

A Novel Search Method for Fractal Video Compression using Block Matching Motion Estimation

Vinisha Assudani
Assistant Professor
Computer Engineering Department
St. Vincent Pallotti College of Engineering &
Technology, Nagpur, India

P

Purshottam J.Assudani
Assistant Professor
Computer Engineering Department
Shri Ramdeobaba College of Engineering &
Management, Nagpur, India

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.

General Terms

Fractal Video Compression using Block matching motion Estimation

Keywords

Bin-tree partitioning, fractal video compression, inter-pixel similarity, intra-pixel similarity, quad-tree partitioning, weighted finite automata

1. 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. This paper has 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.

2. 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 impact on the search speed.

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 trade-off between the accuracy and time.

3. WEIGHTED FINITE AUTOMATA

The Weighted Finite Automata has been successfully worked for image and video compression. A WFA composed of states of images and their expanded sub-images. These sub-images can be combined together and reconstruct the original images as a close approximation. When these WFA further encoded then they are regarded as compressed image[32]. The weighted finite automata exploits the self '+' concept, which is used in fractal compression. To achieve the compression, it represents an image as a weighted sum of sub-images. So that the part of images which are similar are stored only once, in order to achieve high compression ratio and to store the space in domain pool also. This is done in both inter-pixel and intra-pixel sub-images in a video. In this manner it constructs a graph, which represent the image, where top of graph shows the entire image. The different sub-images are compressed and final automata is very much compact. There are two forms of WFA that have been applied to image compression. The first was developed by Culik[29] while the second was developed by Hafner [30]. These two forms of WFA are then employed to represent any given images and will be quantized before being entropy coded into the compressed images. For convenience, Culik's WFA will be referred to as a linear WFA while Hafner's WFA will be termed a hierarchic WFA due to its hierarchic nature. Note

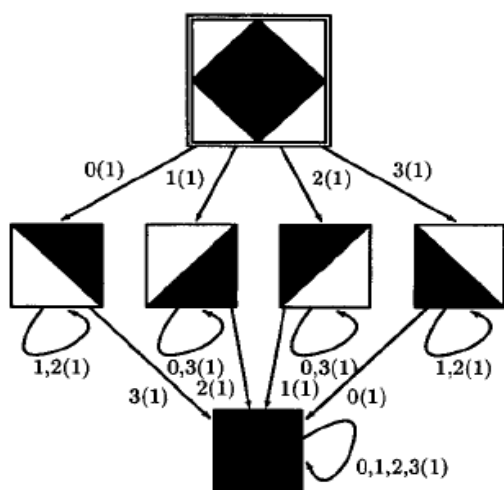


Fig 1. Linear WFA

that a hierarchic WFA can be converted to a linear WFA but not vice versa. Figure 1 shows linear WFA.

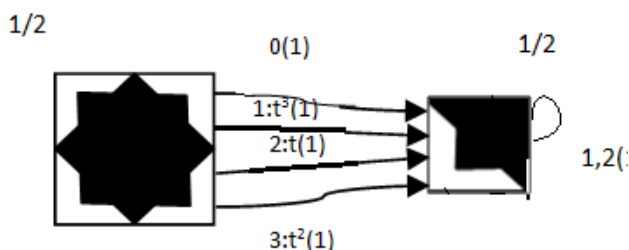


Fig.2 Extended WFA

The main issue is to partition an image into blocks. There are various approaches proposed like bin-tree partitioning, quad-tree partitioning etc. Quad-tree partitioning is very much

popular approach. And also it is shown that rectangular decomposed image has higher quality[35].

4. FRACTAL COMPRESSION ALGORITHMS

There are many algorithms of Fractal Compression, which are proposed and implemented. This paper uses 4SS to explore the algorithm and enhance the performance.

Four-Step Search Algorithm

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[27]. The fig.3 shows the path for motion vector in 4SS. Here is the algorithm for 4SS.

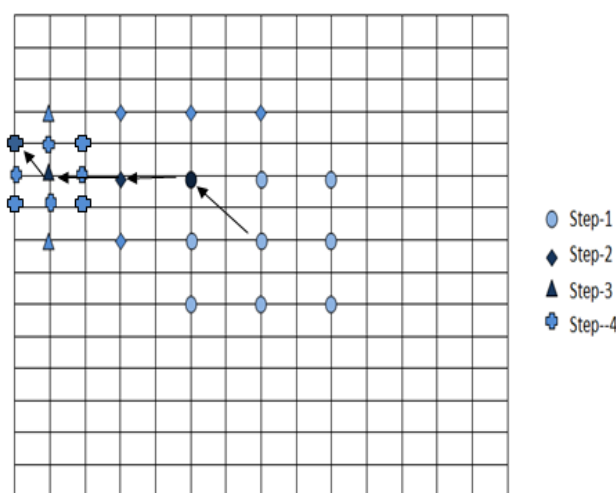


Fig.3 The path used in 4SS

- 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.

4.2 Efficient Four-Step Search - Proposed Algorithm

In efficient four step search, center biased characteristics is utilized. In first step 9 points are considered on 13x13 window at the center of 15x15 search area. Then min. BDM(Block Distortion Measure) point is computed from these 9 points. In second step 5 or 3 additional points are

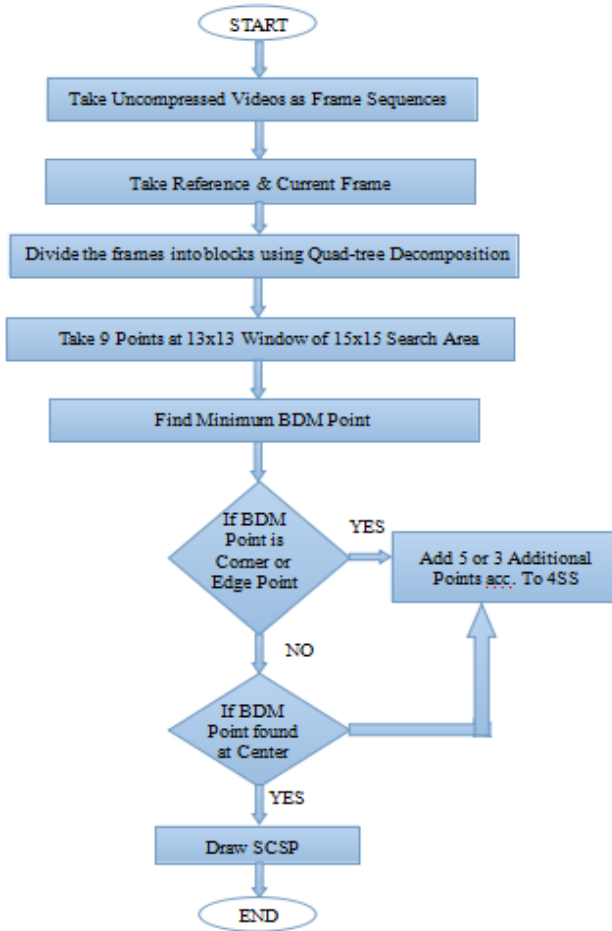


Fig.4 The flow chart for proposed algorithm

added according to the location of min. BDM point. If BDM is vertex point then 5 points are added, while if BDM is edge point then 3 points are added. The third step will be same as second. According to the location of min. BDM point, the center of search window is shifted according to position of BDM. 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.

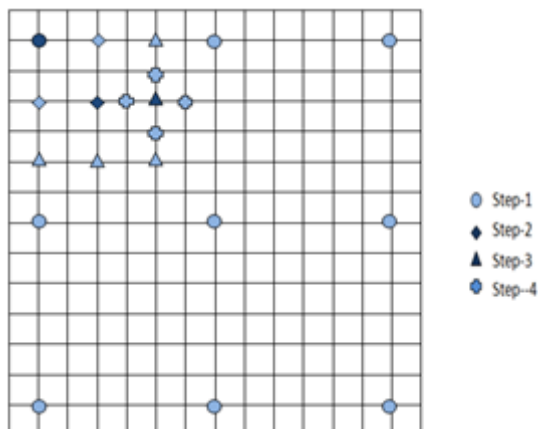


Fig.5 The path used in E4SS

There are two pattern used in E4SS. One is rectangular 9 points and other is SCSP(small cross shaped pattern). Fig.5 shows the path used in E4SS.

5. EXPERIMENTAL RESULTS

The experimentation is done on different standard databases like akiyo, traffic, football and soccer. Some user-defined databases like nature and rhinos. The 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. And lastly image reconstruction is shown.

1) Akiyo Video

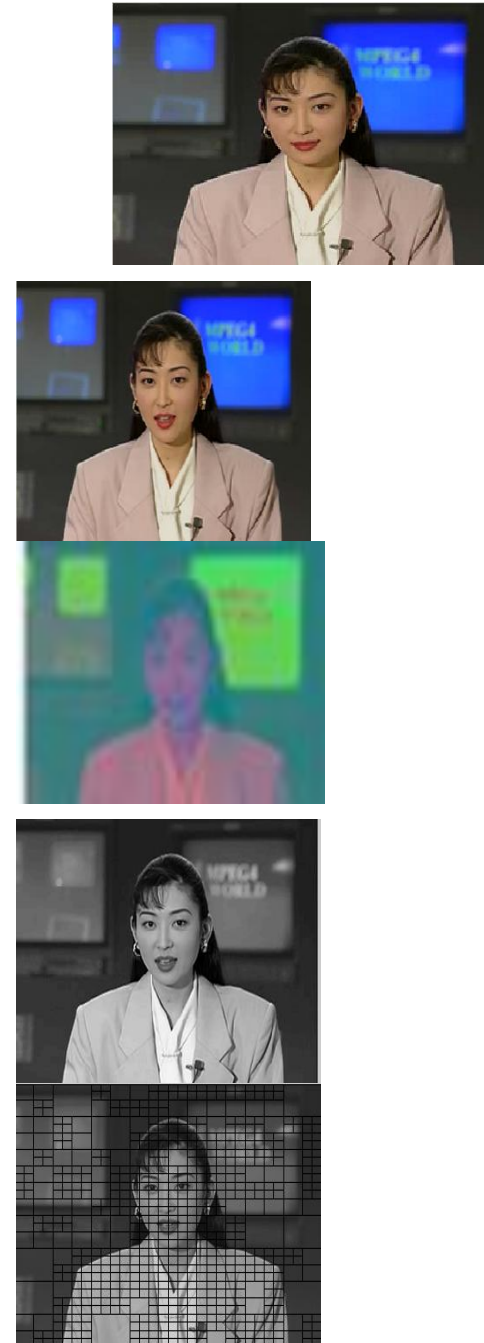


Fig.6: Process of Encoding - Akiyo video(Colored Reference frame, current frame , YCbCr, gray frame and quad-tree decomposition)

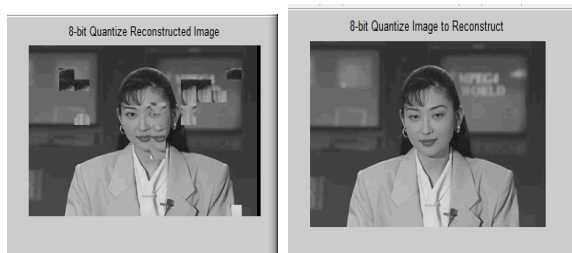


Fig.7: Image Reconstruction

2) Soccer Video



Fig.8 Process of Encoding - Soccer video (Colored Reference frame, current frame , YCbCr frame, gray frame and quad-tree decomposition)



Fig.9 Image Reconstruction

6. COMPARISION FOR DIFFERENT PARAMETERS

There is a comparison among MSE per pixel and no. of frames, which is shown by a graph. The table is also shown for first 20 frames. The graph for MSE per pixel and frame no. is shown below in fig.12, which has very

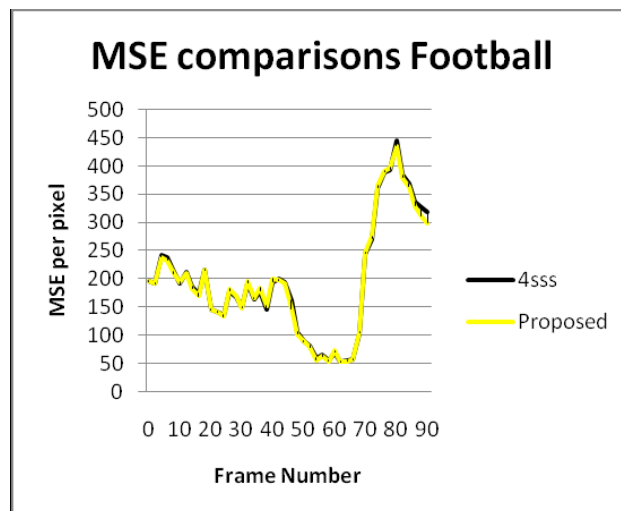


Fig.12: A comparison in 4SS and E4SS in football video

less difference. But as no. of frames increases, the MSE per pixel also get effected and reduced for our algorithm.

7. Conclusion

There are many BMA algorithm used to detect the MV and partitioning technique. Combining two or more of these methods will produce a good result. In this work, an efficient method for block matching motion estimation is presented. Furthermore, the work present a system include a series of steps. First step is color conversion, which involves color to YCbCr., then YCbCr to Gray. Secondly, image partitioning using Quad-tree is presented. Third step is Sorting n-frames. Fourth is BMA - Efficient Four Step Search. Finally, Encoding algorithm is presented and decoding is the last process. The complexity of BMA is the main issue. The paper presented such a method which has better performance and less complexity as compared to 3SS, 4SS and N3SS. In future, an algorithm can be proposed, which can reduce the time to encode an image and then its decoding time. So that better efficiency and results can be seen. Moreover, the database of image can be made secure.

8. 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. Inuma, 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, 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. Kuppasamy, 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 quadrees," in *Fractal Image Compression: Theory and Applications to Digital Images*, Springer-Verlag, New York, USA, 1995
- [25] Rakhi Ashok Aswani, Shailesh D.Kamble, "Fractal Video Compression using Block Matching Motion Estimation - A Study", Vol.4, 2014
- [26] K. Belloulata, "Fast fractal coding of subbands using a non-iterative block clustering", 2005
- [27] Rakhi Ashok Aswani, Shailesh D.Kamble, "An Improved Method for Fractal Video Compression using Block Matching Motion Estimation", Vol.2, 2014
- [28] Rakhi Ashok Aswani, Shailesh D.Kamble, "An Efficient Method for Fractal Video Compression using Block Matching Motion Estimation", *International Journal of Engineering Research and Applications*, Vol.5, 2014
- [29] Culik K and Kari J, "Inference algorithm for WFA and image compression", *Fractal Image Compression: Theory and Application*, edr. Fisher Y, Springer, New York 1995
- [30] Hafner U, "Refining image compression with weighted finite automata", *Proceedings of Data Compression Conference*, eds. Storer J and Cohn M, (1996)359-368
- [31] Jiirgen Albert, Stefan Frank, Ullrich Hafner*, Michael Unger, "Video Compression with Weighted Finite Automata", 1997 *IEEE*
- [32] Zhuhan Jian, Bruce Litow, Olivier de Vel, "An Inference Implementation Based on Extended Weighted Finite Automata" 2001 *IEEE*
- [33] Ullrich Hafner, Stefan Frank, Michael Unger, Jurgen Albert, "Hybrid Weighted Finite Automata for Image and Video Compression", March 1997
- [34] Karel Culik, Peter C. von Rosenberg, "Generalized Weighted Finite Automata Based Image Compression", vol.5, no.4, Springer Pub. Co. 1999
- [35] Ghim Hwee Ong And Kai Yang, "A Binary Partitioning Approach to Image Compression using Weighted Finite Automata for Large Images" 2006 Elsevier
- [36] Marian Mindek, "Finite State Automata and Image Recognition", *Technical University of Ostrava*, 2004
- [37] Helmut Jürgensena,b, Ludwig Staigerc, Hideki Yamasakid, "Finite automata encoding geometric figures", 2007 Elsevier
- [38] Ullrich Hafner, "Image and Video Coding with Weighted Finite Automata"
- [39] F. Katritzkea, W. Merzenich, M. Thomas, "Enhancements of partitioning techniques for image compression using weighted finite automata", 2003 Elsevier
- [40] Zhuhan Jianga, Olivier de Velb, Bruce Litowc, "Unification and extension of weighted finite automata applicable to image compression", 2002 Elsevier
- [41] John Kominek, "Advances in fractal compression for multimedia applications", *Multimedia Systems*, Springer-Verlag 1997