International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 1, 2013. ISSN: 2348-6600
Reference ID: IJCS-009 PAGE NO: 044-049

Novel Color Fractal Image Compression based on Quadtree Decomposition

Dr. K. Kuppusamy [#1], R.Ilackiya [*2]

[#1] Associate Professor ,Department of Computer science and Engineering,
Alagappa University, Karaikudi.
India.

[*2] Department of Computer science and Engineering,
Alagappa University, Karaikudi.
India.

Abstract— **Fractal image compression is one of the most promising techniques for image compression due to advantages such as resolution independence and fast decompression. It exploits the fact that natural scenes present self-similarity to remove redundancy and obtain high compression rates with smaller quality degradation compared to traditional compression methods.**

Keywords—Fractal Image Compression, Quadtree, self - similarity

I. INTRODUCTION

Fractal image compression has become an important lossy technique due to its capacity of achieving high compression ratio and high reconstructed image quality. The basic idea of fractal image compression is to exploit self-similarities present in the image to reduce the amount of redundancy. On the other hand, the main disadvantage of fractal encoding is its high computational cost due to the search used to find self-similarities.

A contractive transformation is used to map each range block to its matched domain block. The
transformation aims at minimizing a metric distortion measure during the search process. Similarity transforms parameters between the range block and the corresponding domain block are used to encode the range block. The encoded image can be reconstructed by iterative evaluation of the transformations until converging to an approximation of the original image. Several efforts have focused on reducing the complexity of the matching process during the fractal image encoding.

Fractal image compression is also called as fractal image encoding because compressed image is represented by contractive transforms and mathematical functions required for reconstruction of original image. Contractive transform ensures that, the distance between any two points on transformed image will be less then the distance of same points on the original image . These transforms are composed of the union of a number of affine mappings on the entire image, known as iterated function system (IFS). Fractal image compression is based on contractive affine transformation is of the form: is sufficient to generate interesting transformations called affine transformations of the plane. Each can skew, stretch, rotate, scale and translate an input image.

***Contractive Transformations*:**

A transformation w is said to be contractive if for any two points P1, P2, the distance
d(w(P1),w(P2)) < sd(P1,P2)for some s < 1, where d = distance. This formula says the application of a contractive map always brings points closer together.

***The Contractive Mapping Fixed Point Theorem*:**

This theorem says something that is intuitively obvious: if a transformation is contractive then applied repeatedly starting with any initial point, we converge to a unique fixed point. If X is a complete metric space and$W:X\rightarrow X$ is contractive, then W has a unique fixed point .

Compression Methods

The task of compressing an image includes three important parts:

1) Partition the image and find transformations for each partitioned part;

2) Encoding (compressing) the image; and

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 1, 2013. ISSN: 2348-6600
Reference ID: IJCS-009 PAGE NO: 044-049

3) Decoding (decompressing) the image.

Partitioning Images

For the issue of fractal image compression, we choose the quad tree method to partition images because of its orderliness, simplicity, and efficiency.

Quad tree Partition

Quad tree is an image structure, which appeared at the end of the 1970s, and was developed and applied in the 1980s and the 1990s. The root of the tree is the initial image. First, the image is divided into domains with different sizes using the quadtree method and the domain pool is built, including sizes and positions of the quadtree partitions are prepared. The squares at the nodes are compared with domains in the domain pool (or domain library) *D*, which are twice the range size. Nodes are compared with domains in the domain pool (or domain library) *D*, which are twice the range size. The pixels in the domain are averaged in groups of four so that the domain is reduced to the size of the range, and the affine transformation of the pixel values is found that minimizes the rms difference between the transformed domain pixel values and the range pixel values. All the potential domains are compared with a range. If the resulting optimal rms value is above a pre-selected threshold and if the depth of the quadtree is less than a preselected maximum depth, then the range square is subdivided into four quadrants, and the process is repeated. If the rms value is below the threshold, the optimal domain and the affine transformation on the pixel values are stored. Thus one map w_i is found. The collection of all such maps $W = \cup \; w_i$ constitutes the encoding.

II. QUADTREE

A quad tree is a tree data structure in which each internal node has up to four children. Quad trees are most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions. The regions may be square or rectangular, or may have arbitrary shapes. A quad tree is a representation format used to encode images. The fundamental idea behind the quad tree is that any image can be split into four quadrants. Each quadrant may again be split in four sub quadrants as shown in figure 2. In the quad tree, the image is represented by a parent node, while the four quadrants are represented by four child nodes, in a predetermined order. Of course, if the whole image is a single color, it can be represented by a quad tree consisting of a single node. In general, a quadrant needs only to be subdivided if it consists of pixels of different colors. As a result, the quad tree need not be of uniform depth.Unlike a normal tree data structure, a quadtree structure requires that each internal node have exactly four children nodes. When illustrating most quad tree structures, you'll see a node that has four children nodes hanging from it, with lines connecting the parent node with its children nodes. The illustration can continue, with four more children nodes hanging from each of the original four children nodes. Other times, the illustration of a quad tree will be a region or square. Whenever the region reaches its maximum capacity for storing data, it is divided into four quadrants as shown in figure 1. Normally, the regions and the quadrants are squares, although they can be rectangles or other shapes, also.

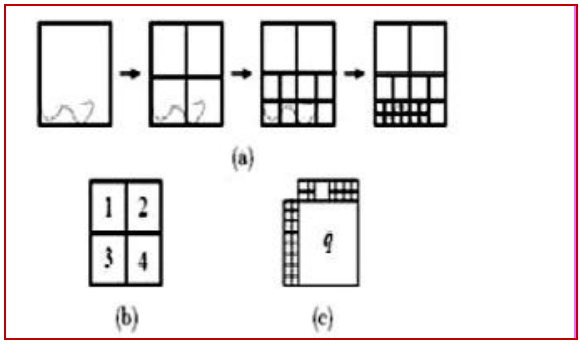


Figure:1

TYPES

Quadtrees may be classified according to the type of data they represent, including areas, points, lines and curves. Quadtrees may also be classified by whether the shape of the tree is independent of the order data is processed. Some common types of quadtrees are:

The region quadtree

The region quadtree represents a partition of space in two dimensions by decomposing the region into four equal quadrants, subquadrants, and so on with each leaf node containing data corresponding to a specific subregion. Each node in the tree either has exactly four children, or has no children (a leaf node). The region quadtree is a type of trie.

A region quadtree with a depth of n may be used to represent an image consisting of $2^n \times 2^n$ pixels, where each pixel value is 0 or 1. The root node represents the entire image region. If the pixels in any region are not entirely 0s or 1s, it is

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 1, 2013. ISSN: 2348-6600
Reference ID: IJCS-009 PAGE NO: 044-049

subdivided. In this application, each leaf node represents a block of pixels that are all 0s or all 1s.

A region quadtree may also be used as a variable resolution representation of a data field. For example, the temperatures in an area may be stored as a quadtree, with each leaf node storing the average temperature over the subregion it represents.

If a region quadtree is used to represent a set of point data (such as the latitude and longitude of a set of cities), regions are subdivided until each leaf contains at most a single point.

Point quadtree

The point quadtree is an adaptation of a binary tree used to represent two dimensional point data. It shares the features of all quadtrees but is a true tree as the center of a subdivision is always on a point. The tree shape depends on the order data is processed. It is often very efficient in comparing two dimensional ordered data points, usually operating in O(log n) time.

Node structure for a point quadtre

A node of a point quadtree is similar to a node of a binary tree, with the major difference being that it has four pointers (one for each quadrant) instead of two ("left" and "right") as in an ordinary binary tree. Also a key is usually decomposed into two parts, referring to x and y coordinates. Therefore a node contains the following information:

- four pointers: quad['NW'], quad['NE'], quad['SW'], and quad['SE']
- point; which in turn contains:
- key; usually expressed as x, y coordinates

Edge quadtree

Edge quadtrees are specifically used to store lines rather than points. Curves are approximated by subdividing cells to a very fine resolution. This can result in extremely unbalanced trees which may defeat the purpose of indexing.

Polygonal map quadtree

The polygonal map quadtree (or PMQuadtree) is a variation of quadtree which is used to store collections of polygons that may be degenerate (meaning that they have isolated vertices or edges). There are three main classes of PMQuadtrees which vary depending on what information they store within each black node. PM3 quadtrees can store any amount of non intersecting edges and at most one point. PM2 quadtrees are the same as PM3 quadtrees except that all edges must share the same end point. Finally PM1 quadtrees are similar to PM2 but in this case black nodes can either contain a point and its edges or just a set of edges that share a point but you cannot have a point and a set of edges which do not contain that point.

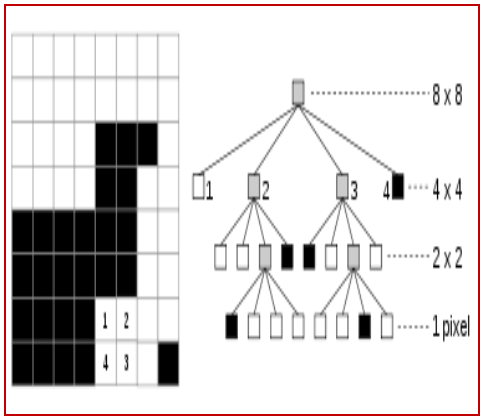


Figure 2

III. QUADTREE DECOMPOSITION

The quad-trees are powerful and simple data structures for representing and/or compressing digital images. It is possible to find applications of quad-tree decomposition in many different contexts, such as the compression of sub-band coefficients in wavelet decomposition and coding of the sub-blocks data in EBCOT algorith. The quad-tree decomposition is an important tool for fractal image compression where many suitable smart variants of quad-trees are used and applied.

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 1, 2013. ISSN: 2348-6600
Reference ID: IJCS-009 PAGE NO: 044-049

After the color palette is generated, the input image is uniformly divided into blocks of size N by N, where N is a power of 2. The choice of block size depends on the size of the input image. Currently, we set N=32 for images with size256x256 or larger. When input images are small, smaller N can be used. The smallest N used is 8. Blocks are processed in a raster scan order. For each block, the number of colors within this block will be examined. There are 3 possible cases

Case1: If a block contains only one color, the one byte block information will be output followed by one byte code to specify the corresponding color of that block as shown in figure 4.

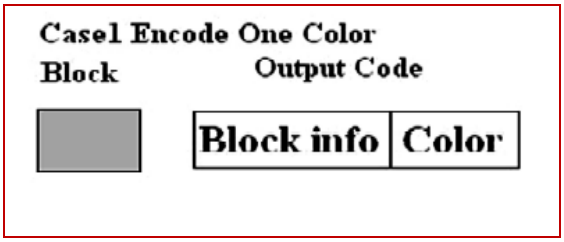


Fig 3 Case1 Illustration

Case 2: If a block contains exactly two colors, 4 bytes will be output. The first byte represents the block information. The next two bytes represent the two colors within that block. The last byte is used to encode the binary bit pattern (0's and 1's) to represent the two colors inside that block, which is the shape information. Here, we do not encode the binary pattern directly but use a look-up table called the "shape palette" to record the pattern of 0's and 1's. The value of the last byte is an index indicating the position in the shape palette.

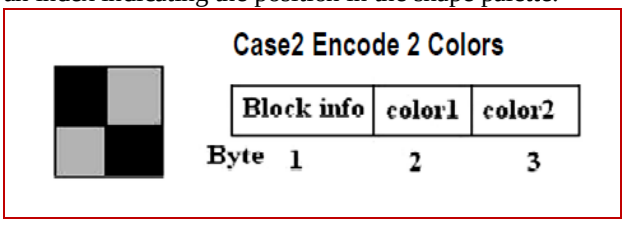


Fig 4 Case2 illustration

Case3: If there are more than 2 colors in a block, the block will be further divided into 4 sub blocks of equal size. The sub-division continues until one of the above two cases is met and the corresponding codes are output. We stop block splitting when the block is of size 2 x 2, even it has more than 2 colors. In this case, a code that indicates colors of each pixel in the block is output. That is, one byte of block information will be output to a stream followed by a 4-byte code to indicate the color of each pixel.

Block- Information Byte

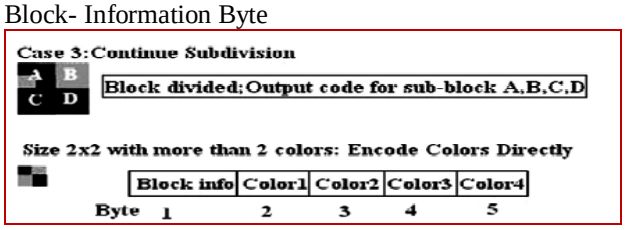


Fig 5 case3 Illustration

Bits 1 to 7 represent levels of sub-division and bit 0 represent number of colors.

• The last bit i.e., bit 0 is set to 0 if the block has one color or else it is set to 1.

• The block information byte is organized in the following way. Bit 7 to Bit 1 indicates levels of subdivision. Bit 1 is set to "1", when the block is subdivided once. The "1" is shifted to the left if there are more levels of subdivision, i.e. when Bits 7-1 are:

• "0000000" – undivided
• "0000001" – one level of subdivision
• "0000010" – two levels of subdivision
• ……
• "1000000" – seven levels of subdivision

If the block has one color (i.e., Case 1), bit 0 is set to "0".If the block contains exactly two colors (i.e., Case 2), bit 0 is set to "1". In the case of a 2 x 2 block with more than 2 colors, colors in this block will be encoded directly. Bits 7-1 will indicate if the subdivision already reaches the single-pixel level. If it does, bit 0 becomes a "Don't Care" bit

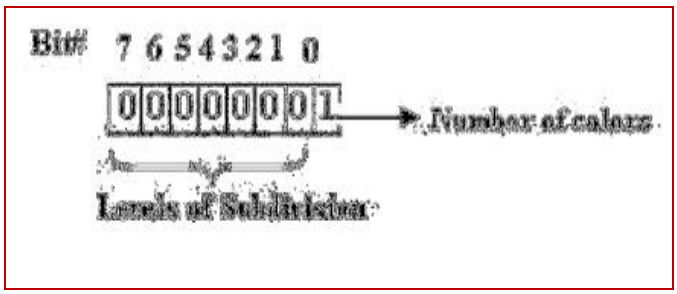


Fig 6 BLOCKS INFORMATION BYTE

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 1, 2013. ISSN: 2348-6600
Reference ID: IJCS-009 PAGE NO: 044-049

Color - Information Byte: This is 8 bit information of the corresponding color of the entire block obtained from color palette index. Each color is coded using one byte. The value of that byte is the index of the color in the palette. At any stage of subdivision, if a block or a sub-block contains exactly two colors, the binary bit pattern showing the position of each color needs to be coded as well. If we set the color at top left corner as "1" and the other color in the block as "0",the binary bit pattern is the pattern of "1" and "0" in the two color block scanned in a raster-scan order. To illustrate the 3 cases let us consider the following example shown in figure 8

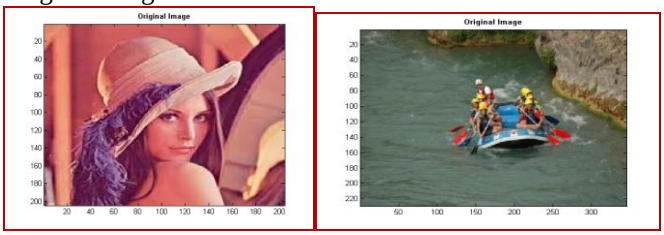
Let <N> denote output code for block N
<A> = "0000 0000 00000010 0100"
<B> = "0000 0000 00000100 0010"
<C> = "00001001 0001 0010 0001"
<D> = "0000 0000 00001000 0010"
<F> = "0001000x 0001 0100 0010"
<G> = "00000101 0010 0000 0001"
<I> = "0000 0000 00000010 0011"

Figure 7

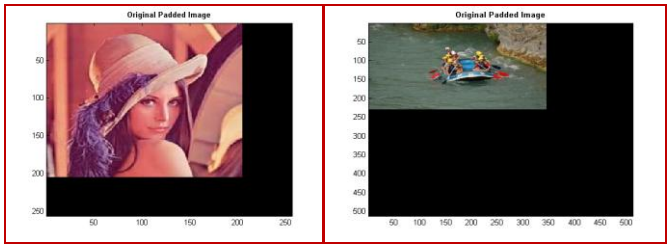
IV. EXPERIMENTAL RESULTS AND DISCUSSION

In order to test the performance of the proposed systems, they are applied using the same settings on four famous color images. These images are called Lena, Sailboat, and Baby,colour; see Fig. (8).The dimension and size of image are Different values respectively. The type of image are also different.The first two experiments are carried out using ;the threshold values are 0.5. The next two experiments are carried out using the threshold values are 0.2.The obtained results are shown in Figures 8,Figure 9

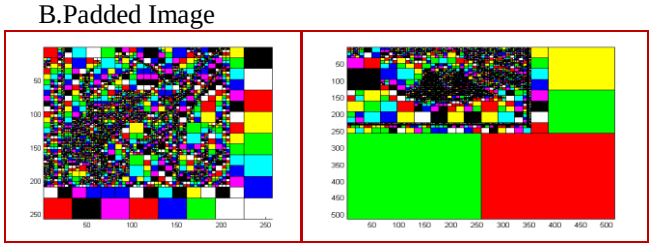
Original Image of threshold value of 0.2:



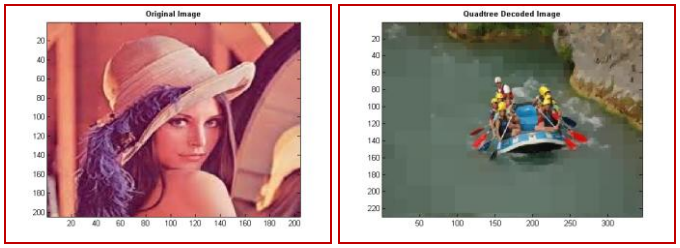
A.Original Image



B.Padded Image



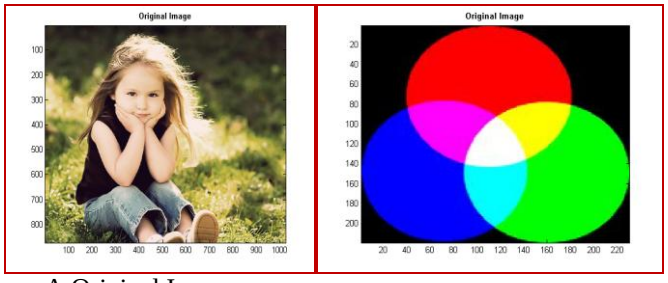
C.Encoding Image



D.Reoconstructed Image

Figure 8

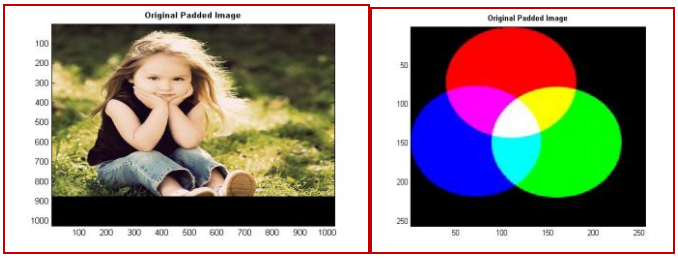
Original Image of threshold value of 0.5:



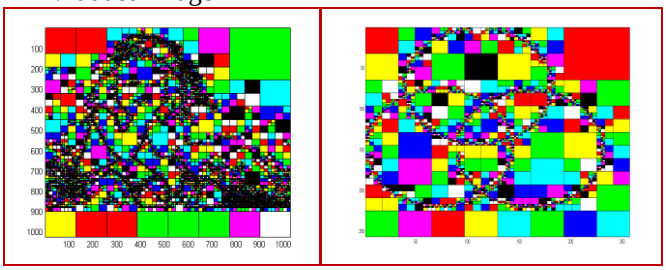
A.Original Image

Published by SK Research Group of Companies (SKRGC).

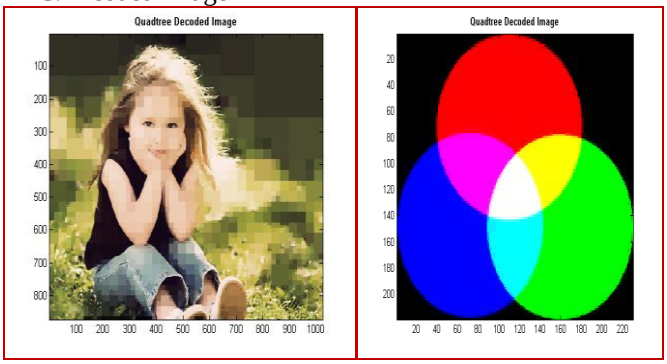
International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 1, 2013. ISSN: 2348-6600
Reference ID: IJCS-009 PAGE NO: 044-049



B.Padded Image



C.Encoded Image



D.Reconstructed Image

Conclusion

This paper has surveyed most important advances in different steps of fractal image compression. In this paper we review and discuss about the quad tree decomposition. This paper presents that have the ability to compress colour images in easy way. The division processes of image into blocks for the two systems are based on quad tree. At the dividing of one component, the other two components are divided using the same division even if the condition of quad tree division is not verified for them.This paper presents that have the ability to compress colour images in easy way. The division processes of image into blocks for the two systems are based on quad tree. At the dividing of one component, the other two components are divided using the same division even if the condition of quad tree division is not verified for them. After the division process is completed, the three components will have same number and size of the blocks.

References

[1] El-Harby α & G. M. Behery," Novel Color Image Compression Algorithm Based on Quad tree", Global Journal of Computer Science and Technology Graphics & Vision Volume 12 Issue 13 Version 1.0 Year 2012.

[2] Sankaragomathi, L. Ganesan, and S. Arumugam, "Fractal Image Compression Applied to Remote Sensing", World Academy of Science, Engineering and Technology 11 2007.

[3] Roshni S. Khedgaonkar, Shailesh D. Kamble, "Application of Quadtree Partitioning in Fractal Image Compression using Error Based Approach, IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org ISSN : 2250-3021 Vol. 2 Issue 1, Jan.2012, pp. 050-054.

[4] Ruhiat Sultana, Nisar Ahmed, Shaik Mahaboob Basha," Advanced Fractal Image Coding Based on the Quadtree", Computer Engineering and Intelligent SystemsISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)Vol 2, No.3,

[5] Tong Chong Sze," Fast Fractal Image Encoding Based on Adaptive",Search IEEE Transactions on Image Processing, 2001, 10(9):933-942.

[6] Saupe D, RuhlM, "Evolutionary Fractal Image Compression[OB/OL],http://www.inf.uni.konstanz.de/cgip/bib/files/SaRu96.pdf, 2004.

[7] Distasi R, Nappi M, Riccio D,"A range/domain approximation error-based approach for fractal image Compression",IEEE Transactions on Image Processing,2006, 15(1):89-97.

[8] Erjun Zhao, Dan Liu. Fractal image compression methods a review,Proceedings of the Third Sonal, D.Kumar,"A Study of Various Image Compression Technique".

[9] M.F. Barnsley and A.E. Slaon, "A better way to compress image, " *BYTE Magazine.*, January 1988.

[10] C. S. Tong and W. Man, "Adaptive Approximation Nearest Neighbor Search for Fractal Image Compression,"*IEEE Transactions on Image Processing,"* vol. 11, No. 6, pp. 605-615, 2002.

[11] Wohlberg and Gerhard de Jager,"A review of the Fractal Image Compression Literature," *IEEE Transactions on Image Processing*, vol. 8, No. 12, pp. 1716-1729, Dec. 1999.

[12] Yung-Kuan Chan (2004) "Block image retrieval based on a compressed linear quadtree", Image and Vision Computing, Vol.22(5), pp. 391-397.

[13] N. Udaya Kumar1 and K. Padma Vasavi (2011) " AN EFFICIENT COLOR CODING SCHEME FOR COLOR IMAGES USING QUAD TREE DECOMPOSITION AND COLOR PALETTES", World Journal of Science and Technology, Vol. 1(8) pp.26-31.

[14] Yuancheng Li, Qiu Yang, Runhai Jiao, "Image compression scheme based on curvelet transform and support vector machine", Expert Systems with Applications, Vol. 37(4), 2010, pp. 3063-3069.

[15] Bibhas Chandra Dhara, Bhabatosh Chanda, "Color image compression based on block truncation coding using pattern fitting principle", PatternRecognition, Vol. 40(9), 2007, pp. 2408-2417.