AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Fibonacci Spiral Block Matching Algorithm with DCT in Video Coding

[1]L.C.Manikandan and [2]Dr.R.K.Selvakumar, Senior Member IEEE

[1]*School of Computer Science and Engineering, Mar Ephraem College of Engineering and Technology, Kanyakumari, Tamilnadu.*
[2]*Department of Information Technology, AGNI College of Technology, Chennai, Tamilnadu.*

ARTICLE INFO

Article history:
Received 19 August 2014
Received in revised form
19 September 2014
Accepted 29 September 2014
Available online 12 November 2014

Keywords:
Block-Matching Algorithm, Fibonacci Spiral Search, H.264, Motion Estimation, Motion Vector, Video Coding.

ABSTRACT

In video coding schemes, block matching algorithm is the key element for motion estimation. Motion estimation can significantly improve video coding efficiency by reducing temporal redundancy existing in a video sequence. In this paper, we propose a new algorithm called "Fibonacci Spiral Block Matching Algorithm with DCT (FSBMADCT)" for fast block motion estimation. Our proposed algorithm is working with combined effect of spiral search, Fibonacci sequences with DCT for transformation. This incorporated matching algorithm emphasized on spiral searching points. The experimental results are compared with the various fast motion estimation algorithms that have been employed in H.264 video standard. The reduction in the number of search points significantly reduces the computational complexity and paved the way for speedy compression.

To Cite This Article: L.C.Manikandan and Dr.R.K.Selvakumar., Fibonacci Spiral Block Matching Algorithm with DCT in Video Coding. ***Aust. J. Basic & Appl. Sci.,*** *8(18): 117-121, 2014*

INTRODUCTION

Video coding is the process of compacting or shrinking a digital video sequence into a number of smaller bits. Video coding has become an essential part of modern multimedia systems, since it enables significant bit rate reduction of the video signal for transmission or storage. Motion Estimation (ME) is a significant part of any video coding system. A video sequence consists of a series of frames.

To achieve video coding, the temporal redundancy between adjacent frames can be exploited. That is, a frame is selected as a reference frame, and the subsequent frames are predicted from the reference frame using a technique known as Motion Estimation (Slijepcevic, S, Potkonjak). The general goals of motion estimation methods are to improve the prediction accuracy, or to reduce the implementation complexity, or a combination of the two. Block Matching Algorithm (BMA) is the method for motion estimation, has been extensively espoused by current video coding standards like H.261, H.263, MPEG-1, MPEG-2, MPEG-4, and H.264 due to its effectiveness and simplicity. Full search (FS) algorithm (S. Immanuel Alex Pandian) is the simplest BMA, which provides an optimal solution by fully search all candidates within the search window. But its high computational complexity made it difficult to be implemented in real-time software-based applications.

In order to reduce the computational complexity of FS, several fast BMA's have been developed like Three Step Search Algorithm (Renxiang Li), New 3-Step Search Algorithm (Renxiang Li), Cross-Search Algorithm (M. Ghanbari *et al*.,), 4-Step Search Algorithm (Lai-Man Po Wing-Chung Ma., Ms. Maya Gaikwad *et al*.,) Diamond Search Algorithm (Shan Zhu and Kai-Kuang Ma), Spiral Search (Deepak Turaga, Th.Zahariadis) and Fast-Adaptive Rood Pattern Search FARPS (B.G. Kim *et al*.,) etc. These different kinds of BMA's journey different search designs and search approaches for verdict the prime motion vector with significantly compact the amount of searching points as matched with FS algorithm.

In this paper, we propose a simple, competent and fast BMA, called Fibonacci Spiral Block Matching Algorithm with DCT (FSBMADCT). In the proposed algorithm, two important aspects are considered to speed up and minimize the number of search block. The search pattern and search strategy as proposed for performing fast block-matching ME.

2. Block Matching Algorithm:

Block Matching Algorithm (BMA) (Barjatya, 2004) is the standard method to estimate the motion vector of a block in the current frame. The BMA eradicates the temporal redundancy, which is originated predominantly

Corresponding Author: L.C.Manikandan, School of Computer Science and Engineering, Mar Ephraem College of Engineering and Technology, Marthandam, Kanyakumari, Tamilnadu.
E-mail: lcmanikandan@gmail.com

Australian Journal of Basic and Applied Sciences, 8(18) December 2014, Pages: 117-121

in any video sequence. The size of the block affects the performance of motion estimation. Small block sizes afford good approximation to the natural object boundaries; they also provide a good approximation to real motion. However, small block sizes produce a large amount of raw motion information, which increases the number of transmission bits or the required data compression complexity to condense this information. Small blocks also suffer from object (block) ambiguity problem and the random noise problem. Large block sizes may produce less accurate motion vectors, since a large block may likely contain pels moving at different speeds and directions. The idea overdue block matching is to divide frames into equal sized non-overlapping blocks and compute the displacement of the best-matched block from the prior frame with the motion vector of the block in the current frame within the search window. This basic operation is repeated until all the candidates have gone through and then the best matches candidate is identified. The location of the best matched candidate forms the estimated displacement vector. The computational load could also be reduced by calculating fewer blocks of an image. To increase search efficiency, we could place the initial search point at a location predicted from motion vectors of the spatially or the temporally adjacent blocks. A best matching can often be obtained by searching a smaller region surrounding this initial point.

We could first separate the moving image blocks from the stationary ones and then conduct block matching only on the moving objects. This is because a moving or change detector can be implemented with much fewer calculations than a motion estimator. During block matching, each target block of the current frame is matched with a prior frame in order to discover the best matching block. BMA's calculate the best match using Mean Absolute Difference (MAD) (Th.Zahariadis and D.Kalivas) or any other distance measurement standards. Fig. 1 shows the general block matching concepts.

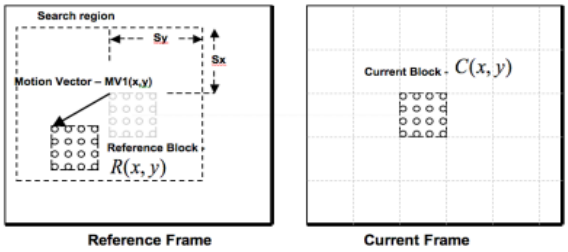

Fig. 1: Block Matching Concept.

2.1 Matching Criteria:

The MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables. For a set of data, the equation for a least square line is determined by Sum of Squares for Error (SSE). The average error of each square is calculated in order to determine the Mean Squared Error (MSE) of a frame. Matching Criteria for video coding are Mean Absolute Error (MAE)(M. Ghanbari *et al*.,), Sum of the Squared Error (SSE), and Mean Square Error (MSE). They are defined in equation (1), (2) and (3).

$$\mathrm{MAE}(m,n) = \frac{1}{MN}\sum_{i=i_0}^{i_0+M}\sum_{j=j_0}^{j_0+N}[F_t(i,j) - F_{t-n}(i+m, j+n)] \quad (1)$$

$$\mathrm{SSE} = \sum_{n=1}^{N}(I_n(x,y) - I'_n(x,y))^2 \quad (2)$$

$$\mathrm{MSE} = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[I(x,y) - I'(x,y)]^2 \quad (3)$$

2.2 Quality Measure Criteria:

Peak Signal to Noise Ratio (PSNR) is measured on a logarithmic scale and depends on the mean square error (MSE) between an original and an impaired image or video frame. PSNR is a very popular quality measure, widely used to compare the quality of compressed and decompressed video images. The formula for calculating PSNR is:

$$\mathrm{PSNR} = 10 \times \log_{10}(255^2/\mathrm{MSE}) \quad (4)$$

3. Fibonacci Spiral Block Matching Algorithm with DCT (FSBMADCT):

3.1 Fibonacci sequence:

The Fibonacci numbers are defined recursively, meaning the value of the nth Fibonacci number depends on the value of previous Fibonacci numbers. The nth Fibonacci number is denoted Fn. The values of the Fibonacci numbers are: F1 = 1, F2 = 1 and Fn = Fn-1 + Fn-2; for n = 3; 4; 5; : : : : For example, F3 = F3-1 + F3-2 = F2 + F1 = 2, and F4 = F3 + F2 = 3. From basic definitions of Fibonacci Numbers F[n], the recurrence relation F[n+1]=F[n]+F[n-1] with initial conditions F[0]=0 and F[1]=1. It is noticed that each number is generated just the sum of the two numbers preceding it. The Fibonacci sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, etc. That is, after the second term, each term is obtained by adding the previous two terms. The1 is found

Australian Journal of Basic and Applied Sciences, 8(18) December 2014, Pages: 117-121

by adding the two numbers before it (0+1). Similarly, the 2 is found by adding the two numbers before it (1+1), and the 3is (1+2), and so on.

3.2 The Fibonacci Spiral Search (FSS) Method:

Fibonacci Spiral can be constructed from an appropriate set of squares. Each block will fit nicely along the edge of the two previous blocks since $F_n = F_{n-1} + F_{n-2}$Given a table of records R1, R2, ..., RN whose keys are in increasing order K1 < K2 < ... < KN, the algorithm searches for a given argument K. Assume N+1 = Fk+1

Step1: [Initialize] i ← Fk, p ← Fk-1, q ← Fk-2 (throughout the algorithm, p and q will be consecutive Fibonacci numbers)

Step 2: [Compare] If K < Ki, go to Step 3; if K > Ki go to Step 4; and if K = Ki, the algorithm terminates successfully.

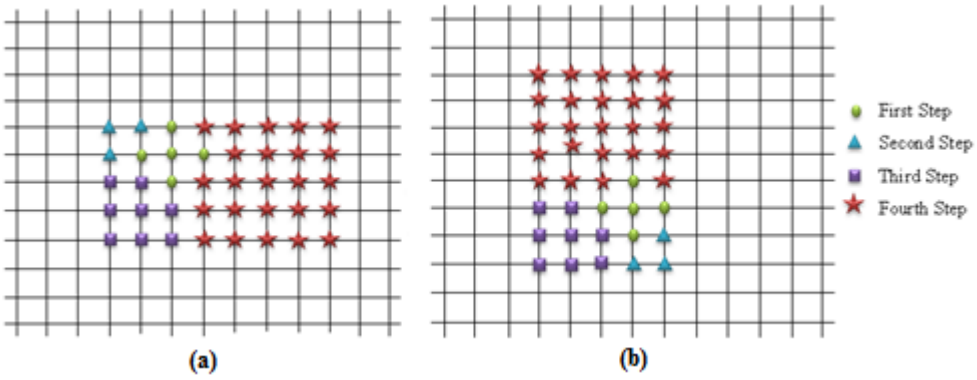

Fig. 2: (a) FSS Anticlockwise directions (b) FSS Clockwise directions.

Step 3: [Decrease i] If q=0, the algorithm terminates unsuccessfully. Otherwise, set (i, p, q) ← (i - q, q, p - q) (which moves p and q one position back in the Fibonacci sequence); then return to Step 2

Step 4: [Increase i] If p=1, the algorithm terminates unsuccessfully. Otherwise set (i,p,q) ← (i + p, p ← p - q, q ← 2q - p) (which moves p and q two positions back in the Fibonacci sequence); and return to Step 2. It is necessary to keep it rotating around as you add blocks, so the shared edge is at the bottom, right, top and then left edge of the new square. Once the squares are drawn, start spiraling it out from the first square you drew. It should look something like the Fig. 2 (a) picture when completed. Fig. 2 (b) shows clockwise direction.

3.3 FSS Algorithm:

In the proposed algorithm FSS measure, speed and the number of searching block are considered. The standard search window size is (2 × w) + 1. The maximum search point of this algorithm is calculated by Fn×F(n+1). Where Fn is the nth element in Fibonacci sequence. For example n = 4, the maximum searching point is F4 x F5 = 15. Fig. 3 shows the Schematic Flow of FSS. In the whole block search process, the first step started with search point 1, if the block matched, then the process will stop else it would move to the next step in searching the direction by crosscheck BDM for the neighborhood four pixels. From which two minimum values will be evaluated and used in prediction of direction. Next level 2 x 2 search points will be used for MAD calculation and it will exceed up to the final level of Fibonacci series until which the block gets matched. In the worst criteria for the comparison of image m x n, total search points of 15 is more than enough to come out of the search loop. The search points are comparatively minimum to other block matching algorithms.

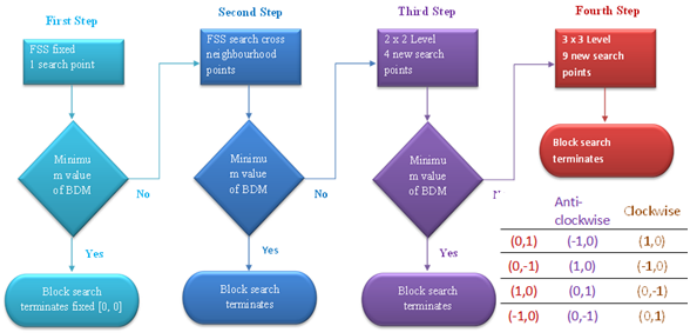

Fig. 3: Schematic Flows of FSS

Australian Journal of Basic and Applied Sciences, 8(18) December 2014, Pages: 117-121

3.4 Algorithm:

Step1: Current frame (cf) searching block is matched with a reference frame (rf) in the same block position. If it is matched, stop the searching process. Otherwise, it will match with 4 directional blocks.
Step2: In 4 directional blocks, if any one of the blocks is matched; stop the searching process else we will find the minimum level direction.
Step3: We will find the minimum of 2 values minimum2 and minimum1 from 4 directions based on this value we fix the direction either clockwise or anticlockwise.
Step4: Next, we will find the matching block with the next Fibonacci level 2 x 2.
Step5: In 2 x 2 levels, there are 4 search points, if any one of the search points is matched; stop the searching process. Otherwise, we will move to next Fibonacci level 3 x 3.
Step6: In 3 x 3 levels, we fix the optimal level and select the best matched optimal block and stop the searching process.

3.4. Distance Measurement:

Transform coding is extensively used in image coding but here we used for comparison. Discrete Cosine Transform (DCT) converts a block of image pixels into a block of transform coefficients of the same dimension (Ahmed, N et.al, 1974). These DCT Coefficients represent the original pixels values in the frequency domain. Any grey scale 8 x 8 pixel block can be fully represented by a weighted sum of 64 DCT basis functions where the weights are just the corresponding DCT Coefficients arranged as shown in the Fig.4. The current frame block is compared with the reference frame block within the search window using DCT.

DC	AC1	AC5	AC6	AC14	AC15	AC27	AC28
AC2	AC4	AC7	AC13	AC16	AC26	AC29	AC42
AC3	AC8	AC12	AC17	AC25	AC30	AC41	AC43
AC9	AC11	AC18	AC24	AC31	AC40	AC44	AC53
AC10	AC19	AC23	AC32	AC39	AC45	AC52	AC54
AC20	AC22	AC33	AC38	AC46	AC51	AC55	AC60
AC21	AC34	AC37	AC47	AC50	AC56	AC59	AC61
AC35	AC36	AC48	AC49	AC57	AC58	AC62	AC63

Fig. 4: DCT – 8 x 8 Block

4. Performance Efficiency:

The proposed FSBMADCT algorithm is simulated using the MPEG video sequences of first 50 frames of "Foreman" and "Football". The experimental results of proposed algorithm and other BMA's are summarized in Table 1(B.G. Kim et.al.,) for the video sequence "Foreman" and "Football".

Table 1: Simulation Results of FSBMADCT and Other Fast BMA's

Video Seq.	Algorithm	PSNR (db)	NSP/MB	Speed-up Ratio
Foreman	FS	37.98	225.00	1
	DS	37.67	17.08	13.17
	NTSS	37.33	22.22	10.12
	FSBMADCT	**32.18**	**2.41**	**96.68**
Football	FS	26.78	225.00	1
	DS	26.49	18.11	12.42
	NTSS	25.97	23.00	9.78
	FSBMADCT	**27.40**	**2.28**	**111.69**

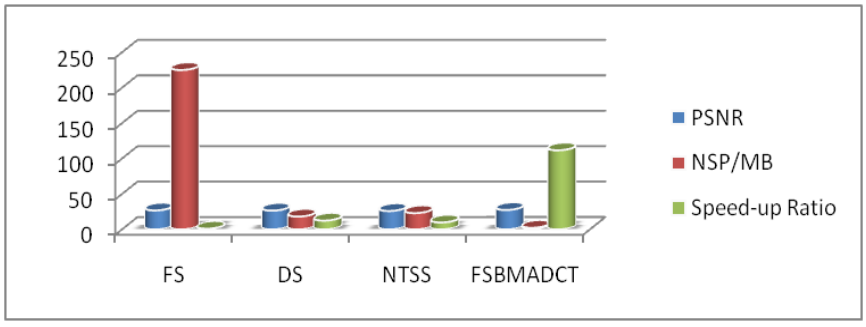

Fig. 5: PSNR, NSP/MB and Speed-up Ratio analysis of Foreman video.

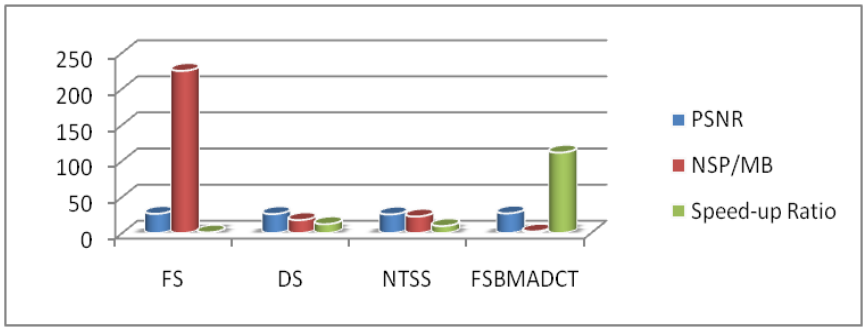

Fig. 6: PSNR, NSP/MB and Speed-up Ratio analysis of Football video.

From the table, it's clearly defined that FSBMADCT is predominant of FS, DS and NTSS block matching algorithms. The PSNR value are comparatively similar than FS, DS and NTSS BMA's. The Number of Search Points per Macroblock (NSP/MB) is lesser than FS, DS and NTSS block matching algorithms. It would directly impact in increasing the speed of block matching process and indirectly influence the compression to take place in a rapid way. Fig. 5 and Fig. 6 shows the performance comparison of FS, DS, NTSS and the proposed FSBMADCT for "Foreman" and "Football" sequence in terms of average PSNR.

Conclusion:

In this paper, search designs and search approaches of certain existing fast BMA's are examined. Based on these examines and annotations, Fibonacci Spiral Block Matching Algorithm with DCT (FSBMADCT) algorithm for motion estimation is developed. Our evaluation was based on two measures, speed up and minimize the number of searching block described the whole flow architecture is best suited in an optimized way. Generally FS algorithm provides better quality image (Minimum MSE and maximum PSNR) with less deviation from the reference frame. The proposed FSBMADCT algorithm effectively increases the searching process speed and provides similar or better quality. The FSBMADCT algorithm is implemented in H.264 video-encoding environment

REFERENCES

Ahmed, N, Natarajan, T and Rao, K R, January 1974, "Discrete Cosine Transform", IEEE Trans. On Computers, pp. 90-93.

Barjatya, 2004. "Block Matching Algorithms for Motion Estimation", Springer., DIP 6620.

Chandra Sekhar, C.H. and J.V.K. Ratnam, "Comparison of Fast Block Matching Algorithms for Motion Estimation", In: Proc. International Journal of Electronics and Computer Science Engineering, 1609-1618.

Deepak Turaga and Mohamed Alkanhal, 1998. "Search Algorithms for Block-Matching in Motion Estimation", Springer.

Ghanbari, M., 1990. "The Cross-Search Algorithm for Motion Estimation", IEEE Transactions on Communications, 950-953.

Immanuel Alex Pandian, S., 2011. "A Study on Block Matching Algorithms for Motion Estimation", In: Proc. International Journal on Computer Science and Engineering, 34-44.

Jain, J.R. and A.K. Jain, 1981. "Displacement measurement and its application in inter frame image coding", IEEE Trans. on Communications, 1799-1808.

Kim, B.G., 2005. "Fast-adaptive rood pattern search for block motion estimation", Electronics Letters

Koga, T., 1981. "Motion compensated inter frame coding for video conferencing", 3.1-5.3.5.

Lai-Man Po Wing-Chung Ma, 1996. "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Transactions on Circuits Syst. Video Technol, 313-317.

Metkar, S. and S. Talbar, 2013. "Motion Estimation Techniques for Digital Video Coding", In: Proc. Springer Briefs in Computational Intelligence.

Ms. Maya Gaikwad, 2012. "Implementation of four step search algorithm of motion estimation using FPGA", In: Proc. International Journal of Advanced Research in Computer Science and Electronics Engineering.

Renxiang Li, 1994. "Circuits And Systems For Video Technology, A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Trans, 438-442.

Shan Zhu and Kai-Kuang Ma, 2000. "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Image Processing, 287-290.

Th.Zahariadis and D. Kalivas, 1995. "A Spiral search algorithm for fast estimation of block motion vectors".