits. Then, every predicted frame (P frame) is encoded based on the motion vectors and mode. The mode is used to recognize the location of the search points as they are encoded in three bits. The first bit denotes the order of the search, whereas the second bit and the third bit denote the direction of motion. The second bit denotes the direction in upward direction or downward direction. In contrast, the third bit denotes the direction of movement in the right or the left direction. The bit coding from the CAVLC is inserted for transmission, only after inserting the mode. This process is continued for the entire frame in the video sequence.

5. Results and discussion

The experimental results of the proposed video compression scheme are revealed in this section. The performance of the proposed algorithm is evaluated using the evaluation metrics like structural similarity index (SSIM) and peak signal to noise ratio (PSNR). The efficiency of the proposed method is compared against the efficiency that is achieved with the elastic model and the H.264 video compression standards.

5.1 Experimental set-up

The details regarding the testing of the proposed AOCSH algorithm is discussed in this section. The experimentation of the proposed system has been achieved with a system that possesses the following:

- i3 processor,
- 2 GB RAM and
- Windows 10 OS.

The software tool that is used here is MATLAB (R2014a). The validation of the proposed methodology is done using five videos, namely, football, garden, tennis, foreman and salesman [31]. The characteristics of the videos that are considered for experimenting are given in table 2.

For the ease of analysis, the frame rate, the scalar value and the macro block size of the selected videos are assigned similar values. The time of the video sequence that is taken for trial is 2 s.

Table 2. Characteristics of input videos.

	Time (s)	Frame rate	Number of frames	Scalar value for quantization	Macro block size
Football	2	30	60	27	16×16
Garden	2	30	60	27	16×16
Tennis	2	30	60	27	16×16
Foreman	60	30	1800	27	16×16
Salesman	60	30	1800	27	16×16

Figure 7. Sample frames from three different videos.

5.2 Evaluation metrics

The performance of the proposed system is evaluated using the metrics like SSIM [32] and PSNR. A comparison of the proposed scheme's performance against the performances of the existing elastic model and the H.264 standard is made for evaluation purpose.

(a) SSIM

SSIM is used to evaluate the visual quality of the video after decompression. The definition of SSIM metrics can be given as

$$\text{SSIM}(F_D, F_O) = \frac{(2M(F_D)M(F_O) + C_1)(2CV(F_D, F_O) + C_2)}{(M^2(F_D) + M^2(F_O) + C_1)(V(F_D) + V(F_O) + C_2)}$$

where $M(F_D)$ specifies the mean of the decompressed frame F_D , $M(F_O)$ refers to the mean of the original frame F_O , $V(F_D)$ represents the variance of the decompressed frame F_D , $V(F_O)$ indicates the variance of the original frame F_O and $CV(F_D, F_O)$ denotes the co-variance of frames F_D and F_O . Finally, C_1 and C_2 specify the constants.

(b) PSNR

PSNR can be calculated using the following equation:

$$\text{PSNR} = 10 \log_{10} \frac{E_{\max}^2 I_w I_h}{\sum \sum (I_{xy} - I_{xy}^*)^2}$$

where I_w and I_h are, respectively, the width and height of the image, I_{xy} is the original pixel value at coordinate (x, y) , I_{xy}^* is the decompressed pixel value at coordinate (x, y) and $E_{\max}$ is the largest energy of the pixels (i.e., $E_{\max} = 255$).

The average PSNR for the compression rate at which the frames are compressed (bit/s) is computed using this equation. The PSNR preserves the quality of the visual image.

5.3 Experimental results

The results of testing that are associated with the proposed AOCOSH video compression algorithm is discoursed in this section. The sample frame from five different videos (football, garden, tennis, foreman and salesman) is shown in figure 7. The compressed frame of the sample frame under consideration is shown in figure 8. Though several changes exist, there is no change in the sample or the compressed video frame, when visualized. The reason is that the visual degradation in the compressed frame is unable to be identified by the human visual system.

5.4 Analysis based on SSIM

The analysis of the experimented football, garden and tennis videos based on the SSIM is discussed in this section. The performance curve is plotted between SSIM and the bit rate.

(a) The performance evaluation curve that is based on SSIM for the football video is shown in figure 9. The SSIM values for the proposed as well as the existing algorithms, when the bit rates are 1.4 and 2.8 Mbit/s, are shown for the football video. At 1.4 Mbit/s, the value of SSIM for the H.264 standard and the elastic model is 0.6495. However, the value of SSIM for the AOSH algorithm is 0.9298. The proposed AOCOSH algorithm has the increased SSIM value of 0.9763, which is 0.3268 higher than that of the existing standard. The value of SSIM for the increased rate of 2.8 Mbit/s is the same as that of 1.4 Mbit/s. It is evident from the curve that the proposed algorithm provides increased SSIM, and hence better compression performance.

(b) Figure 10 shows the performance evaluation curve based on the SSIM for the garden video. For the bit rate of

Figure 8. Decompressed frame from three different videos.

Figure 9. Analysis based on SSIM for football video.

1.4 Mbit/s, the SSIM value of the proposed algorithm is 0.99. The value of SSIM for the AOC SH is less than that of the proposed algorithm, which is 0.9375. The H.264 standard in the elastic model has the value of SSIM as 0.74, which is lesser than the proposed scheme's SSIM. Thus, the value of SSIM for the proposed video is better, when compared with the other compression standards. The increase in the bit rate of 2.8 Mbit/s also shows that the value of the SSIM is high for the proposed method, comparing the Adaptive Order Square Hexagon search (AOSH), H.264 and the elastic model.

(c) The performance curve that relates the SSIM and the bit rate for the evaluation of the tennis video is shown in

Figure 10. Analysis based on SSIM for garden video.

figure 11. The value of SSIM that is achieved for the proposed, AOSH, elastic model and H.264 standards, when the bit rate is 1.4 Mbit/s, is 0.9847, 0.9378, 0.8564 and 0.8564, respectively. The comparison of the SSIM values states that the SSIM value of the proposed system is higher. Similarly, the values for the bit rate of 2.8 Mbit/s are, respectively, 0.984, 0.9378, 0.8564 and 0.8564, with respect to AOC SH, AOSH, elastic model and H.264 standards. The value of SSIM is 0.1283 higher than the SSIM of the other existing compression algorithms.

(d) The performance analysis based on SSIM for the foreman video and salesman video is given in table 3. For the foreman video, when the bit rate is 1.4 Mbit/s, SSIM value for the existing methods, AOSH, elastic model and H.264 is 0.9435, 0.8705 and 0.8705, whereas, in AOC SH, it

Figure 11. Analysis based on SSIM for tennis video.

is 0.99. The SSIM value remains the same, i.e., 0.9105, for the existing methods, elastic model, and H.264, for the bit rate 2.8 Mbit/s, similar to the previous case. Meanwhile, the proposed AOC SH algorithm has SSIM of 0.99. This suggests that AOC SH has better performance than other existing methods for the foreman video. For salesman video, the SSIM value obtained using AOC SH, AOSH, elastic model and H.264, for the bit rate 1.4 Mbit/s, is 0.99, 0.9612, 0.9552 and 0.9552. When the bit rate is 1.4 Mbit/s, the SSIM value is 0.99 in the proposed as well as the existing methods. However, the proposed AOC SH is found to have higher performance than the existing techniques compared.

On comparing the values of SSIM, the proposed AOC SH search algorithm proves to be the best optimal motion estimation algorithm for the compression standards. For all the five videos that are taken for experimentation, the value of SSIM concerned with the proposed system is found to increase than the existing algorithms.

5.5 Analysis based on PSNR

For the football, garden and tennis videos that are taken for experimentation, the analysis based on the PSNR is

described in this section. The analysis based on PSNR is plotted between the PSNR and the bit rate.

(a) For the football video as the input, the value of the PSNR at the bit rates of 1.4 and 2.8 Mbit/s are evaluated. The performance curve that evaluates the supremacy of the proposed algorithm over the existing algorithms is shown in figure 12. Normally, the PSNR value must be improved to yield an efficient compression performance. The PSNR values are shown in percentage in the figure. The value of PSNR for the bit rate of 1.4 and 2.8 Mbit/s are orderly 38.7335, 36.8891, 23.1940 and 23.1940 dB for the proposed AOC SH, AOSH, elastic model and H.264. The PSNR value of the proposed algorithm is found as 38.7335 dB. This proves the fact that the proposed motion estimation algorithm is more suitable for the compression standards to yield the optimized performance. The increase in the bit rate also causes the PSNR value of the proposed scheme to be larger than that of the existing one.

(b) Figure 13 shows the performance evaluation curve of the garden video when the analysis is based on PSNR. For 1.4 Mbit/s, the value of PSNR for both the elastic and the H.264 model is 20.1573 dB. The PSNR of 36.8679 dB is achieved using the AOSH algorithm, while it is 38.7113 dB for the proposed algorithm. The PSNR value for the garden

Figure 12. Analysis based on PSNR for football video.

Table 3. Analysis based on SSIM for foreman video and salesman video.

	Foreman video		Salesman video	
	Rate = 1.4 Mbit/s	Rate = 2.8 Mbit/s	Rate = 1.4 Mbit/s	Rate = 2.8 Mbit/s
AOC SH	0.9900	0.9900	0.9900	0.9900
AOSH	0.9435	0.9835	0.9612	0.9900
Elastic	0.8705	0.9105	0.9552	0.9900
H264	0.8705	0.9105	0.9552	0.9900

Figure 13. Analysis based on PSNR for garden video.

Figure 14. Analysis based on PSNR for tennis video.

video using the proposed method is comparatively higher than that of the existing standards. Similarly, for the bit rate of 2.8 Mbit/s, the value of the PSNR attained by the proposed AOC SH algorithm is considerably higher than the PSNRs of the existing algorithms.

(c) The performance evaluation curve to analyse the proposed system based on PSNR, in the case of the tennis video, is shown in figure 14. The performance curve is drawn between the PSNR values and the corresponding bit rates. The experimental results for two bit rates, namely, 1.4 and 2.8 Mbit/s, are viewed. From the curve, the value of PSNR that is interpreted for the proposed algorithm, AOSH algorithm, elastic model and H.264 is 38.3539, 36.527, 27.9970 and 27.9970 dB, respectively. Here too, the PSNR value of the proposed algorithm is 11% higher than the PSNR of the existing algorithm.

(d) The PSNR analysis result for the foreman video is presented in table 4, by varying the bit rates as 1.4 and 2.8 Mbit/s in the four techniques to be compared. The PSNR value of AOSH, elastic model and H.264 is 38.5971, 30.0763 and 30.0763 dB, respectively, when the bit rate is 1.4 Mbit/s, whereas, in AOC SH, it is 40.527. As the bit rate is increased to 2.8 Mbit/s, the PSNR tends to increase in all the comparative techniques. At this instant, the PSNR value in AOC SH, AOSH, elastic model and H.264 is, respectively, 40.927, 38.9971, 30.4763 and 30.4763 dB.

(e) In table 5, the analysis based on PSNR for the salesman video is given. When the existing techniques, AOSH, elastic model and H.264, attained a PSNR of 38.1216, 37.2209 and 37.2209 dB, respectively, for the bit rate 1.4 Mbit/s, the proposed AOC SH has 40.0277 dB as the PSNR value. For the bit rate 2.8 Mbit/s, the approaches, AOC SH, AOSH, elastic model and H.264, have PSNR, respectively, of 40.5277, 38.6216, 37.7209 and 37.7209 dB.

Table 4. Analysis based on PSNR for foreman video.

	Rate = 1.4 Mbit/s	Rate = 2.8 Mbit/s
AOC SH	40.5270	40.9270
AOSH	38.5971	38.9971
Elastic	30.0763	30.4763
H264	30.0763	30.4763

Table 5. Analysis based on PSNR for salesman video.

	Rate = 1.4 Mbit/s	Rate = 2.8 Mbit/s
AOC SH	40.0277	40.5277
AOSH	38.1216	38.6216
Elastic	37.2209	37.7209
H264	37.2209	37.7209

The evaluation curve proves the compression standard based on the proposed AOC SH algorithm and fuzzy tangent-weighted objective function to be more efficient than the existing algorithms. The proposed algorithm is found to render a better visual quality with reduced bit storage for compression purpose.

5.6 Analysis based on computation time

In this section, the performance analysis based on computation time (in seconds) in all the four techniques for the five videos with varied bit rates is provided using table 6. For the football video, when the bit rate is 1.4 Mbit/s, AOSH, elastic model and H.264 take computation time, respectively, of 5.13, 7.83 and 9.23 s, whereas, in AOC SH, it is 4.33 s. In the same video, for the bit rate 2.8 Mbit/s, the computation time required in AOC SH is 4.01 s, whereas the minimum time

Table 6. Analysis based on computation time (in seconds).

Videos	Bit rates	AOC SH	AOSH	Elastic model	H.264
Football video	Rate = 1.4 Mbit/s	4.33	5.13	7.83	9.23
	Rate = 2.8 Mbit/s	4.01	5.83	7.87	8.96
Garden video	Rate = 1.4 Mbit/s	5.65	6.82	12.01	15.17
	Rate = 2.8 Mbit/s	5.56	6.74	12.61	15.56
Tennis video	Rate = 1.4 Mbit/s	4.87	4.49	6.04	7.18
	Rate = 2.8 Mbit/s	4.75	4.41	6.08	7.28
Foreman video	Rate = 1.4 Mbit/s	3.52	4.61	6.72	5.96
	Rate = 2.8 Mbit/s	3.28	3.74	4.77	6.62
Salesman video	Rate = 1.4 Mbit/s	4.24	3.35	4.22	5.51
	Rate = 2.8 Mbit/s	3.40	3.40	4.39	5.84

taken among the four techniques is 5.83 s, by the AOSH approach. For the garden video, the minimum time taken for the computation by AOC SH is 5.56 s, when the bit rate is 2.8 Mbit/s. Meanwhile, AOSH, elastic model and H.264 take the computation time, respectively, of 6.74, 12.61 and 15.56 s. The minimum computation time observed by AOC SH, for the foreman video, is 3.28 s, when the bit rate is 2.8 Mbit/s, while AOSH, elastic model and H.264 take, respectively, 3.74, 4.77 and 6.62 s. Thus, from the computational time analysis, it is seen that the proposed AOC SH approach has better performance, as it takes the least computation time in most of the cases.

5.7 Discussion

Based on the overall comparative analysis made regarding SSIM, PSNR and computation time, the following discussion is carried out. The performance of the proposed AOC SH approach is compared to the existing AOSH, H.264 and elastic models for five different videos, football, garden, tennis, foreman and salesman. From the analysis, it is seen that the proposed approach attains the maximum SSIM of 0.99 and maximum PSNR of 40.92 dB with reduced computation time of 3.28 s. This shows that the proposed method with the AOC SH-based search algorithm can reduce the computation time, as it makes motion estimation effective with the utilization of depth, direction leaning square and hexagon search processes. Moreover, the fuzzy tangent-weighted trade-off function improves the visual quality and thereby enhances PSNR and SSIM. However, as seen in table 6, for certain cases, the proposed approach takes high computation time due to the fuzzy tangent-weighted function. Hence, in the future work, necessary techniques to reduce computation time are to be considered.

6. Conclusion

This paper has presented a search algorithm, AOC SH, and fuzzy tangent-weighted trade-off function for motion estimation in the video compression standard. The proposed

algorithm is an improved form of H.264 for achieving an improved compression performance. The search patterns that have been incorporated in the proposed AOC SH algorithm include cross search, square search and hexagon search. For an adaptive order of depth, the direction leaning square and hexagon search has been done. This has improved the searching capability with improved searching time and less computation complexity and number of search points. In addition, the fuzzy weighted-tangent function has been used to assess the search points with two parameters, namely, rate and distortion. These two improvements have been applied to the block estimation process of H.264 to improve the visual quality and the compressive performance. The efficiency of the proposed algorithm has been tested using three videos, namely, football, tennis, garden, foreman and salesman, over AOSH, H.264 and elastic model. The evaluation has been carried out using the performance metrics like SSI and PSNR. The proposed method shows improvements in the compression performance, along with the solid visual quality, in comparison against the existing methods.

References

- [1] Barman S, Chattopadhyay S, Samanta D, Bag S and Show G 2014 An efficient fingerprint matching approach based on minutiae to minutiae distance using indexing with effectively lower time complexity. In: *Proceedings of the International Conference on Information Technology (ICIT)*, pp. 179–183
- [2] Singh S, Bag S and Jenamani M 2015 Relative similarity based approach for improving aggregate recommendation diversity. In: *Proceedings of the Annual IEEE India Conference (INDICON)*
- [3] Wiegand T, Sullivan G J, Bjontegard G and Luthra A 2003 Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* 13(7): 560–576
- [4] ITU-T and ISO/IEC JTC 1 2010 Advanced video coding for generic audiovisual services. In: *ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*
- [5] Chen X, Canagarajah N, Nunez-Yanez J L and Vitulli R 2012 Lossless video compression based on backward

- adaptive pixel-based fast motion estimation. *Signal Process. Image Commun.* 27: 961–972
- [6] ITU 1993 *Video codec for audiovisual services at $p \times 64$ kbit/s*. ITU-T Rec. H.261
- [7] ITU 1995 *Video coding for low bitrate communication*, version 1. ITU-T Rec. H.263
- [8] Dufaux F and Moscheni F 1995 Motion estimation techniques for digital TV: a review and a new contribution. *Proc. IEEE* 83(6): 858–876
- [9] Cheung C H and Po L M 2005 Novel cross–diamond–hexagonal search algorithms for fast block motion estimation. *IEEE Trans. Multimed.* 7(1): 16–22
- [10] Liu P, Gao Y and Jia K 2014 An adaptive motion estimation scheme for video coding. *Sci. World J.*, Article ID 381056, 14 pp
- [11] Pan Z and Kwong S 2013 A direction-based unsymmetrical-cross multi-hexagon-grid search algorithm for H.264/AVC motion estimation. *J. Signal Process. Syst.* 73: 59–72
- [12] Muhit A A, Pickering M R, Frater M R and Arnold J F 2010 Video coding using elastic motion model and larger blocks. *IEEE Trans. Circuits Syst. Video Technol.* 20(5): 661–672
- [13] Muhit A A, Pickering M R, Frater M R and Arnold J F 2012 Video coding using fast geometry-adaptive partitioning and an elastic motion model. *J. Vis. Commun. Image R* 23: 31–41
- [14] Duanmu C and Zhang Y 2012 A new fast block motion algorithm based on octagon and triangle search patterns for H.264/AVC. *Int. J. Digital Content Technol. Appl.* 6(10): 369–377
- [15] Bosch M, Zhu F and Delp E J 2011 Segmentation-based video compression using texture and motion models. *IEEE J. Sel. Topics Signal Process.* 5(7): 1366–1377
- [16] Fabrizio J, Dubuisson S and Béréziat D 2012 Motion compensation based on tangent distance prediction for video compression. *Signal Process.: Image Commun.* 27(2): 153–171
- [17] Thambidurai P, Ezhilarasan M and Ramachandran D 2007 Efficient motion estimation algorithm for advanced video coding. In: *Proceedings of the IEEE International Conference on Computational Intelligence and Multimedia Applications*, vol. 3, pp. 47–52
- [18] Kalivas D S and Sawchuk A A 1990 A 2-D motion estimation algorithm. In: *Proceedings of the IEEE International Conference on Pattern Recognition*, vol. 1, pp. 271–273
- [19] Alvar S R, Abdollahzadeh M and Seyedarabi H 2014 A novel fast search motion estimation algorithm in video coding. In: *Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 934–937
- [20] Wei Z and Xin Z 2013 A fast hierarchical 1/4-pel fractional pixel motion estimation algorithm of H.264/AVC video coding. In: *Proceedings of the IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 891–895
- [21] Kim J N, Byun S C, Kim Y H and Ahn B H 2002 Fast full search motion estimation algorithm using early detection of impossible candidate vector. *IEEE Trans. Signal Process.* 50(9): 2355–2365
- [22] Chen I F and Tsaur R C 2016 Fuzzy portfolio selection using a weighted function of possibilistic mean and variance in business cycles. *Int. J. Fuzzy Syst.* 18(2): 151–159
- [23] Igoulalene I, Benyoucef L and Tiwari M K 2015 Novel fuzzy hybrid multi-criteria group decision making approach for the strategic supplier selection problem. *Expert Syst. Appl.* 42(7): 3342–3356
- [24] Aymerich F X, Sobrevilla P, Montseny E and Rovira A 2016 Application of a Mamdani-type fuzzy rule-based system to segment periventricular cerebral veins in susceptibility-weighted images. In: *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 612–623
- [25] Wang G G, Gandomi A H, Zhao X and Chu H C E 2016 Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. *Soft Comput.* 20(1): 273–285
- [26] Mirjalili S and Gandomi A H 2017 Chaotic gravitational constants for the gravitational search algorithm. *Appl. Soft Comput.* 53: 407–419
- [27] Verdoliva L, Cozzolino D and Poggi G 2016 A reliable order-statistics-based approximate nearest neighbor search algorithm. *IEEE Trans. Image Process.* 26(1): 237–250
- [28] Banh X Q and Tan Y P 2005 Efficient video motion estimation using dual-cross search algorithms. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 5485–5488
- [29] Barjatya A 2004 Block matching algorithms for motion estimation. *IEEE Trans. Evol. Comput.* 1–6
- [30] Pramanik S and Mondal K 2015 Weighted fuzzy similarity measure based on tangent function and its application to medical diagnosis. *Int. J. Innov. Res. Sci. Eng. Technol.* 4(2): 158–164
- [31] CIPR sequences from “<http://www.cipr.rpi.edu/resource/sequences/sif.html>”
- [32] Wang Z, Bovik A C, Sheikh H R and Simoncelli E P 2004 Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13(4): 600–612