DOI: 10.21917/ijivp.2015.0145

EFFICIENT BLOCK MATCHING ALGORITHMS FOR MOTION ESTIMATION IN H.264/AVC

P. Muralidhar[1] and C.B. Rama Rao[2]

Department of Electronics and Communication Engineering, National Institute of Technology, Warangal, India

E-mail: [1]pmurali@nitw.ac.in, [2]cbrr@nitw.ac.in

Abstract

In Scalable Video Coding (SVC), motion estimation and inter-layer prediction play an important role in elimination of temporal and spatial redundancies between consecutive layers. This paper evaluates the performance of widely accepted block matching algorithms used in various video compression standards, with emphasis on the performance of the algorithms for a didactic scalable video codec. Many different implementations of Fast Motion Estimation Algorithms have been proposed to reduce motion estimation complexity. The block matching algorithms have been analyzed with emphasis on Peak Signal to Noise Ratio (PSNR) and computations using MATLAB. In addition to the above comparisons, a survey has been done on Spiral Search Motion Estimation Algorithms for Video Coding. A New Modified Spiral Search (NMSS) motion estimation algorithm has been proposed with lower computational complexity. The proposed algorithm achieves 72% reduction in computation with a minimal (<1dB) reduction in PSNR. A brief introduction to the entire flow of video compression H.264/SVC is also presented in this paper.

Keywords:

Scalable Video Coding (SVC), Sum of Absolute Difference (SAD), Blocking Matching, Motion Estimation, Inter-Layer Prediction, Peak Signal to Noise Ratio (PSNR), Spiral Search, H.264/SVC

1. INTRODUCTION

Users need a wide variety of video resolutions for specific purposes: low spatial resolution for small screen of Personal Digital Assistant (PDA) and cellular phone, standard television resolution for a terrestrial television and high resolutions for High Definition Television (HDTV) and digital cinema. It is a real challenge for telecommunication systems to process, transmit and store such a variety of video signals. International Telecommunications Union (ITU) addresses real-time point to point or multi-point communications over a network. H.264/AVC (SVC) is the state-of-the-art scalable video codec jointly developed by the Joint Video Team (JVT) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) [1], [2]. The functionality of scalability is very attractive because of reconstructing lower quality or lower resolution videos from partial bit-streams that adapt to various needs of end users as well as various network conditions. Through SVC, different users with different preferences and capabilities can get different quality/resolution videos from a single bit-stream, which greatly promotes resource sharing resulting in better resource utilization.

The scalability feature always comes along with a significant loss in coding efficiency as well as a large increase in decoder complexity compared to the corresponding non-scalable profiles. Several new coding techniques were developed in the scalable extension and the gap of coding efficiency has been reduced with state-of-the-art non-scalable codec while the complexity increase is reasonably maintained. The H.264/SVC (Scalable Video Coding) [1], [2] standard was published as the state-of-art in scalable video area. The generic block diagram of scalable video encoder (H.264/SVC) is shown in Fig.1. The main goal of SVC standard is to partition a traditional monolithic video stream of a H.264 solution in a combination of distinct complementary layers, providing more adaptability to multimedia software applications and hardware devices, mainly when using heterogeneous networks. In order to provide this flexibility H.264/SVC combines distinct and complex algorithms which look for temporal and spatial redundancies between distinct frames and consecutive layers, like motion-compensation temporal filtering, inter-layer prediction, bi-hierarchical frame decomposition, and others innovations.

This paper a didactic SVC model shown in Fig.5 is implemented, to evaluate the fundamental block matching algorithms and inter-layer prediction algorithms. In case of scalability, the interpolation algorithms are used to down-sample and up-sample the video frames at various stages of the video encoding. Image interpolation algorithms [13-16] widely used in the literature such as Nearest-neighbor replacement, Bi-linear interpolation, Bi-cubic interpolation has been implemented. In this paper, we are mainly concerned about image interpolation with respect to the quality of the image compare with the enhancement layer. The qualitative performance analysis of the reconstructed image is compared with the enhancement layer of the scalable video coding using PSNR values.

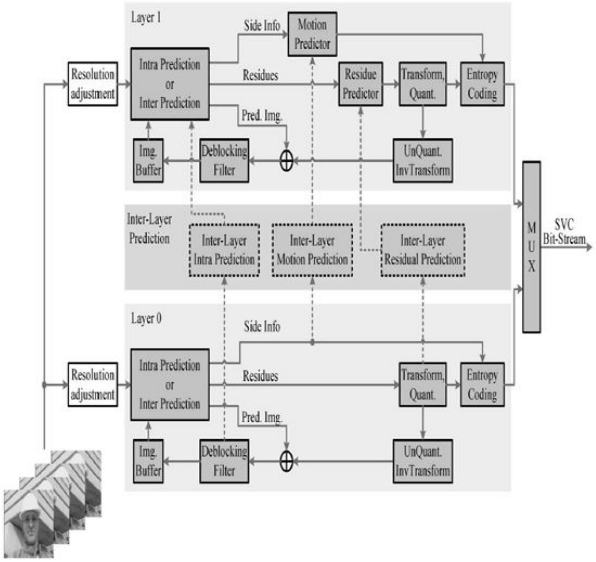

Fig.1.Generic Block Diagram of H.264/SVC video encoder

The paper is further divided as follows: Section 2 gives an overview of motion estimation algorithms. Section 3 explains spiral block matching and its other variants for motion estimation in brief. Section 4 presents the simulation results and their analysis. Section 5 discusses the conclusion.

2. MOTION ESTIMATION ALGORITHMS

The whole idea behind motion estimation based video compression is to save on bits by sending encoded difference images that inherently have less energy and can be highly compressed as compare with the sending a full frame that is encoded. The most computationally expensive and resource hungry operation in the entire compression process is motion estimation. Hence, this field has seen the highest activity and research interest in the past two decades. The well-known full search (FS) [3] is the simplest, but the most computation intensive algorithm, which exhaustively tests all the search points in the search area. Many fast motion estimation (ME) algorithms have been proposed in the literature. These include algorithms with a fixed set of search patterns such as three step search (TSS) [4], new three step search (NTSS) [7]. In TSS [4] it starts with the search location at the centre and sets the 'step size' S = 4, for a search parameter P = 7. It then searches at eight locations +/- S pixels around location (0, 0) from these nine locations searched so far it picks the one giving least cost and makes it the new search origin. It then sets the new step size S = S/2, and repeats similar search for two more iterations until S = 1. At that point it finds the location with the least cost function and the macro block at that location is the best match. It gives a flat reduction in computation by a factor of 9. It has the disadvantage of getting stuck in local minima. Four step search (4SS) [5][6] 4SS looks for 9 locations inside a 5 × 5 window with a step size S = 2 independent of the search parameter value p in the first step. If the least weight is found at the centre the search jumps to fourth step. If the least weight is at one of the eight locations except the centre, then we make it the search origin and move to the second step. The search window is still 5 × 5 pixels wide. In the second step, depending upon the position of the least weight point, we might end up checking 3 or 5 points. Third step is similar to second step. Once again if the least weight is at the centre, jump to fourth step or else move to third step. In the fourth step, window size is reduced to 3 × 3 with step size S = 1. The location with least weight is the best matching macro block and the motion vector is set to point to that location.

Diamond search (DS) [8] DS algorithm uses two diamond shaped search patterns, a large diamond search pattern (LDSP) with nine search points and a small diamond search pattern (SDSP) with five search points shown in Fig.2(a) and Fig.2(b) respectively. The first step of the DS algorithm is to perform LDSP on the coarse search. The best matching point with the minimum SAD is then regarded as the new search centre. If this new centre locates at the diamond's vertex, five additional points are used to perform the next coarse search shown in Fig.2(c) and Fig.2(d). In the similar case, three additional points are used to the centre occurring at the diamond's face shown in Fig.2(d). When the new search centre is located at the centre point, the SDSP is used for the fine search and four additional points are checked in Fig.2(b). The DS process is done when the optimal MV is found by SDSP.

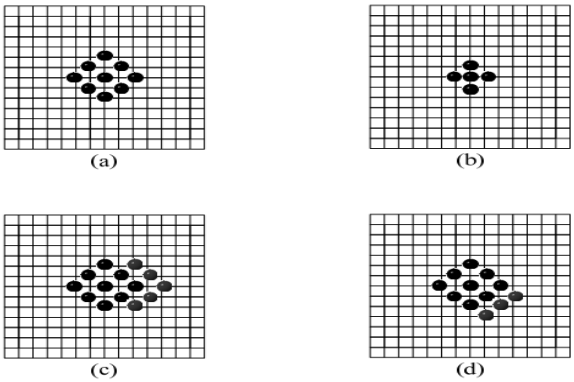

Fig.2. Diamond algorithm search patterns

Hexagon-based search [9][10], cross search [11] and spiral search [12] follow similar type of search procedure

These methods require much less time than the full search technique. However, prediction accuracy may degrade resulting in low coding efficiency along with the degraded image quality.

The motion compensated images are interpolated to the enhancement layer image from the base layer image through various interpolation algorithms like Nearest Neighbour (NN), Bi-linear Interpolation (BL) and Bi-cubic (BC) Interpolation algorithms [13-16] are evaluated based on PSNR. Nearest Neighbour interpolation is simplest form of interpolation in terms of computations. This technique is also known as point shift algorithm and pixel replication. It just replaces the interpolated point with the nearest neighbouring pixel as shown in Fig.3(a).

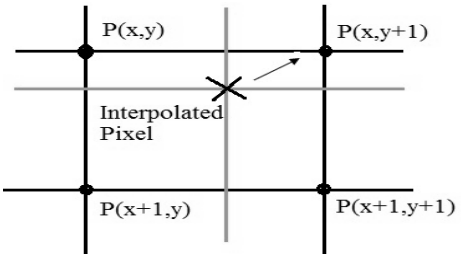

Fig.3(a). Nearest neighbor interpolation method

Bi-linear Interpolation (BL) Bi-linear interpolation is often used to interpolate two dimensional gridded data (Image) between similar data grids i.e. from the corners of a rectangular grid to the centers of the grid boxes which is illustrated in Fig.3(b).

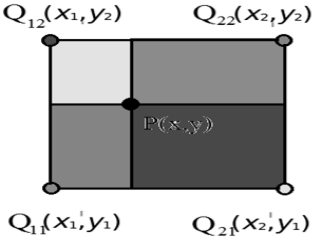

Fig.3(b). Bi-linear interpolation method

Bilinear interpolation can be used where perfect image transformation with pixel matching is impossible, so that one can calculate and assign appropriate intensity values to pixels. B-linear interpolation uses 4 nearest pixel values which are located in diagonal directions from a given pixel in order to find the appropriate color intensity values of that pixel. Suppose that we want to find the value of the unknown function f at the point $P = (X, Y)$. It is assumed that we know the value of f at the four points $Q_{11} = (X_1, Y_1)$, $Q_{12} = (X_1, Y_2)$, $Q_{21} = (X_2, Y_1)$, and $Q_{22} = (X_2, Y_2)$.

We first do linear interpolation in the *x*-direction

$$f(R_1) \approx \left(\frac{X_2 - X}{X_2 - X_1}\right) f(Q_{11}) + \left(\frac{X - X_1}{X_2 - X_1}\right) f(Q_{21}) \qquad (1)$$

where, $R_1 = (X, Y_1)$

$$f(R_2) \approx \left(\frac{X_2 - X}{X_2 - X_1}\right) f(Q_{12}) + \left(\frac{X - X_1}{X_2 - X_1}\right) f(Q_{22}) \qquad (2)$$

where, $R_2 = (X, Y_2)$, we proceed by interpolating in the *y*-direction, which gives us the required $f(X, Y) = f(P)$

$$f(P) \approx \left(\frac{Y_2 - Y}{Y_2 - Y_1}\right) f(R_1) + \left(\frac{Y - Y_1}{Y_2 - Y_1}\right) f(R_2) \qquad (3)$$

Finally, the low-resolution Image data are interpolated to the high resolution Image data by using bi-linear. The coordinates used by bi-linear are grid coordinates, specified by the base resolution image and the scaled grid coordinates of the enhancement image.

Bi-cubic interpolation uses sixteen (4 × 4) neighbouring pixels for estimation. It approximates the local intensity values using a bi-cubic polynomial surface. It determines the grey level value from the weighted average of the 16 closest pixels to the specified input coordinates, and assigns that value to the output coordinates.

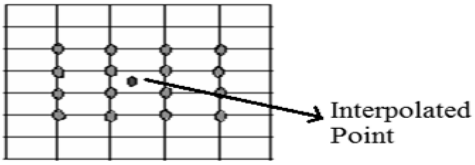

Fig. 3(c). Bi-cubic interpolation method

In case of Bi-cubic Interpolation the number of grid points needed to evaluate the interpolation function is 16, two grid points on either side of the point under consideration for both horizontal and vertical directions. The grid points needed in Bi-cubic convolution interpolation are shown in Fig.3(c).

Further, a modified version of efficient motion estimation algorithm called New Modified Spiral Search (NMSS) is proposed. The proposed algorithms reduces the number of computations by 72% with minimal (<1dB) reduction in PSNR compared with the traditional exhaustive search. The proposed NMSS algorithm presents hardware friendly solution with uniform data access that makes it highly desirable for hardware implementation.

Video encoder estimates the motion of the objects between the reference frame and the current frame. This is called motion estimation (ME). The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame. The block matching motion estimation (BMME) is most widely adopted method for motion estimation in many video compression standards like MPEG-4, H.264/ACV, H.264/SVC.

Usually the macro block is taken as 16 × 16 pixels, and the search parameter *P* is 8 pixels. The matching of one macro block with another is based on the cost function. The macro block that results in the least cost is the one that matches the closest to current block. Among the various cost function available the most widely used cost function is Sum of Absolute Difference (SAD) because it is computationally less expensive compared with the other methods.

$$SAD(w_x, w_y) = \sum_{i=x}^{x+N-1} \sum_{j=x}^{y+N-1} \left| F_t(i, j) - F(i + w_x, i + w_y) \right|$$
$$(u, v) = \min_{-w \le w_x, w_y \le +w} SAD(w_x, w_y) \qquad (4)$$

where, $F_t(x, y)$ represents the pixel value at the coordinate (x, y) in the frame *t* and the (u, v) is the displacement of the candidate MB.

The matched block in the current frame is used to create a vector that stipulates the movement of a macro block from one location to another in the reference frame. The search area for a good macro block match is constrained up to *N* pixels on all fours sides of the corresponding macro block in previous frame. This '*N*' is called as the search parameter or neighborhood of the macro block. In case of large motions, we need to have larger neighbourhood for the search parameter leading to computationally expensive motion estimation. Hence the block matching algorithm is a tradeoff between computational complexity and the quality of the image. To speed up the process of motion estimation without significant affects on the image quality, various fast algorithms [4-10] were developed. Among these the TSS, NTSS, 4SS and DS algorithms have been widely accepted and implemented in many video compression standards.

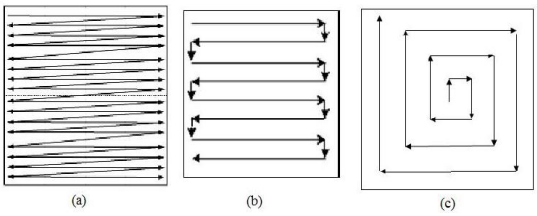

Fig.4(a). Raster Scan Order, (b) Meander Scan Order and (c) Spiral Scan Order

The search pattern/Scan order specifies the movement of macro block within the search region. In most of the previous works the main emphasis has been on reduction of search points and the search patterns are not given any significance. Applying a good search pattern inside a search window can speed up the ME process. In fact, an effective search order should start from a position having the highest probability of requiring the minimum distortion cost, and then move along a path to go through all other positions from high probability to low probability. The distribution of MVs in a typical video sequence is highly biased

towards the central region of the search window. Thus, it is reasonable to place more search points in the central region of the search window to obtain more samples. Most widely used search patterns are the raster scan, snake (meander or s type) scan and the spiral scanning pattern as shown in Fig.4(a), Fig.4(b) and Fig.4(c) respectively. The most generally used scan formats is the raster scan order. In this the search area is scanned from row to row from left to right. The advantage of raster scan is that it is simple and easy to implement but it does not uses the overlapped data making it to use more memory access cycle and it starts scanning from the top left corner of the search area it has to make more search position iteration before reaching the high probability area of the candidate block (i.e. near to search centre), hence more time and update operations are required.

Meander scan is based on the fact that there is high overlapped data among the adjacent blocks. It is basically a modified form of the raster scan to make use of the overlapped data. Meander scan [12] is also known as the snake scan or the s-type scan. In this format, the odd line is scanned from left to right and the even line is scanned from right to left. So, this scan format has three kinds of scan directions and only one row of pixels is changed between the two adjacent blocks. The advantage of this method over raster scan order is that it utilizes the overlapped data thereby reducing the memory access cycle, but it has same problem as of the raster scan, it also starts scanning from the top left corner of the search area. While the meander scan wisely overcame the issue of utilizing the overlapped data but it is unable to start scan from search centre. Thus in order to reuse the overlapped data, and to solve the scan start point problem an outward spiral like scan format of the search area is adopted as shown in Fig.4(c).

3. PROPOSED SPIRAL SEARCH ALGORITHMS

Spiral Search based block matching algorithm compares a current block from the previous fame within search window starting from the centre. Intuitionally, search points closer to search centre usually has more correlation with the current block. Search points near to search centre usually have the highest probability to be the best candidate block. The current block scans the search window spirally as shown in Fig.4(c). Spiral-type motion estimation (STME) [12] is efficient because it is based on the fact that the probability to find the best match is near to the centre of the search window. As the distance from the search centre is increased the probability to find the best match decreases. The search point starts from the search centre and moves to the next neighboring search point with increasing numbers. STME first searches the centre and then slowly increases search area and searches the area in a spiral fashion. The search order can move to the next search point with only one shift to up, down, left, or right. In addition, spiral search order introduces uniform data access suitable for hardware implementation.

3.1 MODIFIED SPIRAL SEARCH (MSS)

In case of Modified spiral search algorithm [12], a uniform decrease in the SAD is observed as it approaches the minimum SAD block. The difference between two blocks separated by 1 pixel was small as we are near the minimum SAD block. Unlike the other fast motion estimation algorithm, a reduction in search points is achieved based on the current SAD. The modified Spiral search is implemented as follows:

Step 1: If the SAD is large (SAD > Th2), we skip the SAD computation of next two blocks in the scan order.

Step 2: If the SAD is moderate (Th1< SAD < Th2), we skip the computation of the next block.

Step 3: If the SAD is small (Th1 < SAD), we continue with the SAD computation for the next block.

The MSS algorithm tries to achieve reduction in the number of search points in the search window which in turn will reduce the number of computation in the motion estimation process. It essentially allows the movement of macro block by one or two or three point during the search scanning for best match.

3.2 NEW MODIFIED SPIRAL SEARCH (NMSS)

In an H.264/SVC encoder, the most time consuming component is variable block-size motion estimation. To reduce the complexity of motion estimation, an early termination algorithm can be used. It predicts the best motion vector by examining only a few search points. With the proposed method, some of the motion searches can be stopped early, and then a large number of search points can be skipped. For the proposed scheme, an assumption is made that the motion vector of a macro block is most probably related to one of its close neighbors, either in the current frame or in previous frames.

The algorithm is designed to find out the optimum threshold to early terminate the search. It is important to note that if the calculated thresholds are too low, then quality restrictions apply meaning that better matches are made, but if the statistics are not accurate, it may take a long time and some macro blocks might even need to use all 961 possible vectors. On the other hand, larger thresholds may lead to earlier exit macro blocks but the quality of the match could be lower. A good solution would require a good initial threshold that will be updated during the macro block's motion search. The proposed algorithm implementation is as follows:

$$T = \mu.T_i\left(1+\frac{p}{k}\right) \tag{5}$$

where, μ is a multiplicative factor, T_i is the threshold at the beginning of the macro block's search procedure, p is a percentage value and k is the number of search positions after which, the threshold is increased by p. In that way, if the original threshold value does not lead to a good match after searching k possible motion vectors, the threshold is increased and so is the possibility for early-exit. Initial threshold is selected based on the fact that the motion vector is related to its close neighbors.

Step 1: If the current SAD is less than adaptive threshold (SAD < T_i) enable early termination else continue with normal full search SAD computation.

Step 2: If the current SAD is high i.e. SAD>Th2, skip the computations of SAD for next two blocks. It indicates the current reference block is away from the best match; hence we skip two blocks to avoid unnecessary computations.

Step 3: If the current SAD is moderate i.e. Th1<SAD<Th2, skip the SAD computation of the next block. The moderate value of the SAD indicates the best match MB is nearer to reference block; hence we skip only one block to reduce the computations.

Step 4: If the current SAD is small i.e. SAD<Th1, continue with computation of SAD for the next block. It indicates the SAD is near the best match hence we do not skip any blocks and continue with the computation.

In the proposed New Modified Spiral Search Algorithm uses both advantages of the above specified methods. In order to exploit the full advantage of the spiral search pattern, as it starts at high probability region around the centre. The Adaptive early termination enables reduction in number of search points in the search window. Further, reduction in computation is achieved by adopting the Modified Spiral Search algorithm which skips the search points based on the SAD. Using higher threshold values the video quality will deteriorate but the complexity will be lower. Experimental results show the Th1 = 300 and Th2 = 450, achieve best tradeoff between computational complexity and video quality [12].

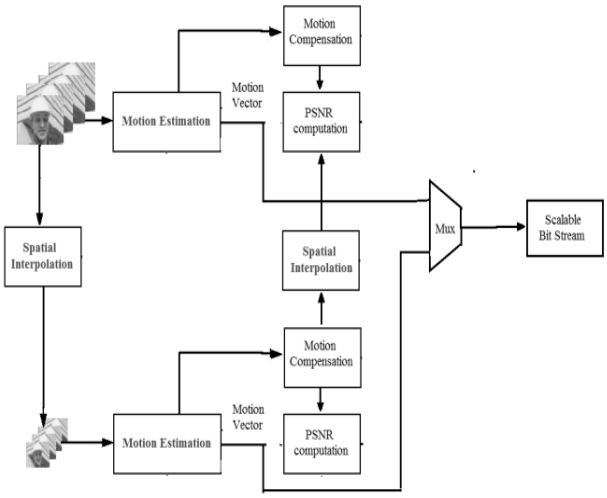

Fig.5. Implementation of Didactic Scalable Video Codec

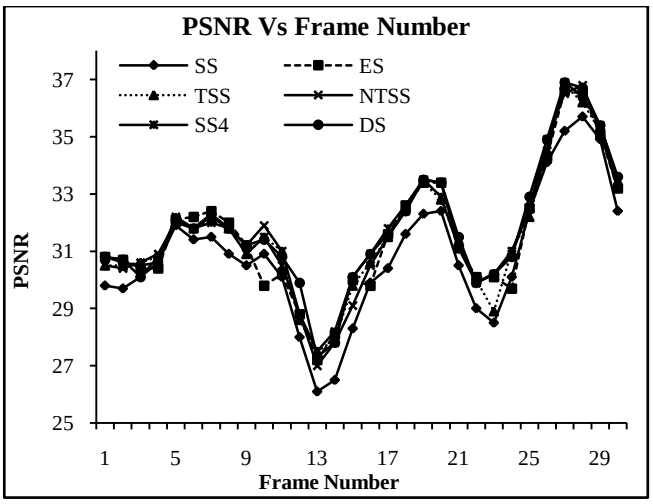

Fig.6(a). PSNR values of Foreman Sequence for Various Search Algorithms

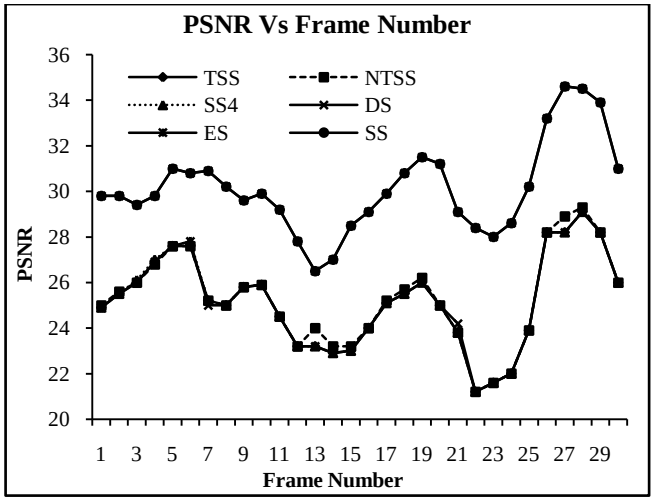

Fig.6(b). PSNR values of Mobile Sequence for Various Search Algorithms

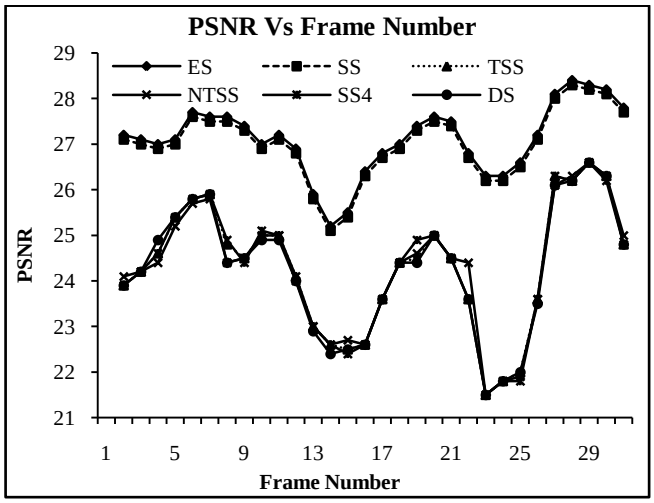

Fig.7(a). PSNR values of enhancement layer in mobile sequence for Bi-linear Interpolation

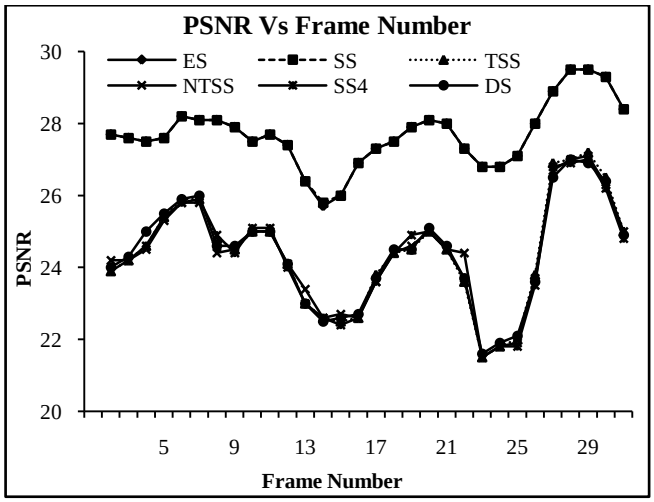

Fig.7(b). PSNR values of enhancement layer in mobile sequence for bi-cubic interpolation

Table.1. PSNR Values for Different Search Algorithms Using Bilinear Interpolation

Input Video	Average PSNR											
	ES		TSS		NTSS		SS4		DS		Proposed NMSS	
	BL	EL	BL	EL	BL	EL	BL	EL	BL	EL	BL	EL
Mobile	28.36	25.68	28.27	25.42	28.25	25.39	28.47	25.36	28.32	25.28	28.35	25.62
Foreman	30.17	27.77	29.97	26.77	29.92	26.76	29.76	26.54	29.84	26.49	30.08	27.59
Coastguard	29.59	25.68	29.47	25.62	29.45	25.64	29.47	25.63	29.48	25.62	29.56	25.67
News	30.07	26.27	29.97	26.13	29.96	25.19	29.96	25.1	30.12	25.12	30.05	26.22
(* BL – Base Layer, EL – Enhanced Layer)												

Table.2. PSNR Values for Different Search Algorithms Using Bi-Cubic Interpolation

Input Video	Average PSNR											
	ES		TSS		NTSS		SS4		DS		Proposed NMSS	
	BL	EL	BL	EL	BL	EL	BL	EL	BL	EL	BL	EL
Mobile	28.36	25.72	28.27	25.47	28.25	25.41	28.47	25.35	28.32	25.31	28.35	25.65
Foreman	30.17	27.79	29.97	26.75	29.92	26.79	29.76	26.59	29.84	26.51	30.08	27.62
Coastguard	29.59	25.69	29.47	25.68	29.45	25.66	29.47	25.66	29.48	25.65	29.56	25.71
News	30.07	26.42	29.97	26.17	29.96	25.21	29.96	25.14	30.12	25.18	30.05	26.24

4. RESULTS AND DISCUSSIONS

In this work different search algorithms, such as Exhaustive Search[ES], Diamond Search[DS], Three Step Search[TSS],New Three step Search[NTS], Spiral Search[SS], and Four Step Search[SS4] have been studied to check their suitability to the Scalable Video Coding. The simulation of the algorithms is done on standard test sequences which include Mobile and Foreman video sequence with a distance of 2 between current frame and reference frame was used to generate the frame-by- frame results of the algorithms.

As shown in the Fig.6(a) the PSNR values of the motion compensated image for Foreman Sequence in the Base Layer (QCIF) remains almost same for all the algorithms. Similarly in case of the Mobile Video Sequence as shown in Fig.6(b) the PSNR values do not show any significant variation with type of the algorithm used. The quantitative analysis of the reconstructed image is done using the Peak Signal to Noise Ratio given by,

$$PSNR = 20\log\left(\frac{PeakPixelValue}{RootMeanSquareError}\right)$$

The PSNR values of the reconstructed frames for two different input video sequences i.e. Foreman Sequence and Mobile Sequence have been tabulated in the Table.1 and Table.2 respectively. It has been observed that the Exhaustive search algorithm shows good performance in the enhancement layer when compared with other algorithms. The average PSNR each frame for SS and proposed NMSS algorithm as shown in Table.3. It is observed that proposed method has small reduction in PSNR. Table.4 lists average number of computations for macro block for various algorithms for mobile and foremen sequences. It is observed that proposed NMSS achieved 72% reduction in computations (SAD) per macroblock in a search range of -16 to +15 compared to SS method. It is also observed that DS, TSS, and FSS algorithms indicating less number of computations compared to proposed NMSS. But these algorithms performance is poor as shown in Table.1 and Table.2. These algorithms suffer from local minima problem. Proposed algorithm data flow is regular and hence hardware implementation becomes simple. Because of spiral scanning more data will be reused. Memory bandwidth becomes less and performance of the architecture will be improved compared to other algorithms.

Hence proposed method has superior performance both in terms performance and computations.

Table.3. Performance (PSNR) of SS and NMSS

Video Sequence	SS (PSNR)	NMSS (Proposed)
Mobile	26.93	26.67
Foreman	27.09	26.65
Coast guard	26.73	26.23
News	31.57	31.43
Bus	26.16	25.84

Table.4. Average number of computations

Search Algorithm	Average Computations per macroblock	
	Mobile Video (QCIF)	Foreman Video (QCIF)
Exhaustive Search	179	178.5
Diamond Search	19.45	27.11
Three Step Search	19.42	27.09
Four Step Search	19.51	27.05
Spiral Search	173.5	173.2
New Modified Spiral Search (NMSS) (Proposed)	48.9	48.8

Computation for each of the search algorithms has been given in the Table.4. Further, Fig.7(a) and Fig.7(b) shows the Bilinear and Bi-cubic interpolated image PSNR vs frame number in Enhanced layer subject to smoothing to remove the blocking effects using the spatial filters, average filter and Median filter. It is observed that TSS and ES have shown better performance compared to other algorithms for mobile video test sequence.

5. CONCLUSION

In this paper different search algorithms, such as Exhaustive search, Diamond search, Three Step Search, New Three Step Search, Four Step Search and Spiral Search algorithms, are implemented for didactic scalable video coding model. Exhaustive Search is computationally more expensive, but it gives high image quality and data flow is regular. The effect is seen more significantly in the enhancement layer after the interpolation. The proposed New Modified Spiral Search algorithm allows reduction in the number search points based on the SAD, unlike most of the fast motion estimation algorithms. It makes efficient utilization of the Spiral Scan through Adaptive early termination. Different interpolation algorithms have been studied with respect to their performance in terms of PSNR in the base layer and Enhancement Layer. The proposed New Modified Spiral Search algorithm achieves 72 % reduction in computation with a minimal (<1 dB) reduction in PSNR in the Enhanced Layer compared with the traditional Spiral Search Algorithm.

REFERENCES

[1] ITU-T Recommendation, "*H.263 - Video coding for low bit rate communication*", 1996.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, pp. 560-576, 2003.

[3] Seung-Man Pyen, Kyeong-Yuk Min, Jong-Wha Chong and Satoshi Goto, "An Efficient Hardware Architecture for Full-Search Variable Block Size Motion Estimation in H.264/AVC", *International Symposium on Visual Computing*-2006, *Lecture Notes on Computer Science*, pp. 554-563, 2006.

[4] Jong-Nam Kim and Tae-Sun Choi, "A fast three-step search algorithm with minimum checking points using unimodal error surface assumption", *IEEE Transactions on Consumer Electronics*, Vol. 44, No. 3, pp. 638-648, 1998.

[5] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 3, pp. 313-317, 1996.

[6] M. F. So and A. Wu, "Four-step genetic search for block motion estimation", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3, pp. 1393-1396, 1998.

[7] R. Li, B. Zeng and M. L. Liou, "A new three-step search algorithm for block motion estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 4, pp. 438-442, 1994.

[8] 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. 292-296, 1997.

[9] C. Zhu, X. Lin and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 5, pp. 349-355, 2002.

[10] C. Zhu, X. Lin, L. Chau and L.-M. Po, "Enhanced hexagonal search for fast block motion estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 10, pp. 1210-1214, 2004.

[11] M. Ghanbari, "The cross-search algorithm for motion estimation", *IEEE Transactions on Communications*, Vol. 38, No. 7, pp. 950-953,1990.

[12] N. Kroupis, M. Dasygenis, K. Markou, D. Soudris and A. Thanailakis, "A modified spiral search motion estimation algorithm and its embedded system implementation", *IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 3347-3350, 2005.

[13] J. A Parker, R. V Kenyon, D. E Troxel, "Comparison of interpolating methods for image resampling", *IEEE Transactions on Medical Imaging*, Vol. MI-2, No. 1, pp. 31-39, 1983.

[14] D. D. Muresan and T. W. Parks, "New image interpolation techniques", *Proceedings of IEEE 2000 Western New York Image Processing Workshop*, 2000.

[15] T. Blu, P. Thevenaz and M. Unser, "Linear interpolation revitalized", *IEEE Transactions on Image Processing*, Vol. 13, No. 5, pp. 710-719, 2004.

[16] Heather studley and Keith T. Weber, "Comparison of Image resampling Techniques for Satellite Imagery", *GIS Training and Research Center Final Report*, pp. 185-196, 2011.

[17] Avinash Nayak, Bijayinee Biswa and S. K. Sabut, "Evaluation and Comparison of Motion Estimation Algorithms for Video Compression", *International Journal of Image, Graphics and Signal Processing*, Vol. 5, No. 10, pp. 9-18, 2013.

[18] P. Muralidhar, C. B. Rama Rao and I. Ranjith Kumar, "Efficient Architecture for variable block size Motion Estimation in H.264 Video Encoder", *International Conference on Solid-State and Integrated Circuit*, Vol. 32, 2012.