International Journal of Pure and Applied Mathematics
Volume 117 No. 15 2017, 513-526
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue
ijpam.eu

Analysis of Using Sparse Matrix Storage Formats in Image Compression Techniques

[1]J. Saira Banu, [2]M. Rajasekhara Babu, [3]I.Sumaiya Thaseen and [4]P.Balamurugan

[1]School of Computer Science and Engineering, VIT University, Vellore, India

jsairabanu@vit.ac.in

[2]School of Computer Science and Engineering, VIT University, Vellore, India

[3]School of Computer Science and Engineering, VIT University, Vellore, India

[4]School of Computer Science and Engineering, Veltech Dr.RR & Dr.SR University, Chennai, India

Abstract

Critical applications such as medical and satellite images which store similar images, use lossless compression techniques. Lossy compression techniques are applied to regular images, where high compression is traded for the loss of few pixels. This paper gives essential insights into the use of sparse matrix storage formats in regular and similar image compression techniques. Here, lossless-multilevel centroid method and lossy-rectangular segmentation method, in which the digital image is converted into sparse matrix, are considered. This work presents the analysis and effectiveness of various sparse matrix storage formats in image compression techniques. Furthermore, we experimentally demonstrate that the usage of various sparse matrix storage formats is an advantage in rectangular segmentation method than in multi-level centroid method.

Key Words: Image compression, sparse matrix, rectangular segmentation, multilevel centroid, CSR, COO, QCSR.

• Introduction

Images are essentially 2-D signals in analog form. For easier representation and processing, the image is stored in the computer in the digital form. The matrix representation of images improves the spatial relationship between the images when compared to vector representation [7]. However, the matrices can be of large sizes and can be very space inefficient. Image compression involves reducing the redundancy present in an image, to reduce the size of the matrix stored. Redundancy in an image can be of three different forms- coding, Inter-pixel and psychovisual [10]. Image compression can be broadly classified into two methods- lossless compression and lossy compression. Traditional image compression techniques concentrate on how to reduce the storage space by removing the redundancy of a single image. Lossless compression techniques such as Huffman coding and arithmetic coding provide a reconstructed image which is exactly the same as the original image. In this method, compressing the image does not affect the image resolution. Medical images and satellite images are generally a set of similar images. 'Set Redundancy', a term coined by [12] refers to the inter-image redundancy prevalent in a set of similar images. Lossless image compression techniques such as Min-Max differential method, Min-Max predictive method and multi-level centroid method are used to store these images as each pixel in the image can contain vital information necessary for analysis. In Lossy compression methods such as jpeg, transform coding, fractal compression and rectangular segmentation technique, the pixel values are sacrificed for the sake of storage reduction. In this paper, we first analyse the impact of using various sparse matrix storage formats in rectangular segmentation image compression technique [9][16] for single images. Rectangular segmentation has been proven to be better than Quarter-tree segmentation technique as the block sizes need not be in the powers of two. We then analyse the impact of using sparse matrix storage formats in Multi-level Centroid technique [13], for a set of similar images as the difference images obtained in the end are sparse matrices. The organization of the paper is given as: section 2 gives the literature survey, section 3 gives an overall description about the image compression techniques used, section 4 describes the various sparse matrix storage formats, section 5 discusses the rectangular segmentation method and multi-level centroid technique in section 6, section 7 gives the results obtained, and section 8 elucidates the conclusions.

• Literature Review

Sparse matrices are used in a wide range of applications, and active research has been going on in this field. Various storage formats for storing these matrices efficiently have been developed over the years. In their paper, Nathan Bell and Michael Garland [1] analyse the use of various famous storage formats, by implementing them on CUDA platform, with emphasis on the memory bandwidth efficiency. Xiangzheng Sun et al [2], have proposed the Compressed Row Segment with Diagonal-Pattern (CSRD) and have implemented it on GPU using OpenCL. This is an improved storage format for diagonal sparse matrices. D. Guo and W. Gropp, have proposed another format, the streamed storage format, which increases the memory bandwidth using the prefetch data stream and gives an improved SpMV performance when compared to that of the CSR and the Blocked-CSR formats. The matrix is divided into blocks called nstreams, such that each block contains n rows. These nstreams determine the format of the matrix [3]. The current authors, in a recent paper [6], used benchmark matrices to analyse the spatial efficiency of various formats and found that the Minimal Quadtree format is the best. However, since this is just a string of bits, reconstruction of the matrix is not possible. Hence the Compressed Row Storage (CSR) format, which was found to be the next best format, was considered to be the most efficient general storage format. This paper analyse the use of current sparse matrix storage formats such as COO, CSR, QCSR and Minimal Quad tree format in image compression techniques.

An image can be represented in the computer memory as a 2-dimensional matrix. Redundancy of data present in images can be removed using image compression techniques to store the images efficiently. If these images were represented in the form of sparse matrices, they can be represented using the various sparse matrix storage formats, hence increasing the compression ratio. Zhang Tianxu and Zeng Yonghui [7], have been working on the same. Tanaya Guha and Rabab K. Ward [14] have worked on using sparse matrix storage formats to compress similar images. Yi-Chen Tsai et al [10] have improved the performance of cartoon images using a quad tree decomposition technique. Kitty Arora and Manshi Shukla [15], have presented a survey on various lossless and lossy image compression techniques. Analysis on quarter-tree decomposition of image compression was performed by He Xing-heng and Chen Hui [8]. Though it is an easy and fast method of image compression, the drawback of this technique is that the size of the blocks to be segmented has to be in powers of 2. R.K Shengli Chen, et al [9] gave an improved compression algorithm based on the quarter tree segmentation method, called the rectangular segmentation method, using the COO sparse matrix storage format. In this paper, we implement this algorithm and analyse the effects of using various sparse matrix storage formats in place of the COO format used. Kosmas Karadimitriou [12] in his thesis presented various Set Redundancy Compression (SRC) techniques that can be used to compress a set of similar images. Yasser El-Sonbaty et al. [13] in their paper have presented an improved SRC technique called the Multi-level centroid technique, which leads to the representation of images as sparse matrices. In this paper, we analyse the effects of using various sparse matrix storage formats to store these images.

• Image Compression

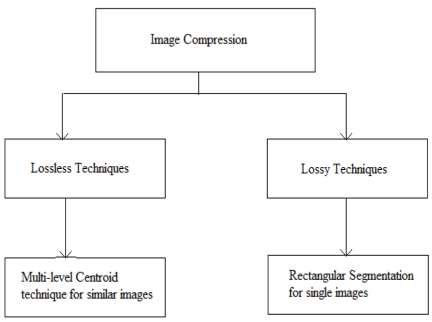


Figure 1: Image Compression Algorithms

Image compression is a technique that is used to remove redundant data from the image to store it efficiently. This can be generally classified into two categories- the lossy compression techniques and the lossless compression techniques. The lossy compression techniques include techniques that results in the loss of some data, to obtain a better degree of compression. There is a certain distortion created during reconstruction, and compression can be adjusted based on the distortion allowed by the application. The lossless techniques on the other hand, give back the exact data compressed on reconstruction. Though the compression ratio is not as high as the lossy techniques, these techniques are used in applications where the distortion has to be zero.

Figure 1 represents the overall block diagram of the branches of image compression techniques. Another type of compression technique is the SRC (Set Redundancy Compression) technique that compresses a set of similar images [11]. Three basic methods of achieving this are:

- Min-Max differential method
- Min-Max predictive method
- Centroid method

In this paper, we implement an enhancement of the centroid method- the multilevel centroid method.

• Sparse Matrix Storage Formats

Coordinate (COO) format [1][6]

This is the most general type of storage format. It uses three 1-D arrays- two to store the row and column indices, and one to store the corresponding data values. The space complexity of this format is given by Eq.(1).

$$S\ (COO) = 2\ .\ N.\ S\ (n) \qquad (1)$$

Where n is the order of the matrix and N is the total number of non-zeroes.

0 0 0 0

0 0 0 1

1 0 0 0

0 2 0 5

Figure 2: Example Matrix

The matrix in Fig. 2 can be represented as:
Data = [1 1 2 5]
Row = [1 2 3 3]
Column = [3 0 1 3]

Compressed Row Storage (CSR) Format [1][6]

This is the most widely used type of storage format, as it is both general and is very space efficient. The space complexity of this format is given by Eq.(2).

$$S(CSR) = N \,.\, S(n) + n \,.\, S(N) \qquad (2)$$

Where N is the total number of non-zeroes and n is the order of the matrix. For example, the matrix in Fig. 2 can be represented as:

Data = [1 1 2 5]
Column = [3 0 1 3]
Ptr = [0 0 1 2 4]

Quad Tree Compressed Row Storage (QCSR) Format [4][6]

This is a combination of the Quadtree format, which recursively divides the matrix into four quadrants, until each quadrant is of a given size (density), and the CSR format. These quadrants are classified into three categories:

- Empty node: If all the elements in the quadrant are zeroes.
- Mixed node: If there is a mix of zeroes and non-zero elements.
- Full node: If all the elements in the quadrant are non-zeroes.

For example, the matrix in Fig. 2 can be represented as shown in Fig. 3.

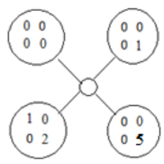


Figure 3: Quadtree Decomposition of Matrix in Fig. 1

In the QCSR format, first, the matrix is divided into quadrants like in the Quadtree format. The empty nodes are ignored and the full and the mixed nodes are stored using the CSR format. This format however incurs a space overhead because additional pointer values for the blocks need to be stored. It is used in applications where sparse matrix vector multiplication is necessary as it promotes very fast calculations.

The maximum overhead over the CSR format is given by Eq.(3).

$$S_{oh}\,(QCSR) = (2S_r + S_l) \times O(4d\text{-}1) \qquad (3)$$

Where, S_r is the space occupied by the index pointer to the region, S_l is the region length and d is the maximum depth of the tree. It is clear that the space complexity depends very much on the density chosen.

Minimal Quad Tree Format [5][6]

This storage format is used for efficient storage. One bit is produced for each quadrant of the matrix that is divided using the Quadtree method. We output a 0 if the quadrant is empty and 1 otherwise. This format can be used in cases where only the arrangement of the non-zero elements is necessary, as we do not store the value of the non-zero elements themselves. The matrix in Fig. 1 for example will be encoded as 0111. The minimum size of the MQT format is given by Eq.(4).

(4)

The maximum size of the MQT format is given by Eq.(5).

(5)

• Rectangular Segmentation-Single Image

Rectangular segmentation technique [9] overcomes the disadvantage of the quarter tree segmentation technique by allowing the sizes of the similar blocks to be of any size (need not be of size 2^n as like in quarter tree segmentation). It is a lossy image compression technique. The basic architectural diagram is shown in Fig. 4.

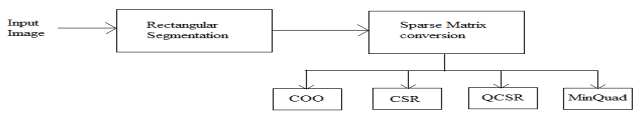


Figure 4: Rectangular Segmentation with Sparse Matrix

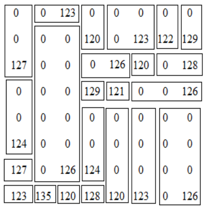
The following steps are followed to compress images using this technique.

- Starting from the first pixel, the pixels are compared with each other by column wise, and then row wise until they satisfy the consistency condition. The consistency condition is a threshold condition that the pixel should satisfy, which is the maximum absolute value of the difference between each pixel and the average value.
- Once the pixel in a particular row violates the condition, the pixels in the next row are taken into consideration. The minimum row and column values are taken to be the end vertices of the rectangle.
- All the satisfying pixels are then put into one block.
- Keeping only the right-bottom pixel value as such, and replacing all the other elements in the blocks with zeroes, we construct the reduced image. This will however lead to a certain amount of distortion in the image.

The reduced image constructed is a sparse matrix. Hence the image can be further compressed by representing this using the various sparse matrix storage formats discussed. Fig. 5(a) and Fig. 5(b) show the example matrix and the segmented matrix.

127	123	125	120	123	123	122	128
128	126	128	122	125	125	123	129
128	124	128	126	127	120	128	129
124	127	128	129	121	128	129	128
126	125	128	126	126	125	127	128
125	127	126	126	128	128	128	126
127	127	126	124	120	127	128	126
123	135	120	128	121	123	126	126

(a)



(b)

Figure 5: (a) Original Matrix (b) Segmented Matrix

We get the following matrices when we convert the given sample image into the various storage formats:

COO Format

Row= [0 1 1 1 1 2 2 2 2 2 3 3 3 5 6 6 6 7 7 7 7 7 7 7]

Column = [2 3 5 6 7 0 3 4 5 7 3 4 7 0 0 2 3 0 1 2 3 4 5 7]

Data= [123 120 123 122 129 127 126 126 120 128 129 121 126 124 127 126 124 123 135 120 128 120 123 126]

CSR Format

Column = [2 3 5 6 7 0 3 4 5 7 3 4 7 0 0 2 3 0 1 2 3 4 5 7]

Data= [123 120 123 122 129 127 126 126 120 128 129 121 126 124 127 126 124 123 135 120 128 120 123 126]

Ptr= [0 1 5 10 13 13 14 17 24]

QCSR Format

0	2	1	3
2	0	3	1

2	2	3	3
0	4	1	5
0	6	1	7
2	4	3	5
2	6	3	7
4	0	5	1
6	0	7	1
6	2	7	3
6	4	7	5
6	6	7	7

MinQuad format

1111011111001100000011111100010011111100101101100101011111111001100110001

- ## Multi-level Centroid Technique- Similar Images [13]

This is a set redundancy technique that can be used for storing sets of similar medical and satellite images. It is a lossless compression technique. An estimate image is obtained by calculating the median of all the pixel values in that particular position in all the images, and a set of difference images are found by subtracting the pixel value with the median value calculated. The mathematical model of the centroid method is given by Eq (6) and Eq (7)

$$F_{i+1,j} = m_{i+1,j} + x_{i,j} - m_{i,j} \quad (6)$$
$$D_{i+1,j} = x_{i+1,j} - F_{i+1,j} \quad (7)$$

The multi-level centroid technique involves applying the centroid technique recursively on the difference images until a sparse matrix is obtained. This is given by the following mathematical model by Eq(8) and Eq(9).

$$C_{i+1,j,l} = M_{l,i+1} + D_{i,j} - M_{l,i} \quad (8)$$
$$FD_{i+1,j,l} = D_{i+1,j} - C_{i+1,j,l} \quad (9)$$

Fig. 6 represents the block diagram of the flow of the multi-level centroid method, when coupled with the sparse matrix storage formats.

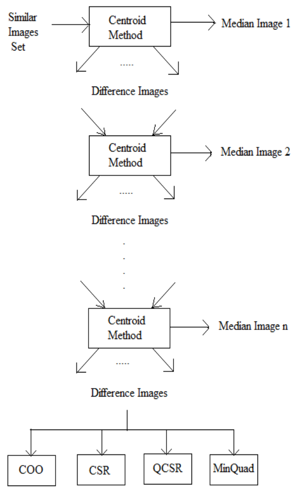


Figure 6: Multi-level Centroid Method with Sparse Matrix

The matrices in Fig. 7 show a sample of the evolution of the images when they are subjected to the above shown technique. The difference images of the last level are stored using the various sparse matrix formats.

Original image 1:

$$\begin{matrix} 6 & 6 & 7 & 7 & 8 & 7 & 7 & 7 \\ 6 & 6 & 6 & 7 & 7 & 7 & 7 & 7 \\ 6 & 6 & 6 & 6 & 7 & 7 & 7 & 7 \\ 6 & 6 & 6 & 6 & 6 & 7 & 7 & 8 \\ 6 & 6 & 6 & 6 & 6 & 7 & 7 & 8 \\ 6 & 6 & 6 & 6 & 6 & 6 & 7 & 8 \\ 5 & 5 & 6 & 6 & 6 & 6 & 7 & 8 \\ 5 & 5 & 5 & 6 & 6 & 6 & 6 & 7 \end{matrix}$$

Difference image 1:

$$\begin{matrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Median image 1:

$$\begin{matrix} 6 & 6 & 7 & 7 & 8 & 7 & 7 & 7 \\ 6 & 6 & 6 & 7 & 7 & 7 & 7 & 7 \\ 6 & 6 & 6 & 6 & 7 & 7 & 7 & 7 \\ 6 & 6 & 6 & 6 & 6 & 7 & 7 & 8 \\ 6 & 6 & 6 & 6 & 6 & 7 & 7 & 8 \\ 6 & 6 & 6 & 6 & 6 & 6 & 7 & 8 \\ 5 & 5 & 6 & 6 & 6 & 6 & 7 & 8 \\ 5 & 5 & 5 & 6 & 6 & 6 & 6 & 7 \\ 5 & 5 & 5 & 6 & 6 & 6 & 6 & 7 \end{matrix}$$

Median image 2:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

Figure 7: Matrices when Using Multi-level Centroid Technique

The difference images obtained in the last level of the implementation are sparse matrices. Hence the various sparse matrix storage formats that were discussed can be used to further increase the compression ratio of the images. The shown difference image can be represented as follows:

COO Format

Data= [1 1 1 1 -1 1 1 1 1 1 1]
Column= [2 4 3 4 7 5 0 1 2 6 7]
Row= [0 0 1 2 2 4 5 5 6 6 6]

CSR Format

Data= [1 1 1 1 -1 1 1 1 1 1 1]
Column= [2 4 3 4 7 5 0 1 2 6 7]
Ptr= [0 2 3 5 5 6 8 11 11]

QCSR Format

0	2	1	3
0	4	1	5
2	4	3	5
2	6	3	7
4	0	5	1
6	2	7	3
4	4	5	5
6	6	7	7

Min Quad Format

1111010010011011100010000100100100111000100101001100

- ## Experimental Results

We first took 8x8 blocks of images and compressed it using the rectangular segmentation technique and the sparse matrix storage formats. We assumed two threshold values (Q) - 2 and 5. We calculated the space complexity when using different formats, and also the Peak Signal to Nosie Ratio (PSNR) for the two thresholds to compare the level of distortion obtained. Table 1 shows the results obtained for the same. This shows that the Min quad format gives the best results. However, since it is not possible to reconstruct the image using this format, we use the next best case, the CSR format. The PSNR values of the reconstructed images are high. This indicates that there is not much distortion. When we increase the threshold value, the PSNR value decreases as there is an increase in distortion, but there is also an increase in the compression ratio.

Table 1: Compression Ratio when Using Sparse Matrix Storage Formats with Different Threshold Values (Q) and their Corresponding PSNR Values with Rectangular Segmentation Technique

Images	**RSSMS with COO**		**RSSMS with CSR**		**RSSMS with QCSR**		**RSSMS with MinQuad**		**PSNR**	
	Q=2	Q=5	Q=2	Q=5	Q=2	Q=5	Q=2	Q=5	Q=2	Q=5
Boat	1.56	2.99	1.82	3.63	2.88	5.34	5.19	7.79	49.5056	43.0044
Lena	1.98	5.21	2.48	5.3	3.53	7.95	6.11	11.35	49.1509	43.046
Goldhill	1.91	5.42	2.43	6.45	3.12	12.46	5.5	18.7	49.599	42.862

Next, we took various images as a whole of size 256 x 256 and of type bmp and performed the same. Table 2 shows the results of the space complexities obtained. The threshold was assumed to be 2.

Table 2: Space Complexities of Images when Stored Using Various Sparse Matrix Storage Formats (in bytes) with Rectangular Segmentation Technique

Images	**Original**	**RSSMS**	**RSSMS with COO**	**RSSMS with CSR**	**RSSMS with QCSR**	**RSSMS with MinQuad**
Barbara	1186	777	764	1334	984	314
Boat	1227	796	743	1348	1045	329
Goldhill	1170	741	612	1215	1024	324
Lena	1020	717	336	1047	964	309
Peppers	1146	725	519	1149	1007	320
Zelda	1142	686	250	969	980	314

From Table 2, it is apparent that next to the Min Quad format, the COO format gives the best results. On analysis, it was noticed that this was because as the size of the image increases, the number of non-zero elements in the sparse matrix increases. This large number when stored in increasing multiple times in the ptr array of the CSR format causes a space overhead. Hence, for larger images, the COO format coupled with the RSSMS technique gives the best result. Fig. 8 shows the graph of the compression ratio of rectangular segmentation for Lena image when using various storage formats.

Figure 8: Graph Showing the Compression Ratio of Using Various Sparse Matrix Storage formats with Rectangular Segmentation Technique

Next, we analysed the effect of introducing sparse matrix storage formats in the Multi-level centroid technique for a set of similar images (Brain CT scan images). Fig. 9 shows the results obtained for the same. We assumed the number of levels to be 2 as discussed in paper [13] that 2 is the optimum number of levels needed to increase the compression ratio. From Fig. 9, it is clear that the usage of these sparse matrix storage formats is not suitable for this technique as it does not give a reasonable level of compression. On analysis, we found that each non-zero element in the difference image, when stored in COO format needs three integer values (row, col and data) to store its complete data. Each integer requires 4 bytes of data. Each non-zero element thus requires (3 x 4 x 8) 96 bits of storage space. The conventional compression techniques of Huffman coding and Arithmetic coding however require much less number of bits for storing each pixel. Hence, usage of sparse matrix storage formats in multi-level centroid technique is not suitable. Though Minimal Quadtree gives a certain level of compression, as mentioned earlier, it will not be possible to reconstruct the image using this technique.

Figure 9: Graph Showing the Compression Ratios of Various Sets of Images when Using Multi-level Centroid Technique along with Various Sparse Matrix Storage Formats

• Conclusion

In this paper, we analysed the effects of introducing various sparse matrix storage formats in various image compression techniques. In the Rectangular segmentation technique for single image lossy image compression, we found that for smaller images, the CSR format gives the best results, while for larger images; the COO format gives optimum results. In the case of Multi-level Centroid technique we found that the usage of these formats does not give a positive result. Hence for this technique, Huffman and Arithmetic coding must be used to obtain further compression.

References

[1]. Bell N., Garland, M., Efficient sparse matrix-vector multiplication on CUDA Nvidia Technical Report NVR-2008-004, Nvidia Corporation 2(5) (2008).

[2]. Sun X., Zhang Y., Wang T., Zhang X., Yuan L., Rao L., Optimizing SpMV for diagonal sparse matrices on GPU, International Conference on In Parallel Processing (2011), 492-501.

[3]. Guo D., Gropp W., Optimizing Sparse Data Structures for Matrix–vector Multiply, International Journal of High Performance Computing Applications (2011).

[4]. Jilin Zhang, Enyi Liu, Jian Wan, Yongjian Ren, Miao Yue, Jue Wang, Implementing Sparse Matrix-Vector Multiplication with QCSR on GPU, Journal of Applied Mathematics & Information Sciences (2013).

[5]. Simecek I., Langer D., Tvrdik P., Minimal Quadtree Format for Compression of Sparse matrices storage, 14th International Symposium on Symbolic and Numeric Algorithms for scientific computing (2012).

[6]. Saira Banu J., Vaideswaran J., Sumithra S., Performance Analysis on Parallel Sparse Matrix Vector Multiplication Micro-Benchmark Using Dynamic Instrumentation Pintool, ARTCOM (2013).

[7]. Zhang Tianxu, Zeng Yonghui, An effective image compression algorithm based on sparse matrix, journal of Huazhong University of Science and Technology (Natural Science Edition), 34(2) (2006).

[8]. He Xing-heng, Chen Hui, Method based on Quarter-tree method of Decomposition, Journal of Computer Engineering and applications 9 (2008).

[9]. Shengli Chen, Xiaoxin Cheng, Jiapin Xu, Research on Image Compression Algorithm based on Rectangle Segmentation and Storage with Sparse Matrix, 9th International Conference on Fuzzy Systems and Knowledge Discovery (2012), 1904-1908.

[10]. Tsai Y.C., Lee M.S., Shen M., Kuo, C.C.J., A Quad-Tree Decomposition Approach to Cartoon Image Compression, IEEE 8th Workshop on Multimedia Signal Processing (2006).

[11]. Samy Ait-Aoudia, Abdelhalim Gabis, A Comparison of Set Redundancy Compression Techniques, Hindawi Publishing Corporation, EURASIP Journal on Applied Signal Processing, (2006), 1–13.

[12]. Kosmas Karadimitriou, Set redundancy, the enhanced compression Model, and methods for compressing Sets of similar images, Dissertation (1996).

[13]. Yasser El-Sonbaty, Meer Hamza, George Bacily, Enhanced Centroid Technique for Compressing Sets of Similar Medical Images, 46th Midwest Symposium on Circuits and Systems-IEEE 1 (2003), 348-351.

[14]. Tanaya Guha, Rabab K. Ward, Image Similarity Using Sparse Representation and Compression Distance, IEEE Transactions on Multimedia 16(4) (2014).

[15]. Kitty Arora, Manshi Shukla, A Comprehensive Review of Image Compression Techniques, International Journal of Computer Science and Information Technologies 5(2) (2014), 1169-1172.

[16]. Kitty Arora, Manshi Shukla, Performance Comparison of K-means and Rectangular Segmentation Algorithms in Compression of Colour Images, International Journal of Computer Science and Information Technologies 5(5) (2014), 6838-6845.

[17]. Rajesh, M., and J. M. Gnanasekar. "Multiple-Client Information Administration via Forceful Database Prototype Design (FDPD)." IJRESTS 1.1 (2015): 1-6.

[18]. Rajesh, M. "Control Plan transmit to Congestion Control for AdHoc Networks." Universal Journal of Management & Information Technology (UJMIT) 1 (2016): 8-11.

[19]. Rajesh, M., and J. M. Gnanasekar. "Consistently neighbor detection for MANET." Communication and Electronics Systems (ICCES), International Conference on. IEEE, 2016.