Fast Motion Estimation for Quad-Tree Based Video Coder Using Normalized Cross-Correlation Measure

Eskinder Anteneh Ayele *eskinderanteneh@yahoo.co.uk*
Research Scholar/ Department of Electronics Engineering
Visvesvaraya National Institute of Technology
Nagpur,440022, India

R. E. Chaudhari *rec77@rediffmail.com*
Asst. Professor/Dept. of ECE
St. Francis Institute of Technology
Mumbai,400103, India

S. B. Dhok *sbdhok@vnit.ece.ac.in*
Asso. Professor/Department of Electronics Engineering
Visvesvaraya National Institute of Technology
Nagpur,440022, India

Abstract

Motion estimation is the most challenging and time consuming stage in block based video codec. To reduce the computation time, many fast motion estimation algorithms were proposed and implemented. This paper proposes a quad-tree based Normalized Cross Correlation (NCC) measure for obtaining estimates of inter-frame motion. The measure operates in frequency domain using FFT algorithm as the similarity measure with an exhaustive full search in region of interest. NCC is a more suitable similarity measure than Sum of Absolute Difference (SAD) for reducing the temporal redundancy in video compression since we can attain flatter residual after motion compensation. The degrees of homogeneous and stationery regions are determined by selecting suitable initial fixed threshold for block partitioning. An experimental result of the proposed method shows that actual numbers of motion vectors are significantly less compared to existing methods with marginal effect on the quality of reconstructed frame. It also gives higher speed up ratio for both fixed block and quad-tree based motion estimation methods.

Keywords: FFT, Motion Estimation, Normalized Cross Correlation, Quad-tree, Video Compression.

1. INTRODUCTION

Motion estimation and compensation are the two crucial processes in block based video coding standards. A major technique known as motion estimation is used to compress the videos by removing the redundant information form successive frames. Inter-prediction explores temporal redundancy between frames to save coding bits [1]. By using motion compensated prediction, the best matching position of current block is found within the reference picture so that only prediction difference needs to be coded. Each prediction unit coded using inter-prediction, has a set of motion parameters, which consists of a motion vector, a reference picture index, and a reference list flag. Motion estimation is widely used in various applications related to computer vision and image processing, such as object tracking, object detection, pattern recognition and video compression, etc. Especially, block-based motion estimation is very vital for motion-compensated video compression, since it reduces the data redundancy between frames to achieve high compression ratio. Because of the high redundancy that exists between the consecutive frames of a video image sequence, a current frame can be reconstructed from a previous reference frame and the difference between the current and previous frames by using the motion information.

The idea behind block matching is to divide the current frame into a matrix of macro blocks that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. International standards for video communications such as MPEG-1/2/4 and H.261/3/4 employ motion compensation prediction which is based on regular (fixed- or near-fixed-size) block-based partitions of incoming frames. While such partitions require a minimal amount of overhead information they provide little or no adaptation to picture content. A notable departure from this practice has been the recently emerged H.264 standard which allows a degree of flexibility in the choice of block size. Motion estimation based on quad-tree partitions achieves a good balance between a degree of adaptation to picture content on one hand and low-complexity, low-overhead implementation on the other [2].

Block matching algorithms used for motion estimation in video compression differ in the matching criteria (e.g. Mean Square Error (MSE), SAD, cross-correlation), the search strategy (e.g. Full Search, Three Step Search, Four Step Search etc.), and the determination of block size (e.g. hierarchical, adaptive) [3]. In this paper, we adopt the normalized cross-correlation methodology and employ it in the framework of a quad-tree motion estimation scheme that provides a level of adaptation to picture contents without incurring substantial overheads. Our approach lies in the combination of these two concepts, namely quad-tree decomposition and cross correlation applied in the frequency domain using Fast Fourier Transform (FFT) algorithm, yielding partitions for which a monotonic decrease of the motion compensated prediction error.

Upon this introductory section, the rest of the paper is organized as follows: In Section 2, we briefly review the block partitioning principles underlying quad-tree partitioning of a frame for motion estimation. In Section 3, we formulate our quad-tree FFT-based normalized cross-correlation algorithm approach. In Section 4, we present the experimental/simulation results with a brief observational discussion. At the end, Section 5 concludes the paper.

2. QUAD TREE PARTITIONING

The proposed motion estimation scheme involves the quad-tree partitioning of a frame which provides a better level of adaptation to scene contents compared to fixed block size approaches. Quad-tree decompositions are achieved by using the motion compensated prediction error to control the partition of a parent block to four children quadrants [1] [4] [5] [6].

Figure 1a demonstrates a quadtree structure implemented in this paper. Here, we employed four level of quad-tree partitioning with block sizes, 32x32, 16x16, 8x8, and 4x4 pixels in which the macro block (32x32) can be partitioned into four 16x16. A sub-macro block (16x16) can be further partitioned to four 8x8 blocks and finally an 8x8 block can be also portioned into four 4x4 blocks as quadtree structure depending on threshold. The algorithm is implemented along the recursive raster scanning path [7], which has been traditionally used in the quadtree decomposition as can be seen from Figure 1b. To simplify the search for a good scanning path, we require that child blocks, which belong to the same parent block, are scanned in sequence. Therefore a high correlation between successive blocks will result in an efficient encoding.

A motion vector is generated for each block after a search is conducted to best match the movement of each block from a previous reference frame. However, large block-sizes generally produce poor motion estimation, thereby producing a large motion-compensated frame difference (error signal). Conversely, small block-sizes generally produce excellent motion estimation at the cost of increased computational complexity and the overhead of transmitting the increased number of motion vectors to a receiver. Thus, the balance between high motion vector overhead and good motion estimation is the focus of a quadtree-based variable block-size motion estimation method. The scheme is based on normalized cross correlation and uses key features of the cross correlation to control the partition of a parent block to four children quadrants.

2.1 Partition Criteria

In this paper, different sizes of block partition are used to minimize the number of motion vectors to be sent. This is based on the assumption that the higher the degree of homogeneous and stationary blocks, the larger is the block partition used. However, the thresholds to determine the degree of homogeneity are empirically selected, and the resulting criteria cannot provide very accurate block partitioning.

Initially the current frame is divided into non-overlapping blocks of size 32x32. The first threshold th1 is decided based on the output video quality of homogeneous and stationary regions and tested at the same location in the previous reconstructed frame. If the error exceeds the threshold th1, the bigger size of macroblock is partitioned into four quadrants of size 16x16. For higher levels another threshold is applied at four different cases. Results show that, the scheme provides a better level of adaptation to scene contents and outperforms fixed block size scheme in terms of different number of motion vectors for the same level of motion compensated prediction error and Structure Similarity (SSIM). The partition criterion also guarantees a monotonic decrease of the motion compensated prediction error with an increasing number of iterations [8].

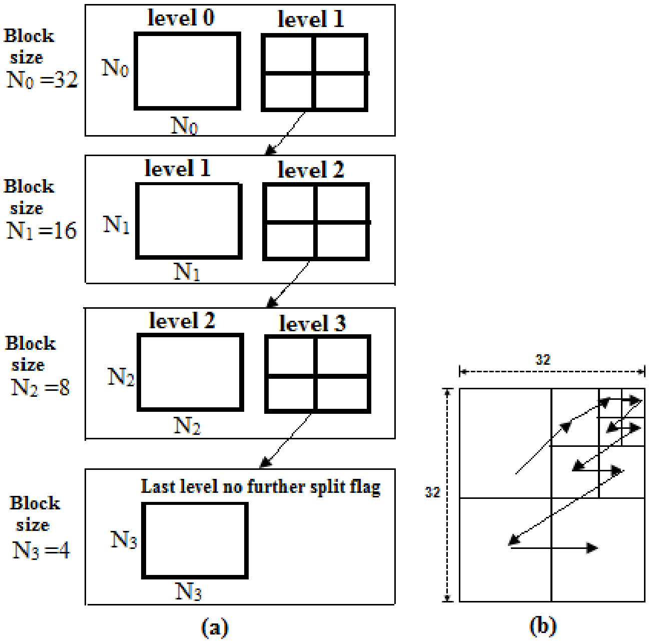

FIGURE 1: (a) A Quad tree Structure, (b) Recursive raster scan for Quad tree decomposition.

3. FFT-BASED NORMALIZED CROSS-CORRELATION

Many block-based motion estimation algorithms have been proposed and developed for finding the block with the smallest matching error including, phase-domain methods [9], time/space-domain methods [10], and spline-based methods [11]. Time-domain (1-D) or space-domain (2-D) methods have been widely and frequently used because of their high accuracy, precision, and resolution, and relative simplicity in implementation [12]. In terms of block distortion measure, the SAD is commonly used. In addition to SAD, the NCC is also a popular similarity measure. The NCC measure is more dynamic than SAD under uniform illumination changes. NCC can improve subjective visual quality as well as coding efficiency in video compression [2]. However, the NCC is a more complex criterion compared to SAD. SAD is used to find the best match with the lowest matching error and NCC is to find the best macro block whose overall intensity variation is most similar to current macro block. Though the error of NCC for motion estimation is larger, but it is more uniformly distributed than SAD based. These flat error results in large DC term and smaller AC term DCT coefficients, which mean less information loss after quantization.

One of the main motivations for this paper has been the current interest in motion estimation techniques operating in the frequency domain. These are commonly based on the principle of cross correlation and offer well-documented advantages in terms of computational efficiency due to the employment of fast algorithms [1]. Correlation is widely used as an efficient similarity measure in matching tasks. However, traditional correlation based matching methods are limited to the short baseline case. NCC is the most robust correlation measure for determining similarity between points in two frames (images) and provides an accurate foundation for motion estimation. However, implementing directly in spatial domain is too computationally intense especially for rapidly managing several large frames [2]. A significantly faster method of calculating the NCC is presented using FFT method to speed up block matching for computationally efficient video encoding.

3.1 The Algorithm

The best match is defined in terms of NCC [13] by shifting a macro block pixel by pixel across the search window. Correlation planes provide information where the macro block best match the search window. The correlation coefficient conquers the difficulties in [14, 15] by normalizing the current and reference frame vectors to unit length, yielding a cosine-like correlation coefficient. It is defined in spatial domain as:

$$NCC(x,y) = \frac{c(x,y)}{\|T(x,y)\|.\|I(x+i,y+j)\|} \qquad (1)$$

The pixel location (x, y) corresponding to the maximum NCC value matches to best location (motion vector) of MB in the search window. At which, x Є {0, 1, ..., N-M} and y Є {0, 1, … ,N-M}. For example if N =24 and M = 8, the number of NCC coefficients are (17x17).

Where, c (u, v) is the cross-correlation, T is the current macro block of size MxM, and I is the search window of reference frame of size NxN (N>M). The norms of the current and the reference frame in (1) are defined, respectively, as follows:

$$\|T(x,y)\| = \sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} T(i,j)^2}$$

$$\|I(x+i,y+j)\| = \sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I(x+i,y+j)^2}$$

A significantly more efficient way of calculating the NCC is by computing the numerator of (1) via FFT because the calculation of the numerator dominates the computational cost of the NCC. More specifically, cross-correlation in the spatial domain, which is the numerator in (1), is equivalent to multiplication in the frequency-domain:

$$\text{i.e } c(x,y) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} T(i,j).I(x+i,y+j)$$

$$\Rightarrow C(u,v) = T(u,v)I(u,v)$$

$$c(x,y) = \Im^{-1}(C(u,v)) \qquad (2)$$

Basically Equation (2) corresponds to computing a 2D FFT on the current and the search window of the frames followed by a complex-conjugate multiplication of the resulting Fourier coefficients. However, in order to avoid a complex-conjugate multiplication, we computed the current frame macro block via IFFT as shown in Figure 2. The final products are then inverse Fourier transformed to produce the actual coefficient cross correlation plane. The use of FFT in numerator calculations of (1) is used to reduce the number of NCC calculations.

Similarly the denominator calculations are performed by pre-computing the energy of the entire searching windows of a frame. Whenever the search window moves from block to block with respect to current macro block, we change the location of energy block which is stored as a look-up table. Thus it further reduces the computation complexity of the algorithm.

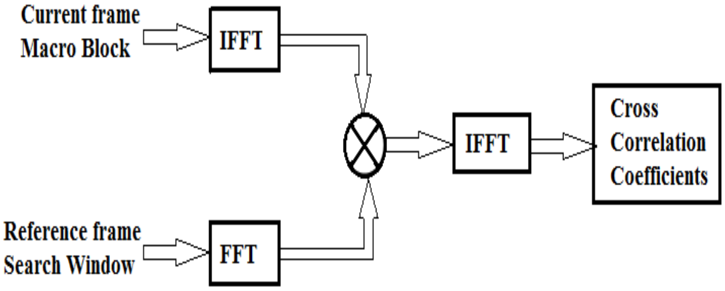

FIGURE 2: Implementation of the numerator of NCC by using FFT algorithm.

4. OBSERVATION AND RESULTS

As performance measure parameters, MSE and Peak Signal to Noise Ratio (PSNR) are used to evaluate the subjective quality of a reconstructed video sequence. Applying the NCC as the matching criterion to motion estimation leads to more uniform residuals. Hence, the NCC can improve subjective visual quality as well as coding efficiency in video compression. Recently, visual quality measures focusing on the human visual system (HVS) have been devised in place of PSNR. Among these measures, SSIM has become popular. The SSIM index is more consistent with human perception and is designed to measure structural information degradation, including the three comparison points of luminance, contrast, and structure.

Apart from the prediction error criterion, computational complexity is also a key criterion for the performance evaluation of fast block matching algorithms. The computational complexity can be directly compared by counting the number of searching points required. The number of searching points is a measure of search speed whereas the computation time is another speed measure that also takes into account the overhead of the algorithm. The overhead includes time spent on storing and fetching spatio-temporal predictors, making comparisons etc. Hence in general the computation time is a better measure of determining computational speedup.

The experimentations are performed on five standard video sequences with frame size of 288x352 through four different threshold cases for quadtree partitioning: case-I (10, 16), case-II (12, 18), case-III (15, 21) and case-IV (18, 24). For each case (th2, th3), the threshold th2, th3 are randomly chosen in the second and third levels of partitioning respectively to get minimum number of Motion Vectors (MVs). Based upon visual quality observation from the simulated result, we fixed the first level threshold =10 for the bigger size of macro block (32x32) as a stationary (static) block. By keeping this threshold, we determined the PSNR and SSIM results of the reconstructed videos, which are similar to fixed block result with less number of bits. The corresponding average numbers of static blocks and bits per block is shown in table-1.

Video Sequences	Avg. No. of static blocks	Avg. no. bits per statics blocks
Foreman	51.20	2.50
Paris	3.58	3.00
Carphone	61.58	2.00
Tennis	66.18	2.54
News	94.30	1.00

TABLE 1: Average numbers of Static/Stationary blocks and bits per block of the test videos

Tables 2 shows simulations of the average PSNR, SSIM, and encoding times of different videos for fixed block size (8x8) using SAD and FFT-based NCC. All simulations were done on Matlab-7.9 using a Pentium 4 desktop with 3.0GHz CPU and 1.0GMb of RAM. The experimental results show that efficient FFT based NCC full search algorithm can provide slightly higher PSNR and better SSIM in the reconstructed frame than the traditional SAD-based fixed block ME method. Further it reduces the encoding time by more than 50%.

Video sequences	PSNR (dB)	SSIM	Encoding Time(Sec)
Using SAD			
Foreman	34.51	0.8995	3.87
Paris	30.98	0.9514	3.89
Carphone	39.29	0.9279	3.86
Tennis	29.94	0.8013	3.84
News	39.48	0.9309	3.89
Using FFT-based NCC			
Foreman	34.75	0.9042	1.50
Paris	31.02	0.9536	1.53
Carphone	39.68	0.9342	1.50
Tennis	29.97	0.8028	1.52
News	39.54	0.9290	1.48

TABLE 2: Comparisons of average PSNR, SSIM, and Encoding time for fixed block size (8x8)

Table 3 shows the comparison of the performance parameter viz. average number of motion vectors, encoding time, PSNR and SSIM for the Quadtree FFT-based NCC method with four threshold cases as mentioned above. The results show in all cases that the Quadtree FFT-based NCC method is just as accurate as SAD method but the Quadtree FFT-based NCC method faster than the standard SAD method. Depending on the type of the video sequences and the threshold levels, the method is about 2 to 5 times faster than SAD-based search criteria.

Figure 3 summarizes the number of motion vectors for the luminance components of the first 12 inter-frames of the ‘Tennis’ video sequence. The graph shows that the proposed algorithm is substantially dependent on threshold levels and it shows the variation of MVs from frame to frame.

Video Sequences	PSNR (dB)	SSIM	Time (Sec)	No. of Motion Vectors			
				32x32	16x16	8x8	4x4
Case-I							
Foreman	34.07	0.8852	1.28	51.36	151.95	131.73	98.55
Paris	30.83	0.9436	2.16	3.55	262.27	452.18	104.00
Carphone	38.82	0.9075	1.15	60.82	125.45	83.73	101.45
Tennis	29.64	0.7979	1.55	65.64	45.18	320.82	129.09
News	39.42	0.8967	0.77	93.36	15.45	21.00	29.45
Case-II							
Foreman	34.01	0.8851	1.14	51.45	165.45	82.91	64.00
Paris	30.78	0.9432	1.91	3.55	297.45	319.00	73.82
Carphone	38.80	0.9076	1.07	60.64	134.64	56.64	74.55
Tennis	29.92	0.7977	1.17	65.73	96.00	125.54	91.27
News	39.44	0.8967	0.70	93.36	17.09	16.45	21.45
Case-III							
Foreman	33.98	0.8849	1.03	51.45	177.73	40.91	35.64
Paris	30.78	0.9423	1.71	3.55	322.54	224.09	52.00
Carphone	38.89	0.9075	0.98	60.55	142.18	34.45	48.36
Tennis	29.87	0.7968	0.99	65.64	115.45	59.36	50.55
News	39.46	0.8967	0.67	93.36	18.91	11.73	11.27
Case-IV							
Foreman	33.97	0.8850	0.98	51.45	183.27	23.18	17.82
Paris	30.73	0.9419	1.60	3.55	339.64	156.45	49.09
Carphone	38.92	0.9074	0.94	60.55	146.91	20.09	30.18
Tennis	29.82	0.7961	0.92	65.18	126.54	25.73	36.73
News	39.47	0.8967	0.65	93.36	19.91	8.82	6.91

TABLE 3: Performance parameters for Quadtree FFT-based NCC method with four randomly selected threshold.

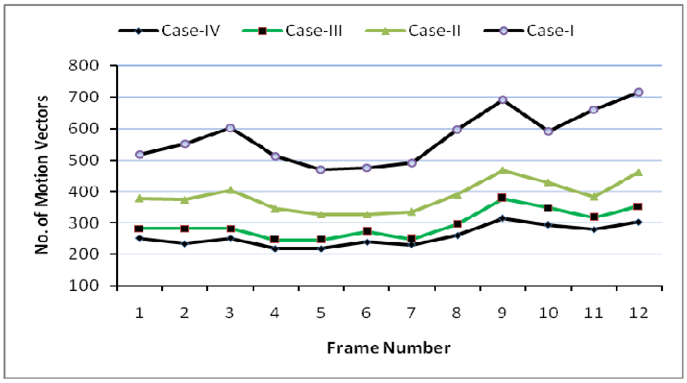

FIGURE 3: Number of MVs vs frame number for Tennis video at different thresholds.

The NCC can be often a better criterion than the SAD in terms of PSNR. In order to verify this, a full search based on SAD was compared to that based on NCC, where the search range and matching block size were fixed to ±8 pixels and 8×8 pixels in terms of integer-pel accuracy. Figure 4 shows the experimental results for 12 sequences of Tennis video. Here, the PSNR values are computed from the motion compensated version of the second frame of each sequence in order to evaluate the performance of the motion estimation only. The results show that the NCC provides slightly better PSNR performance than the SAD. This means that the motion compensated frames using NCC-based motion estimation are visually better than that those using SAD-based motion estimation in general.

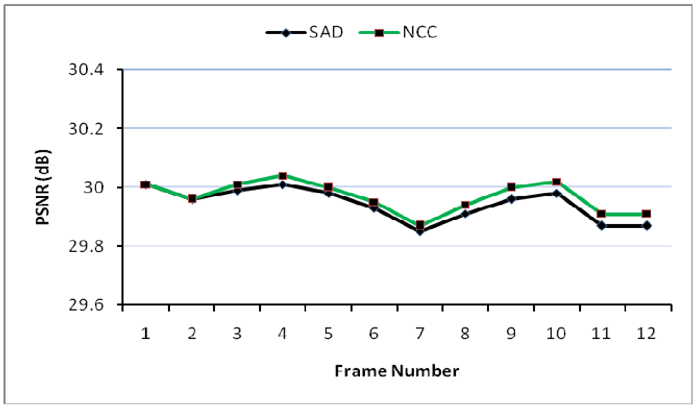

FIGURE 4: PSNR vs frame number using SAD and FFT-based NCC methods for Fixed Block Size.

Finally, the proposed algorithm (QT-NCC using FFT) was compared with Fixed Block (FB)-NCC and FB-SAD using same video frames. As seen from figure 5, the proposed algorithm can improve the speed-up ratio up to about 2.5 and 4.0 times in comparison with the FB-NCC and FB-SAD algorithms respectively but keeping SSIM and PSNR values almost the same for all algorithms.

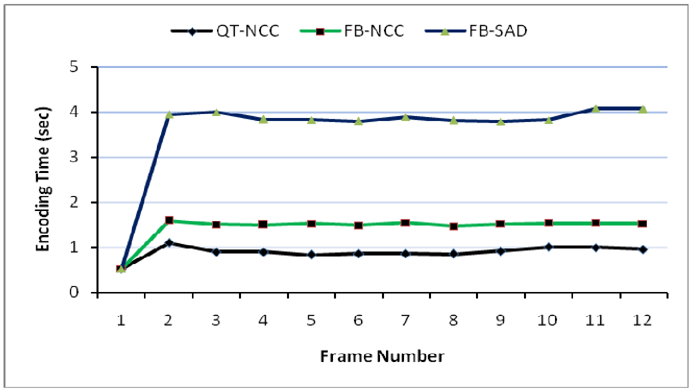

FIGURE 5: Encoding Time vs frame number for QT-NCC, FB-NCC, and FB-SAD algorithms.

5. CONCLUSIONS

This paper proposes a fast Quadtree FFT- based NCC, where re-using the energy part of search window is employed to skip unnecessary block-matching calculation and the cross correlation is determined in frequency domain based on FFT algorithm. Because of the quad-tree partitioning of a frame, it provides a better level of adaptation to scene contents compared to fixed block size approaches. Hence, the proposed algorithm considerably reduces the computational complexity and improves the speed-up ratio of about 4 times in comparisons with FB-NCC and FB-SAD algorithms. Moreover, for video sequences which contain more static data it requires less number of bits to encode without motion vector. For further quality improvement of reconstructed frames, one can use half or quarter pixel interpolation techniques. Correspondingly to enhance the speed ratio the algorithm can be modified and implemented using basis functions.

6. REFERENCES

[1] V. Argyriou and T. Vlachos, “Motion estimation using quad-tree phase correlation”, IEEE International Conference on Image Processing, 2005, vol. 1, pp. I-1081-I-1084.

[2] B. C. Song, “A Fast Normalized Cross Correlation-Based Block Matching Algorithm Using Multilevel Cauchy-Schwartz Inequality”, ETRI Journal, vol. 33, no.3, pp. 401-406, June 2011.

[3] A. Barjatya, “Block Matching Algorithms for Motion Estimation”, Digital Image Processing (DIP 6620) ,Final project paper, Utah State University, Spring 2004.

[4] G. J. Sullivan and R. L. Baker, “Efficient Quadtree Coding of Images and Video”, IEEE Transactions on Image Processing, vol. 3, issue 3, pp. 327-331, May 1994.

[5] V. Seferidis and M. Ghanbari, “Generalised Block-Matching Motion Estimation using Quad-Tree Structured Spatial Decomposition”, IEE Proceedings- Vision, Image and Signal Processing, vol. 141, issue 6, pp. 446-452, 1994.

[6] J. Lee, “Optimal quadtree for variable block size motion estimation”, IEEE International Conference on Image Processing, Oct. 1995, vol. 3, pp. 480-483.

[7] G. M. Schuster and A. K. Katsaggelos, “An Optimal Quadtree Based Motion Estimation and Motion-Compensated Interpolation Scheme for Video Compression”, IEEE Transactions on Image Processing, vol. 7, issue 11, pp. 1505-1523, Nov. 1998.

[8] V. Argyriou and T. Vlachos, ”Quad-Tree Motion Estimation in the Frequency Domain Using Gradient Correlation”, IEEE Transactions on Multimedia, vol 9, issue 6, pp. 1147-1154, Oct. 2007.

[9] C. Kasai, K. Namekawa, A. Koyana and R. Omoto “Real-Time Two-Dimensional Blood Flow Imaging Using an Autocorrelation Technique”, IEEE Transaction on Sonics and Ultrasonics, vol. 32, issue 3, pp. 458-464, May 1985.

[10] S. Langeland, J. D’hooge, H. Torp, B. Bijnens, and P. Suetens, “Comparison of Time-Domain Displacement Estimators for Two-Dimensional RF Tracking”, Ultrasound in Medicine and Biology, vol. 29, no. 8, pp. 1177–1186, 2003.

[11] F. Viola and W. F. Walker, “A spline-based algorithm for continuous time-delay estimation using sampled data”, IEEE Transactions on Ultrasonics. Ferroelectrics. Frequency Control, vol. 52, no. 1, pp. 80–93, 2005.

[12] J. Luo and E. E. Konofagou, “A Fast Normalized Cross-Correlation Calculation Method for Motion Estimation”, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 57, issue 6, pp. 1347-1357, June 2010.

[13] A. J. H. Hii, C. E. Hann, J. G. Chase, and E. E. W. Van Houten, “Fast Normalized Cross Correlation for Motion Tracking Using Basis Functions”, Journal of Computer Methods and Programs in Biomedicine, vol. 82, no. 2, pp. 144-156, 2006.

[14] P. Anandan, ”A Computational Framework and an Algorithm for the Measurement of Visual Motion”, International Journal of Computer Vision, vol. 2, pp. 283-310, 1989.

[15] S. D. Wei, W. H. Pan and S. H. Lai, “A novel motion estimation method based on normalized cross correlation for video compression”, Proceedings-14th International Multimedia Modeling Conference, MMM 2008, Kyoto, Japan, Jan. 2008, pp. 338-.347.