RESEARCH ARTICLE OPEN ACCESS

An Efficient Method for Fractal Video Compression using Block Matching Motion Estimation

Rakhi Ashok Aswani*, Shailesh D.Kamble**
*(Department of Computer Technology, Yeshwantrao Chavan College of Engineering, Nagpur
Email: rakhi7.aswani@gmail.com)
** (Department of Computer Technology, Yeshwantrao Chavan College of Engineering, Nagpur
Email: shailesh_2kin@rediffmail.com)

ABSTRACT
The large usage of multimedia applications on internet and mobiles has increased the demand of compressed data, in order to reduce the requirement of bandwidth and time to transfer the data. So block based method for video compression is getting more importance due to its effectiveness and easy implementation. Fractal video compression follows the property of self-similarity. That's the biggest reason for high compression ratio and also attracts more researchers to work for fractal compression. In block matching motion estimation various algorithm has been proposed having different search pattern and strategies. As the search pattern gives large impact on efficiency of the algorithm. The paper presents an efficient method on fractal video compression, which follows half-way stop technique.
Keywords - bin-tree partitioning, fractal video compression, inter-pixel similarity, intra-pixel similarity , quad-tree partitioning, bin-tree partitioning

I. INTRODUCTION

Block matching motion estimation becomes the most attractive area for research. If the frame rate is high, then there will be great amount of similarity in successive frames. The temporal redundancy needs to be eliminated between successive frames. The frames are divided into rectangular blocks and block displacement need to be calculated from previous frame with the current frame. This process is computational intensive and takes 80% of the encoding time.

The concept of block matching is based on matching of a block with other blocks. So the image is partitioned into domain blocks and range blocks. Domain blocks are non-overlapping blocks, while range blocks are overlapping blocks. As an image can have some similar blocks also. It is not necessary that all the blocks in an image are completely dissimilar with each other. So main task is finding those blocks in entire video by using intra-pixel and inter-pixel property, to save the storing space as well as for compression purpose. Intra-pixel similarity refers to similarity in successive frames of a video while in inter-pixel similarity refers to similarity within a frame only.

An image can be partitioned into blocks by bin-tree partitioning or Quad-tree partitioning or triangular partitioning, Where bin-tree partitioning some times refers to horizontal vertical partitioning. While in Quad-tree partitioning the image is divided into four quadrants. the partitioning is continued till no dissimilarity remained in image.

We have proposed an efficient Four-Step Search, which gives better results than FS, 3SS, 4SS & DS. The rest of the paper is organized as follows. The second section explains about fractal video compression. The third section explains 4SS. The fourth section explains E4SS. Then experimental results are shown.

II. FRACTAL VIDEO COMPRESSION

The fractals uses the concept of self similarity. Fractal image compression reduces the redundancy of images by using self-similarity property and is a favorable method for compressing an image due to its high compression ratio, fast decompression and resolution independence[4]. The process of Fractal Video Compression is same as Fractal Image Compression. The major difference between them is that, video compression takes advantage of the similarity between successive frames and hence we obtain higher compression rates, if compared with image compression[5].

There are many block matching algorithm (BMA) has been proposed. BMA serves as the basic method in many international standards like MPEG-1, MPEG-2, H.261 and H.263. Full search is optimal process to find the global solution. It searches for all the candidate points in the search area. But it takes a lot of time for block matching motion estimation. There are many fast block matching algorithm have been proposed like Three-Step Search[6], Efficient Three-Step Search[7], New Three-Step Search[8], Improved Three-Step Search[9], Four-Step Search[10], Diamond Search[11], Hexagonal Search etc. Most of the algorithm follows the center biased search strategy. Which will directly gives large.

The Three-Step Search (3SS) is the simplest algorithm and also easy to implement in [6]. It uses 9x9 window, which is reduced by half in every step, maintaining the pattern as rectangular shape. New Three-Step Search (N3SS) uses the center biased characteristics in [8] and improve the performance of 3SS by adding central 8 points and allowing termination after first or second step. Improved Three-Step Search (I3SS) employs 5x5 window in [9] and also reduces no. of search points. In last step the window size is reduced to 3x3. Four-Step Search (4SS) gives better result than 3SS and similar to N3SS in [10]. It has four steps rather than three steps in 3SS and also utilizes center biased characteristics. Diamond search (DS) uses small and large diamond shaped pattern instead of rectangular shape in other algorithm in [11]. Also does the overlapping of checking points between the steps, in order to reduce the complexity. It has better performance than 3SS, 4SS and N3SS.

The size of block also gives impact on the speed. As large block size has less accuracy and small block size has more accuracy. So there is tradeoff between the accuracy and time.

III. FOUR-STEP SEARCH

The 4SS algorithm exploits the center biased property. Here 9 points are located on the center of 5x5 window of 15x15 search space in first step. In second step additional 5 or 3 points are considered according to the position of min. BDM point. In step three the process is repeated. If min. BDM point found at center then search will go to fourth step. Here is the algorithm for 4SS.

Step 1: Firstly 9 checking points are considered on the center of 5x5 window in 15x15 search area and min. BDM point is found among 9 points. If the minimum BDM point is found to be center of search window, go to last step; otherwise go to next step.

Step 2: Now, the search pattern will depend on the location of the previous minimum BDM point.

a) If the min. BDM point is at the corner then additional five checking points are used.
b) If the min. BDM point is at the middle of horizontal or vertical edge then additional three checking points are used.

If the min. BDM point is found at the center of the search window, go to Step 4; otherwise go to Step 3.

Step 3: The search pattern strategy will be same as Step 2, but if min. BDM point found to be center of the search window then it will go to Step 4.

Step 4: The search window is now reduced to 3 x 3 and 8 checking points is now considered surrounding min.BDM point. And again min. BDM point is found.

The figure1 shows the path for motion vector in 4SS.

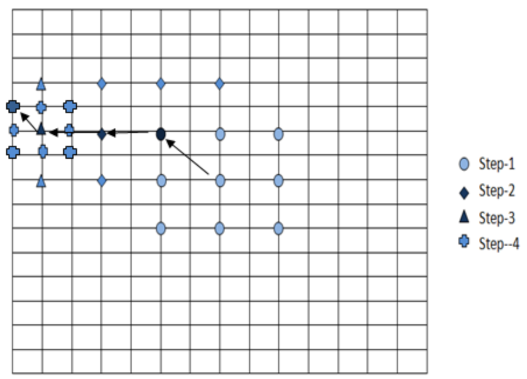

Fig.1 The path used in 4SS

IV. AN EFFICIENT FOUR-STEP SEARCH

In efficient four step search, center biased characteristics is utilized. In first step 9 points are considered on 5x5 window at the center of 15x15 search area. Then min. BDM(Block Distortion Measure) point is computed from center. In second step 5 or 3 additional points are added according to the location of min. BDM point. The third step will be same as second. According to the location of min. BDM point, the center of search window is shifted. In last step small cross shape pattern is designed around the BDM point. And if min. BDM point found at center, then search will stop.

If min. BDM point found at center in any step, then search will proceed to final step. this will give immense impact on search strategy. Fig.2 shows the flow chart of the algorithm.

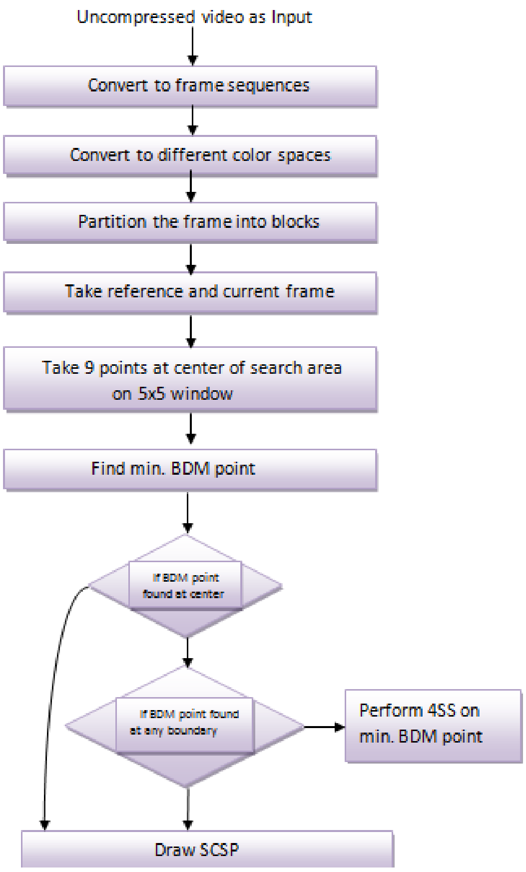

Fig.2 The flow chart for proposed algorithm

There is two pattern used in E4SS. One is rectangular 9 points and other is SCSP(small cross shaped pattern). Fig.3 shows the path used in E4SS.

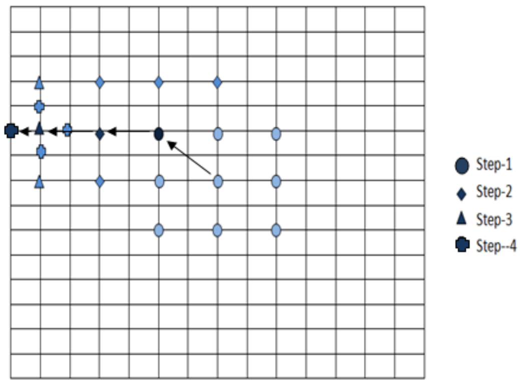

Fig.3 The path used in E4SS

V. EXPERIMENTAL RESULTS

The experimental results shows color conversion to different color spaces as color to YCbCr and then YCbCr to gray. Then image is decomposed using Quad-tree decomposition.

REFERENCES

[1] M.Barnsley and H.Rising, "Fractals Everywhere", Academic Press, San Diego,CA,USA, 1993
[2] A.E.Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations" *IEEE* 1992
[3] Mr.M.Manikandan, Mr.P.Vijayakumar, Mr.N. Ramadass, "Motion Estimation Method for Video Compression –An Overview", *IEEE* 2006
[4] Kamel Belloulataa, Shiping Zhub and Zaikuo Wangb, "A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation"
[5] Meiqing Wang, Rong Liu, Choi-Hong Lai, "Adaptive Parttion and Hybrid Method in Fractal Video Compression", *ELSEVIER 2006*.
[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motioncompensated interframe

coding for video conferencing," in *Proc. New Orleans, LA, Nov. 1981*
[7] Xuan Jing and Lap-Pui Chau, "An Efficient Three-Step Search Algorithm for Block Motion Estimation" *IEEE Trans. Vol. 6,* June 2004
[8] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technology vol. 4,* 1994
[9] Donglai Xu, Chris Bailey and Reza Sotudeh, "An Improved Three-Step Search Block-Matching Algorithm for Low Bit-Rate Video Coding Applications", *IEEE* 1998
[10] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technology vol. 6*,1996
[11] Ming-gang Liu, Chao-huan Hou," A Fast Block-Matching Motion Estimation Algorithm Based on Spatial-Temporal Motion Vector Correlation" *International Symposium on intelligent Multimedia, Video and Speech Processing* 2001
[12] Chun-Ho Cheung and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", *IEEE Transactions on Circuits and Systems for Video Technology,* December 2002.
[13] Queen Mary, "A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation", *EURASIP Journal on Applied Signal Processing* 2002
[14] Kamel Belloulataa, Shiping Zhub, Jun Tianb, Xiaodong Shenb, " A Novel Cross-Hexagon Search Algorithm for Fast Block Motion Estimation " *IEEE* 2011
[15] Chun-Ho Cheung and Lai-Man Po," Novel Cross-Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation*", IEEE Transactions on Multimedia, Vol. 7, No. 1,* February 2005
[16] Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, and Ashraf Ali Kassim, " A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation", *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 4,* August 1998
[17] Shiping Zhu, Yangshuan Hou, Zaikuo Wang, Kamel Belloulata, A Novel Fractal Video Coding Algorithm Using Fast Block-matching Motion Estimation Technology" IEEE 2010
[18] N.A.Koli and M.S.Ali, "A Survey on Fractal Image Compression Key Issues", *Information Technology Journal* 2008.
[19] Video Coding And Motion Estimation- Block Based Motion Estimation Algorithms, Version 2 ECE IIT, Kharagpur
[20] Dr. K. Kuppusamy, R.Ilackiya, "Fractal Image Compression & Algorithmic Techniques", 2013
[21] Meenu Bansal, Sukhjeet K. Ranade, "A review on fractal image compression", 2012
[22] Miroslav Galabov," Fractal Image Compression", *CompSysTech* - 2003
[23] Kamel Belloulataa, Shiping Zhub and Zaikuo Wangb, "A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation"
[24] Y. Fisher, "Fractal encoding with quadtrees," in Fractal Image Compression: Theory and Applications to Digital Images, *Springer-Verlag*, New York, USA, 1995
[25] T. Ochotta and D. Saupe," Edge-based partition coding for fractal image compression", 2004
[26] K. Belloulata, "Fast fractal coding of subbands using a non-iterative block clustering", 2005
[27] Information Technology, Coding of Audio Visual Objects - Part 2: Visual, ISO/IEC, (MPEG-4 Visual), 1999
[28] John Kominek," Advances in fractal compression for multimedia applications", Multimedia Systems, *Springer-Verlag* 1997