Quadtree-Based Centroid Technique for Compressing Sets of Similar Medical Images

Yasser El-Sonbaty
Arab Academy for Science & Technology
Dept. of Computer Sci.
Alexandria 1029, Egypt
yasser@aast.edu

Meer Hamza
Arab Academy for Science & Technology
Dept. of Computer Sci.
Alexandria 1029, Egypt
mhamza@aast.edu

George Awad
Arab Academy for Science & Technology
Dept. of Computer Eng.
Alexandria 1029, Egypt
mbawad@aast.edu

Abstract – **The need for Lossless data compression in medical imaging is becoming essential. Medical image databases often store large collections of similar images. Traditional compression techniques focused on exploiting redundancy presented in individual images ignoring the set redundancy, which is the inter-image redundancy. Previous research has introduced the centroid method, which gets benefit from the set redundancy. In this paper a new algorithm is proposed as an extension of the centroid method combined with the quadtree structure widely used before to represent binary images. Experimental results with two sets of CT and MRI brain images demonstrate the efficiency and superiority of the proposed algorithm in respect to compression ratio.**

I. INTRODUCTION

Increasingly, medical images are acquired or stored digitally. This is especially true of grayscale images that are used in radiology applications. These images may be very large in size and number, hence Lossless compression offers a perfect mean to reduce the cost of storage, increase the speed of transmission [1], and save image fidelity. Recent techniques for compressing images concentrate on how to reduce the redundancy presented in individual images such as the interpixel redundancy, the coding redundancy and the psychovisual redundancy [11]. Whereas few research has focused on how to get benefit from redundancy between set of similar images which appears in many application areas such as in medical images inside a hospital database or in geographical information systems. In this paper a new algorithm is proposed as an extension to the existing centroid algorithm [2] for compressing a set of similar medical images. The rest of this paper is organized as follows. Section 2 contains a review on related work. In section 3, the proposed algorithm is introduced. The experimental results on CT (Computed Tomography) and MRI (Magnetic Resonance Imaging) data sets are presented in section 4. Finally, conclusions are discussed in section 5.

II. RELATED WORK

Approaches for compressing similar images can be classified into two main categories. The set redundancy category [10], which is based on predictive coding to get benefit from the inter-image redundancy and statistical correlation between images. Examples of set redundancy methods are the Min-Max Differential (MMD) [3], the Min-Max predictive (MMP) [4] and the centroid method [2]. While the second category defines different methods based on quadtrees for the representation and manipulation of clusters of images. Examples of such approach are the overlapping quadtrees [9] for storing sequence of images in overlapped quadtrees, the inverted and generic quadtrees [5,6,8] that uses one quadtree structure to store a cluster of images. The next sections review the centroid method and quadtree decomposition principle that the proposed algorithm is based on.

A. Centroid Method

The centroid method [2] depends on predictive decorrelation where an estimate of the image is obtained and then subtracted from the original image. If the prediction is efficient enough, the difference image will contain small values and has a laplacian distribution with most of the values very close to zero. For a set of k images with N pixels per image, the formula for predicting the value of a pixel i in image j can be expressed as follows:

$$C_{i,j} = m_i \qquad (1)$$

Where $C_{i,j}$ is the predicted value and m_i is the average value of position i across all images. This model is simple but not very efficient. A more advanced model is also proposed as follows:

$$C_{i+1,j} = m_{i+1} + x_{i,j} - m_i \qquad (2)$$
$$D_{i+1,j} = x_{i+1,j} - C_{i+1,j} \qquad (3)$$

Where C_{i+1} is an estimate at position i+1 in image j, xi,j is the pixel at position i in image j, m_i is the average value of position i across all images, and $D_{i+1,j}$ is the difference value of position i+1 in image j between the original value and the predicted one. The detailed derivation of (2) and (3) is shown in [2]. Equation (2) is so called the centroid method. The median image can be used instead of the average image in the centroid method to reduce the influence of outliers. The set redundancy methods are fast, lossless, easy to implement, and can compress and decompress individual images from the set without requiring global calculations on the whole set.

B. Quadtree Decomposition

The quadtree [12,13] is one of the widely used structures for image representation, especially for binary images. This structure is efficient to store 2D images and has been frequently used in the field of computer graphics [14] and content-based image retrieval [15]. Recursive division of the space in four quadrants or squares of the same size build a quadtree so that a node of the quadtree represents a quadrant. The root node represents the initial quadrant containing the whole image. The most widely known quadtree allows cutting an image in regions or quadrants according to a given criterion. If an image is not homogeneous (according to a particular criterion), the quadtree root has four descendant nodes representing the four first level image quadrants. A node is a leaf when its corresponding image quadrant is homogeneous; otherwise the node is internal [16]. In general, the quadtree is unbalanced.

III. PROPOSED ALGORITHM

The proposed Centroid Quadtree algorithm is a combination between the set redundancy and quadtree-based approaches. The new algorithm applies the centroid method [2] on a segmented image where segmentation is done using quadtree decomposition to group connected pixels lying within a small dynamic gray level range. The algorithm was tested using two sets of medical images: CT and MRI, and has shown an improved performance with respect to compression ratio.

A. Centroid Quadtree Algorithm

The new technique model can be demonstrated in Fig. 2. Assume that there is a set of similar images $X = \{x_1, x_2, \ldots\ldots x_N\}$ that we want to process. The corresponding median or average image *m* is determined (centroid image in Fig. 2). Using the quadtree decomposition principle, we recursively divide image *m* into quadrants of homogeneous pixel values. We define the homogeneity of the quadrants by being able to represent the pixel values inside the quadrant using *n* bits per quadrant (difference between the maximum and minimum values of a quadrant can be represented in *n* bits). To minimize the number of quadrants, we applied the quadtree decomposition using a condition of *s* x *s* minimum quadrant size, i,e if the quadrant is not homogeneous and its size is *s* x *s*, then it is assumed that it is a leaf node in the quadtree. The minimum value of each quadrant (base value) is subtracted from all quadrant pixel values and stored at the base vector. This subtraction reduces the dynamic gray level range of pixel values in the centroid image *m*. Two structures are obtained from the decomposition of image *m*; the quadtree structure and quadrants base vector.

The input image x_i to be compressed is then decomposed using the same quadtree structure of the centroid image *m* and the base value of each quadrant is subtracted from all quadrant pixel values generating the quadrants base vector. The last step applies the centroid method [2] on the input image x_i using the centroid image *m*. The centroid method is also applied to compress the base vector of the input image using the centroid image bases. Standard entropy encoders can then be used to compress the difference output image and the base vector. Note that reducing the dynamic gray-level range of values in both, the centroid and input image helps to improve the compression performance. In the decompression process, the output of the entropy decoder is processed by the centroid method for both the image data and base vector, and then the bases are added to the quadrant pixel values to retrieve the original image. Fig.1 show an illustrative example on applying the centroid quadtree algorithm on part of a CT brain image, where Fig.1-a shows a 16x16 8 b/p original data image, the corresponding average image is shown in Fig.1-b with its quadrants (22 block) which was decomposed by quadtree with splitting condition of 3 bit/quadrant and minimum quadrant size of 2x2. After decomposition of the average image, the base values are subtracted from each quadrant and the updated values are shown in Fig.1-c. Quadtree decomposition is then applied on the input image with the same decomposition structure that was used in the average image. Similarly, the input image base values are subtracted from its quadrants and the updated values are shown in Fig.1-d. The difference image after applying the centroid method on the updated values are shown in Fig.1-e, on the other hand Fig.1-f shows the difference image if the traditional centroid method is used on the original data. Base values of both images are shown in figures 1-g and 1-h with their difference from the centroid method in Fig.1-i. Comparing the entropy values of both difference images can test improvements of the new algorithm. The new algorithm gives entropy of 1.0 while the traditional centroid method gives 2.585, also observing the minimum and maximum values in both difference images, we notice that in Fig.1-e data can be represented in 3 bits while in Fig.1-f it requires 4 bits.

IV. EXPERIMENTAL RESULTS

The performance of the proposed algorithm was tested using 10 512x512 brain CT images, and 10 256x256 MRI brain images. Figures 3 and 4 show the test images used of each group. Both groups of images are 8 bits/pixel gray-level images, each set of images are similar to each other, and are the same test images that were used by [2] in testing the centroid method. They were collected at M.D. Anderson Cancer Center in Houston, Texas out of a small image database with 51 CT brain images and 57 MRI brain images, from random patients of both sexes, and ages, and a variety of pathological conditions [2]. The algorithm has been tested with variations in quadtree splitting condition from 3 bits up to 6 bits and minimum quadrant sizes from 1x1 up to 8x8 block size, that to demonstrate the performance of the new algorithm with different parameters. The algorithm was tested in combination with entropy-based compression techniques: Huffman encoding and Arithmetic encoding. The recursive splitting method of Huffman and Arithmetic encoding that was shown in [7] was used as the implementation method for both techniques. The compression ratio is defined as:

$$C = \text{original size / compressed size} \quad (4)$$

$$\text{Compressed size} = \text{compressed image data} + \text{compressed bases} \quad (5)$$

$$\text{Improvement \%} = \frac{C_{new} - C_{old}}{C_{old}} * 100 \quad (6)$$

Where C_{old} is the compression ratio of other techniques and C_{new} is the compression ratio of the proposed algorithm.

76	76	76	75	76	76	76	76	76	76	76	75	75	75	75	75
76	76	76	76	76	76	76	76	76	76	76	75	75	75	75	75
76	76	76	76	76	76	76	76	76	76	76	75	75	75	75	75
76	76	76	76	76	76	76	76	76	76	76	75	75	75	75	75
76	76	77	76	76	76	76	76	76	76	76	76	75	75	76	76
77	77	77	77	77	77	77	77	77	76	76	76	76	76	76	76
78	78	78	77	77	77	78	78	77	77	77	77	76	76	77	77
79	79	79	79	78	79	79	79	79	78	78	77	77	78	78	78
81	81	81	80	80	80	80	80	80	79	79	79	79	79	79	79
81	81	81	80	80	80	80	80	80	79	79	79	79	79	79	79
83	83	83	82	82	82	82	82	81	81	81	81	80	80	80	80
86	85	85	84	84	84	83	83	83	82	82	82	82	82	82	82
87	87	87	87	86	86	85	85	84	84	84	83	83	83	84	84
89	88	88	88	88	87	87	87	86	85	85	85	85	85	85	85
88	88	88	88	88	88	88	88	87	87	87	86	86	86	86	86
81	82	82	83	83	84	85	85	85	85	85	85	85	85	85	84

a- Original 16 x16 image data

74	74	73	74	74	74	74	74	74	73	74	74	74	74	74	74
74	73	73	74	74	74	74	73	74	74	74	74	74	74	74	74
73	73	73	74	73	74	74	74	74	74	74	74	74	74	74	74
74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74
74	74	74	74	74	74	74	74	74	74	74	75	74	74	74	74
75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	74
76	76	76	75	76	76	76	76	76	76	76	76	75	75	75	75
77	77	77	77	77	77	77	77	77	77	77	76	76	76	76	76
79	78	79	78	78	79	79	79	79	79	78	78	78	78	78	78
80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80
82	82	81	82	82	82	82	82	82	82	82	82	81	81	82	82
83	83	84	84	84	84	83	84	84	84	84	83	83	83	83	83
84	85	85	85	85	85	85	85	85	85	85	84	85	84	84	84
85	85	85	85	85	85	85	85	85	85	85	85	85	84	84	84
82	82	83	84	84	84	84	83	83	83	83	82	82	82	82	82
75	75	77	77	77	77	77	76	76	76	76	75	76	75	75	76

b- Average image with quadtree decomposition

1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
3	3	3	2	3	3	3	3	3	3	3	3	2	2	2	2
4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3
1	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	4	3	4	4	4	4	4	4	4	4	4	3	3	4	4
5	5	6	6	6	6	5	6	6	6	6	5	5	5	5	5
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0
7	7	6	7	7	7	8	7	7	7	8	7	7	7	7	7
0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1

c- Average image after subtracting the quadrants base values

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

d- Original image after subtracting the quadrants base values

0	0	1	-2	1	0	0	0	0	1	-1	-1	0	0	0	0
0	1	0	-1	0	0	0	1	-1	0	0	-1	0	0	0	0
1	0	0	-1	1	-1	0	0	0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
0	0	1	-1	0	0	0	0	0	0	0	-1	0	0	1	0
0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	1
0	0	0	0	-1	0	1	0	-1	0	0	0	0	0	1	0
0	0	0	0	-1	1	0	0	0	-1	0	0	0	1	0	0
0	1	-1	0	0	-1	0	0	1	-1	1	0	0	0	0	0
-1	0	0	-1	0	0	0	0	1	-1	0	0	0	0	0	0
-1	0	1	-2	0	0	0	0	0	0	0	0	0	0	-1	0
1	-1	-1	-1	0	0	0	-1	1	-1	0	1	0	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0	-1	1	0	0
1	-1	1	0	1	-1	1	0	0	-1	0	0	0	1	-1	0
0	0	0	-1	-1	0	-3	1	-1	0	-1	0	0	0	1	0
0	1	-1	1	-1	1	-2	1	0	0	-1	1	-1	1	1	-2

e- Difference image of the new centroid quadtree algorithm
(entropy = 1.0)

2	0	1	-2	1	0	0	0	0	1	-1	-1	0	0	0	0
2	1	0	-1	0	0	0	1	-1	0	0	-1	0	0	0	0
3	0	0	-1	1	-1	0	0	0	0	0	-1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
2	0	1	-1	0	0	0	0	0	0	0	-1	0	0	1	0
2	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	1
2	0	0	0	-1	0	1	0	-1	0	0	0	0	0	1	0
2	0	0	0	-1	1	0	0	0	-1	0	0	0	1	0	0
2	1	-1	0	0	-1	0	0	0	-1	1	0	0	0	0	0
1	0	0	-1	0	0	0	0	0	-1	0	0	0	0	0	0
1	0	1	-2	0	0	0	0	-1	0	0	0	0	0	-1	0
3	-1	-1	-1	0	0	0	-1	0	-1	0	1	0	0	0	0
3	-1	0	0	-1	0	-1	0	-1	0	0	0	-1	1	1	0
4	-1	0	0	0	-1	0	0	-1	-1	0	0	0	1	0	0
6	0	-1	-1	0	0	0	1	-1	0	0	0	0	0	0	0
6	1	-2	1	0	1	1	1	0	0	0	1	-1	1	0	-2

f- Difference image of the traditional centroid method
(entropy = 2.585)

73	78	84	75	85 …

g- Average image base vector

75	80	87	81	87 …

h- Input image base vector

2	0	1	3	-4 …

i- Difference base vector

Fig. 1. Example of applying the centroid quadtree on a 16 x 16 section of a brain CT image

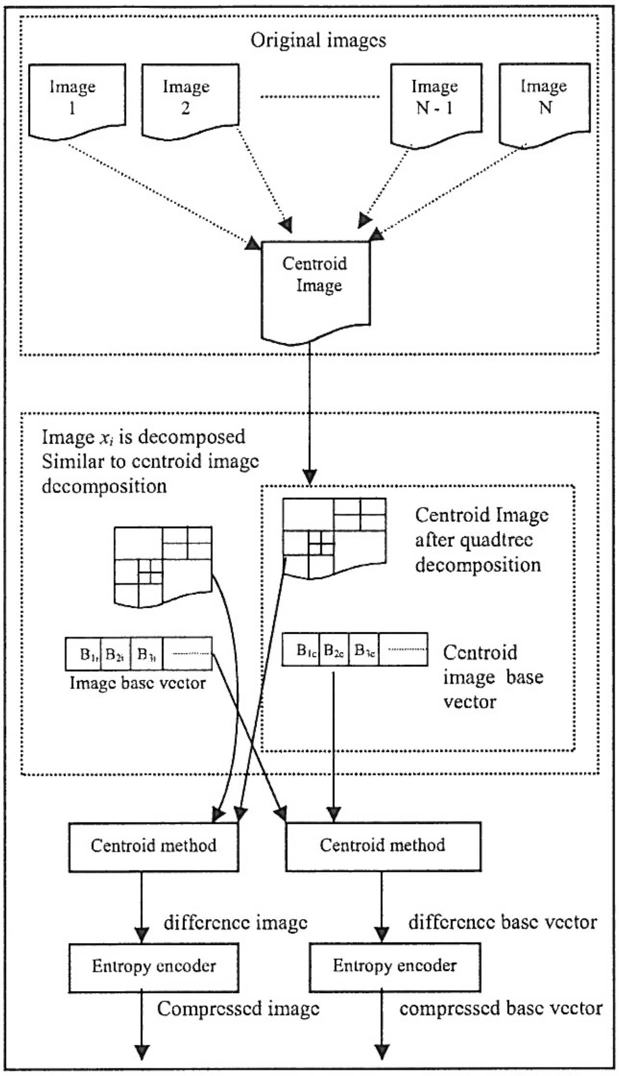

Fig. 2. Centroid quadtree compression technique model

Tables 1 and 6 show the results obtained of applying the centroid method using Huffman and Arithmetic encoders on CT and MRI images.

A. CT Experimental Results

Tables 1 to 5 show the CT experimental results of the average compression ratio achieved when applying the proposed centroid quadtree algorithm on the test images shown in Fig. 3 using the average and median images with bit range of 3,4,5 and 6 bits/quadrant quadtree splitting condition and minimum quadrant size range of 1x1, 2x2, 4x4, and 8x8 pixels. From tables 1, 2 and 4, it can be concluded that the improvement range over traditional centroid method [2] using the average image varies from 21% to 24% for both the Huffman and Arithmetic encoders respectively. On the other hand from tables 1, 3 and 5, improvement range reached from 18% to 20% for both Huffman and Arithmetic encoders respectively using the median image. It can be noticed that maximum average compression ratio is achieved using 6 bits/quadrant and minimum quadrant size 8x8. In comparison with standard entropy encoders, the new algorithm improved the compression performance of Huffman encoder with about 86% to 130% while improved the arithmetic encoder by 53% to 97%.

B. MRI Experimental Results

Tables 6 to 10 show the MRI experimental results of the average compression ratio achieved when applying the proposed centroid quadtree algorithm on the test images shown in Fig. 4 using the average and median images with bit range of 3,4,5 and 6 bits/quadrant quadtree splitting condition and minimum quadrant size range of 1x1, 2x2, 4x4, and 8x8 pixels. From tables 6, 7 and 9, it can be concluded that the improvement range over traditional centroid method [2,10] using the average image varies from 10% to 11% for both the Huffman and Arithmetic encoders respectively. On the other hand from tables 6, 8 and 10, improvement range reached from 8% to 11% for both Huffman and Arithmetic encoders respectively using the median image. Similar to CT results, the best results are achieved with 6 bits/quadrant and 8x8 minimum block size parameters. In comparison with standard entropy coders, the new algorithm improved the compression performance of Huffman encoder with about 14% to 41% while improved the Arithmetic encoder by 6% to 31%. In general the compression ratio of MRI images is low due to the low signal to noise ratio in that type of images in comparison to the CT images.

TABLE 1. CT AVERAGE COMPRESSION RATIO OF STANDARD ENTROPY METHODS AND TRADITIONAL CENTROID METHOD

	Compression ratio
Standard compression method (Huffman)	1.3790 : 1
Standard compression method (Arithmetic)	1.7010 : 1
Centroid method (using average image) + Huffman	2.6200 : 1
Centroid method (using median image) + Huffman	2.5940 : 1
Centroid method (using average image) + Arithmetic	2.7010 : 1
Centroid method (using median image) + Arithmetic	2.6734 : 1

TABLE 2. CT AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE AVERAGE IMAGE USING HUFFMAN ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	2.5853:1	2.8150:1	3.0209:1	3.1596:1
Min 2x2 block size	2.6323:1	2.8214:1	3.0209:1	3.1596:1
Min 4x4 block size	2.8617:1	2.9230:1	3.0354:1	3.1600:1
Min 8x8 block size	3.0512:1	3.0656:1	3.0948:1	3.1671:1

TABLE 3. CT AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE MEDIAN IMAGE USING HUFFMAN ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	2.5702:1	2.7011:1	2.8927:1	3.0349:1
Min 2x2 block size	2.6171:1	2.7313:1	2.8967:1	3.0349:1
Min 4x4 block size	2.8438:1	2.8796:1	2.9470:1	3.0402:1
Min 8x8 block size	3.0020:1	3.0126:1	3.0282:1	3.0693:1

TABLE 4. CT AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE AVERAGE IMAGE USING ARITHMETIC ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	2.6557:1	2.9313:1	3.1743:1	3.3360:1
Min 2x2 block size	2.7253:1	2.9410:1	3.1743:1	3.3360:1
Min 4x4 block size	2.9850:1	3.0560:1	3.1906:1	3.3365:1
Min 8x8 block size	3.2077:1	3.2253:1	3.2623:1	3.3461:1

TABLE 5. CT AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE MEDIAN IMAGE USING ARITHMETIC ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	2.5966:1	2.7750:1	2.9991:1	3.1641:1
Min 2x2 block size	2.6995:1	2.8235:1	3.0063:1	3.1641:1
Min 4x4 block size	2.9460:1	2.9867:1	3.0635:1	3.1693:1
Min 8x8 block size	3.1257:1	3.1386:1	3.1567:1	3.2036:1

TABLE 6. MRI AVERAGE COMPRESSION RATIO OF STANDARD ENTROPY METHODS AND TRADITIONAL CENTROID METHOD

	Compression ratio
Standard compression method (Huffman)	1.2940 : 1
Standard compression method (Arithmetic)	1.4040 : 1
Centroid method (using average image) + Huffman	1.6500 : 1
Centroid method (using median image) + Huffman	1.6500 : 1
Centroid method (using average image) + Arithmetic	1.6500 : 1
Centroid method (using median image) + Arithmetic	1.6097 : 1

TABLE 7. MRI AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE AVERAGE IMAGE USING HUFFMAN ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	1.5139:1	1.6166:1	1.7297:1	1.7976:1
Min 2x2 block size	1.5744:1	1.6598:1	1.7429:1	1.7984:1
Min 4x4 block size	1.7242:1	1.7426:1	1.7791:1	1.8065:1
Min 8x8 block size	1.8068:1	1.8079:1	1.8160:1	1.8233:1

TABLE 8. MRI AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE MEDIAN IMAGE USING HUFFMAN ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	1.4776:1	1.5697:1	1.6725:1	1.7396:1
Min 2x2 block size	1.5371:!	1.6168:1	1.6954:1	1.7443:1
Min 4x4 block size	1.6907:1	1.7045:1	1.7371:!	1.7600:1
Min 8x8 block size	1.7666:1	1.7670:1	1.7741:1	1.7802:1

TABLE 9. MRI AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE AVERAGE IMAGE USING ARITHMETIC ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	1.5212:1	1.6246:1	1.7402:1	1.8084:1
Min 2x2 block size	1.5836:1	1.6689:1	1.7525:1	1.8093:1
Min 4x4 block size	1.7348:1	1.7534:1	1.7894:1	1.8174:1
Min 8x8 block size	1.8193:1	1.8204:1	1.8278:1	1.8353:1

TABLE 10. MRI AVERAGE COMPRESSION RATIO OF APPLYING PROPOSED ALGORITHM ON THE MEDIAN IMAGE USING ARITHMETIC ENCODER

Centroid Quadtree	3 bit	4 bit	5 bit	6 bit
Min 1x1 block size	1.4824:1	1.5748:1	1.6796:1	1.7478:1
Min 2x2 block size	1.5440:1	1.6243:1	1.7040:1	1.7528:1
Min 4x4 block size	1.6989:1	1.7126:1	1.7460:1	1.7693:1
Min 8x8 block size	1.7773:1	1.7777:1	1.7847:1	1.7911:1

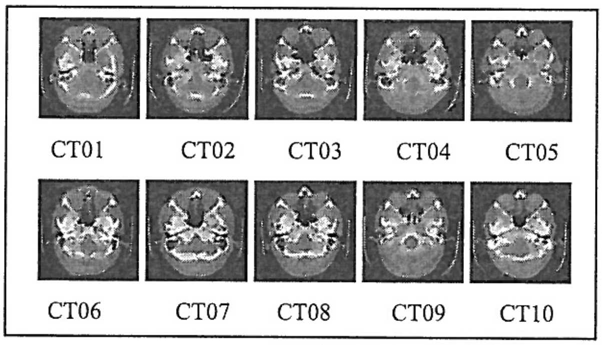

Fig. 3. CT brain test images

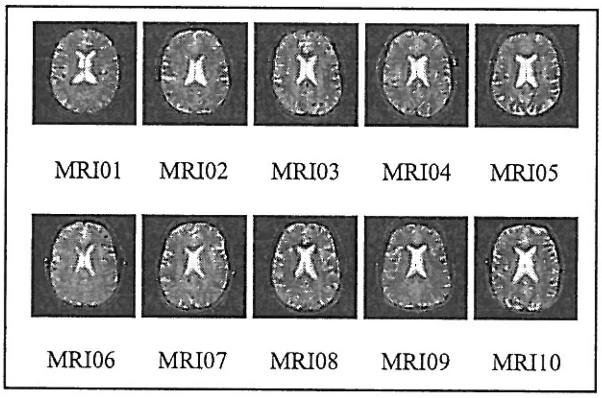

Fig. 4. MRI brain test images

V. DISCUSSIONS AND CONCLUSIONS

In this paper, a new algorithm has been proposed for compressing a set of similar medical images. The centroid quadtree algorithm is a combination between the set redundancy based approach and quadtree based approach. Segmenting the image into quadrants of small dynamic gray-level range and then applying the centroid algorithm helps to improve the compression ratio of the lossless compression methods. Experimental results have shown that choosing different parameters to split quadrants and choosing minimum quadrant size affect the final compression ratio. Choosing low bits/quadrant splitting condition leads to increased number of bases as size of quadrants become smaller. Similarly, low minimum quadrant size gives the chance to increase number of bases. In general, high number of bases reduces the compression ratio because it is added to the compressed image size. On the other hand, high bits/quadrant or high minimum quadrant size does not consume high number of bases and reduces their total overhead. Also choosing these parameters can depend on the type of image and its structure or regularity. There is a need to compromise between different parameters that gives optimum number of bases with high total compression ratio. Improvement of compression ratio over traditional centroid algorithm [2] has reached 24% in CT images, 11% in MRI images and improved the standard Huffman entropy encoder by 130% in CT images and 41% in MRI images, while improved the standard Arithmetic entropy encoder by 97% in CT images and 31% in MRI images.

VI. ACKNOWLEDGEMENTS

The authors would like to thank Dr. K.Karadimitriou for his helpful comments, and offering test datasets of CT and MRI brain images.

VII. REFERENCES

[1] D. Clunie, "Lossless compression of grayscale medical images – Effectiveness of traditional and state of the art approaches", *in Proc. SPIE (Medical Imaging)*, Vol. 3980, Feb. 2000.

[2] K. Karadimitriou et al., "Centroid method for compressing sets of similar images", *Pattern Recognition Letters*, vol.19, no.7, pp.585- 593, 1998.

[3] K. Karadimitriou and J.M. Tyler, "The Min-Max Differential Method for Large-Scale Storage and Compression of Medical Images", *in Proc of Annual Molecular Biology and Biotechnology Conference*, Baton Rouge, Feb. 1996.

[4] K. Karadimitriou and J.M. Tyler, "Min-Max Compression Methods for medical image databases", *ACM SIGMOD* Record 26, 47-52, 1997.

[5] J.P. Cheiney, A. Touir, "FI-Quadtree: a new data structure for content-oriented retrieval and fuzzy search", *Proceedings of the Second Symposium on Spatial Databases (SSD'91)*, 1991.

[6] G. Jomier, M. Manouvrier, M.Rukoz, "Storage and management of similar images", *Journal of the Brazilian Computer Society (JBCS)* 3 (6) (2000) 13-26.

[7] K. Skretting, J.H.HusØy, S.O.Aase, "Improved Huffman Coding Using Recursive Splitting", *NORSIG-99 Conference on Signal Processing and Signal Compression*, 9-11 sep, 1999.

[8] M. Vassilakopoulos, Y. Manolopoulos, "Dynamic inverted quadtree- a structure for pictorial databases", *information systems, Special Issue on Multimedia Information Systems* 20 (6) (1995) 483-500.

[9] M. Vassilakopoulos, Y. Manolopoulos, K. Economou, "Overlapping quadtrees for the representation of similar images", *Image and Vision Computing* 11 (5) (1993) 257- 262.

[10] K. Karadimitriou, "Set Redundancy, the Enhanced Compression Model, and Methods for Compressing Sets of Similar Images", Ph.D. Dissertation, Louisiana University, Baton Rouge, LA, Aug. 1996.

[11] M. Bassiouni , M.A., "Data compression in scientific and statistical databases", *IEEE Trans. On Software Engineering*, vol. 11, no. 10, pp. 1047-1058, Oct. 1985.

[12] R.A.Finkel, J.Bentley, "Quadtrees: a data structure for retrieval on composite keys", *Acta Informatica* 4 (1) (1974) 1- 9.

[13] H. Samet, "The quadtree and related hierarchical structures", *Computer Surveys* 16 (2) (1984) 187-260.

[14] H.Samet, R.E. Webber, "Hierarchical data structures and algorithms for computer Graphics and Applications", 8 (5) (1988) 48-68.

[15] H. Lu, B.-C. Ooi, K.-L. Tan, "Efficient Image Retrieval by Color Contents", *International Conference on Applications of Database (ADB)*, Vadstena, Sweden, Lecture Notes in Computer Sciences, vol. 819, Springs, New York, 1994.

[16] M. Manouvrier et al., "Quadtree representations for storage and manipulation of clusters of images", *Image and Vision Computing* 20 (2002) 513-527.