A New Block Matching Algorithm for Video Coding using Fibonacci Sequence

L.C.Manikandan[#1], Dr.R.K.Selvakumar[#2]

#1 Head &Asst.Professor of CSE, Mar Ephraem College of Engineering and Technology, Anna University, Marthandam, Kanyakumari, Tamil Nadu, India
#2 Head & Professor of IT, Agni College of Technology, Anna University, Chennai, Tamil Nadu, India

ABSTRACT

Through the literature survey several search designs and search approaches of certain existing fast Block Matching Algorithms (BMA) are examined. For inter-frame compression, block matching algorithm is the key element for motion estimation. In this paper, we propose a new block matching algorithm called "Fibonacci Spiral Search" for inter linked frames motion estimation with minimal search points. Our proposed algorithm is working with combined effect ofspiral search and Fibonacci sequences. This incorporated matching algorithm emphasized on spiral searching points. The search pattern and search strategy concepts are explained in the following sections.

Key words: Video coding, Block Matching algorithm, Motion Estimation, H.264, Motion Vector, Fibonacci Spiral Search.

Corresponding Author:L.C.Manikandan

INTRODUCTION

Digitized video is widely using in several consumer electronic applications like VCD, DVD, Video phone, Portable media player, video conferencing, e-learning etc. The data quantity of the digital video is very high. So it required more memory for storage and more channel bandwidth for the transmission, it induce the compression of digital video. Compression is the process of reducing or removing the amount of redundant spatial/temporal video datawhile maintaining as much of the original quality as possible. Digital video contains much spatial and temporal redundancy. In a single frame, nearby pixels are often correlated with each otheris called spatial redundancy or the intraframe correlation. Temporal redundancymeans adjacent frames are highly correlated orcalled the interframe correlation. Therefore, our goal is to efficiently reduce spatial and temporal redundancy to achieve video compression. One of the popular and effective methods to reduce or remove the temporal redundancy is called block-matching motion estimation [1], [2].

The redundancy factor can be resolved by many compression standards and these redundancy ranges from single rate to multi-rate video coders. So the compression codecs are implicitly creating impact on the video and audio transmission systems.Motion estimation [11]-[13] is the process of finding motion vectors that designate the transformation of pixel from one frame to another. How the inter-frame motion is adaptively segregated from the statutory pixels remaining in the frame entitled the importance of different motion estimation algorithms. Those motion estimation algorithms should recaptured the original resolution as in the conceptual video for compression. In motion estimation how minimal the search point is the laid by the fixation of search algorithms.

BLOCK MATCHING METHODS

The vital objective of block matching algorithm is used to interframe redundancy. For the resultant outcome the frame is splitted into non-overlapped blocks of equal size. The block in current in frame was in best way to search the most matched pixels in the previous frame. In order of that there are so many searching algorithms had implemented, the one which is Full Search BMA [3] but it considers only limited number of motion vectors and the complexity is more. Next come the Fast Full Search BMA, here using the basic three step search and major complexity are irregularity in data flow and overloading error will occur. Based on this algorithm the search points are more and takes data to be overloaded and able to complex the system.

We reviewed the following Block Matching Algorithms (BMA): Cross-Search Algorithm[5], Two Dimensional Logarithmic Search[7,2], Full Search Motion Estimation[3],Three Step Search Algorithm[4], New Three-Step Search Algorithm[4], Four-Step Search Algorithm[6, 9], Binary Search[7], Diamond Search Algorithm[9], Cross Diamond Search Algorithm[10], Hexagonal Search[11], Spiral Search[8] and other Block Matching Algorithms[1]-[15]. Based on this review we discussed the comparative study at end.

SPIRAL SEARCH ALGORITHM

The spiral search [8]algorithm was proposed by Zahariadis and Kalivas in 1995. It seeks to combine the ideas of the Three Step Search and the Binary search. By doing so, it tends to not only speed up the computation, but also removes the problem of the Binary search, where there is a zone of pixels that is never evaluated. Fig1 shows the Example path for convergence of Spiral Search.The algorithm may be described as follows:

Step 1: The step size is picked to half the maximum displacement in the search window. The point of minimum distortion is found from among the nine points picked in the following manner. Five points are picked in the shape of a "+" around the centre of the search window (at a distance of step size in the vertical and horizontal directions). The remaining four points are picked at the corners of the search window.

Step 2: The step size is reduced and a search is performed around the point with the smallest distortion. This is repeated till the step size falls to 1.

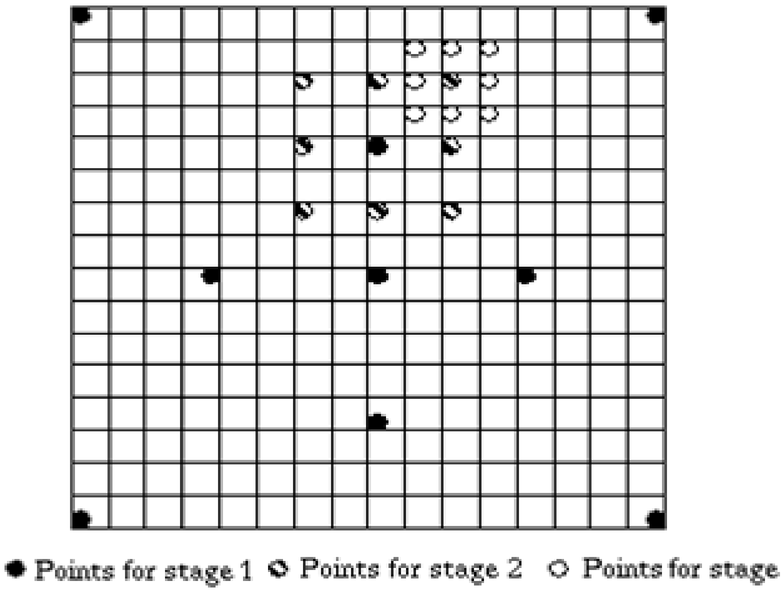

Fig1: Example path for convergence of Spiral Search

FIBONACCI SEQUENCE

Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding numbers. Beginning with 0 and 1, the sequence of Fibonacci numbers would be 0,1,1, 2, 3, 5, 8, 13, 21, 34, etc. using the formula n = n(-1) + n(-2), where the n(-1) means the last number before n in the series and n(-2) refers to the second last one before n in the series.From the Fibonacci sequence, number 2 is found by adding the two numbers before it (1+1). Similarly, the 3 is found by adding the two numbers before it (1+2), and the 5 is (2+3), and so on.

THE PROPOSED FIBONACCI SPIRAL METHOD

The proposed 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 $F_n = F_{n-1} + F_{n-2}$. It is necessary to keep rotating around as you add blocks so the shared edge is the bottom, right, top and then left edge of the new square. Once the squares are drawn, start spiralling out from the first square you drew. It should look something like the Fig 2 picture when you're done. Fig 3 shows clockwise direction.

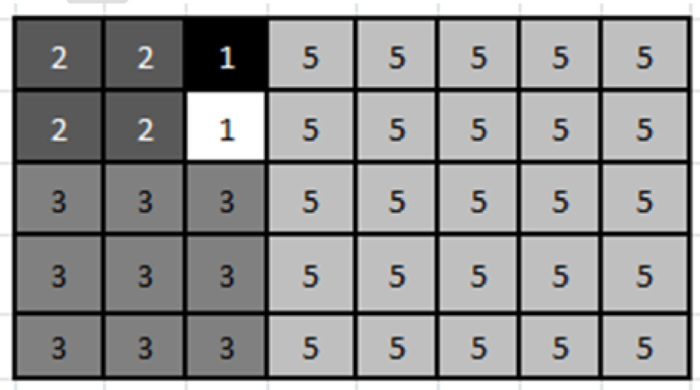

Fig 2: FSS Anticlockwise direction

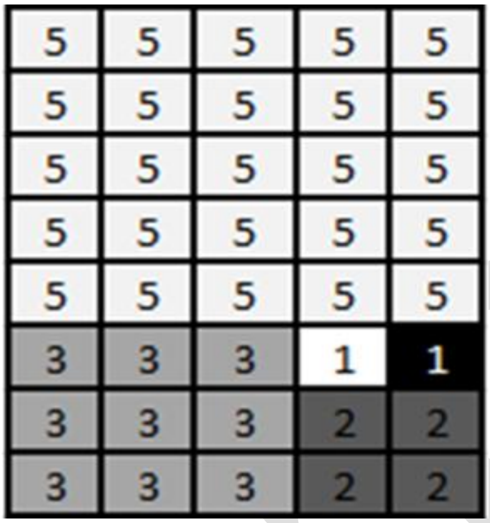

Fig 3: FSS Clockwise direction

FSS can be constructed from an appropriate set of squares when one draws circular arcs entered on one of the corners of each square. Starting with the black dot on the point where the boxes of sides 1 and 3 meet the edge of the box of side 2, connect the black dots in an outward spiral. This is the Fibonacci spiral. It should look something like the Fig 4 picture when you're done. Fig 5 shows FSS clockwise direction.

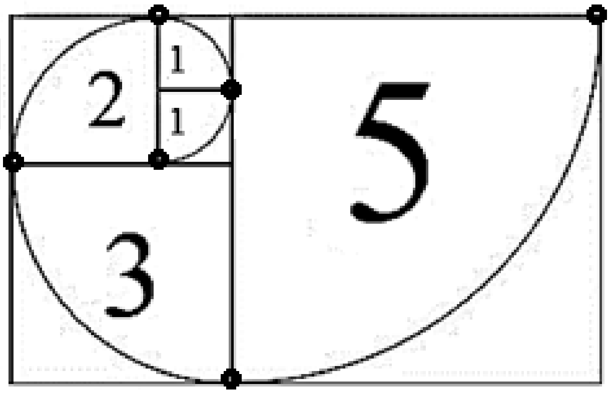

Fig 4: FSS Anticlockwise directions

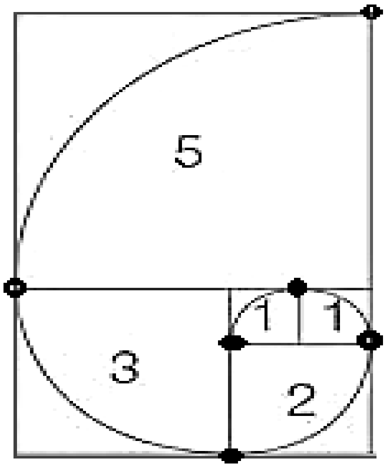

Fig 5: FSS Clockwise directions

ALGORITHM AND ANALYSIS

In the proposed algorithm, two important aspects are considered to speed up and improve the matching procedure. The search window size is calculated through $(2 \times w) + 1$. The maximum searching point of this algorithm is calculated by $F_n \times F_{n+1}$. Where F_n is the n^{th} element in Fibonacci sequence and n+1 is the next element in Fibonacci sequence. Suppose n=4 means, $F_n=F_4$, $F_{n+1}=F_5$, that is the value of F_4 is 3 and the value of F_5 is 5 in Fibonacci sequence. The mathematical analysis derived that Maximum 15 searching points are required for comparison. The summarization of algorithm is as follows.

Step1: Current frame (cf) searching block is matched with reference frame (rf) in 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 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 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 point 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.

We are presenting the table, which is comparing our proposed method with several block matching algorithms based on the searching time and the searching points in different window size. It proves that the searching time is less compared with other searching algorithms. The Table 1 shows the computational complexity of the searching points.

Table 1. Comparison Table – Maximum Searching Points

Algorithm	**Maximum Searching Time**	**Maximum number of search points – Window Size W = 4, 8, 16**

		4	8	16
FS	$(2w + 1)^2$	81	269	1089
CSA	$5 + 4 \log_2 w$	13	17	21
TDL	$2 + 7 \log_2 w$	16	23	30
BS	$9+\max\{24,14,8\}*\log_2 w$	57	81	105
TSS	$1 + 8 \log_2 w$	17	25	33
NTSS	$[1+8 \log_2 w] + 8$	25	33	41
ETSS	$13+1.87*\log_2 w$	16.7	18.6	20.5
4SS	18 (log2(w/4)) + 9	81	289	1089
DS	$9+\mathrm{msx}\{5,4,3\}*\log_2 w$	19	24	29
CDS	$3 + 2w$	11	19	35
HEXBS	$7+3*\log_2 w+4$	17	20	23
SS	$1 + 8 \log_2 w - 1$	17	25	33
FSS	$(2 \times w) + 1$	**9**	**17**	**33**

CONCLUSION

A new Fibonacci Spiral Search algorithm is proposed in this paper. In this proposed algorithm we are considering two measures, speed up the searching process and minimize the number of searching block. . The comparison table based on execution time is presented at end. The proposed algorithm will be implemented in H.264 standard. We expect the proposed algorithm provides high compression ratio with better quality compared with full search.

REFERENCE

[1] J. R. Jain and A. K. Jain, “Displacement measurement and its application in inter frame image coding,” IEEE Trans. on Communications, Vol. COM-29, No. 12, pp. 1799-1808, Dec. 1981.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion compensated inter frame coding for video conferencing," pp. G5.3.1-5.3.5, Pro. Nat. Telecommun. Conf., New Orleans, Nov. 1981.

[3] Jong-Nam Kim and Tae-Sun Choi, “Fast Full-Search Motion-Estimation Algorithm Using Representative Pixels and Adaptive Matching Scan”, IEEE Transactions on Circuits and Systems for Video Technology, Vol.10, No.7, October 2000.

[4] Renxiang Li, Bing Zeng, and Ming L. Liou, “A New Three-Step Search Algorithm for Block Motion Estimation”, IEEE Trans. Circuits And Systems For Video Technology, Vol 4., No. 4, pp. 438-442, August 1994.

[5] M. Ghanbari, “The Cross-Search Algorithm for Motion Estimation”, Vol. 38, No. I, pp 950-953, IEEE Transactions on Communications, July 1990.

[6] Lai-Man Po Wing-Chung Ma, “A Novel Four-Step Search Algorithm for Fast Block Motion Estimation”, vol. 6, No. 3, pp. 313-317, IEEE Transactions on Circuits Syst. Video Technol., June 1996.

[7] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", Vol. 9, No. 2, pp 287-290, I EEE Transactions on Image Processing, FEBRUARY 2000.

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

[9] Ms. Maya Gaikwad, "Implementation of four step search algorithm of motion estimation using FPGA", Vol. 1, Issue 3, International Journal of Advanced Research in Computer Science and Electronics Engineering, May 2012

[10] B.G. Kim, S.T. Kim, S.K. Song and P.S. Mah, "Fast-adaptive rood pattern search for block motion estimation", Vol.41, No.6, ELECTRONICS LETTERS, August 2005.

[11] Barjatya, "Block Matching Algorithms for Motion Estimation," DIP 6620, Spring 2004.

[12] S. Metkar and S. Talbar, "Motion Estimation Techniques for Digital Video Coding", V0l 12, pp 64, Springer Briefs in Computational Intelligence, 2013.

[13] Puri, H. M. Hang and D. L. Schilling, "An efficient block matching algorithm for motion compensated coding," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc., pp. 1063-1066, 1987.

[14] Chun-Ho Cheung and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", Vol. 12, No. 12, pp 1168-1177, IEEE Transactions on Circuits and Systems for Video Technology, December 2002.

[15] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", Vol. 12, No. 5, pp 349-355, IEEE Transactions on Circuits and Systems for Video Technology, May 2002.

AUTHOR DETAILS

L.C.Manikandan is working as an Assistant Professor at Mar Ephraem College of Engineering and Technology, Marthandam, Tamilnadu, INDIA. He has received M.Sc., Degree in Computer Science, M.Tech., Degree in Computer and Information Technology and currently pursuing Ph.D at ManonmaniamSundaranar University. His main research interest includes video coding in image processing.

R.K.Selvakumar received his Ph.D degree in Computer Engineering from ManonmaniamSundaranar University, India. He has completed the Master degrees M.Sc.(Maths), DOEACC-C(CT), M.C.A., M.Phil.(CS) and M.Tech.(C&IT). He is currently working as Professor at Agni College of Technology, Chennai, Tamilnadu, INDIA. He carries out research in Digital Image Processing, Video Surveillance, Video coding, Digital Watermarking and Soft Computing.