

# A Comparative Study on ROI-Based Lossy Compression Techniques for Compressing Medical Images

V. Radha, *Member, IAENG*

**Abstract**—Medical image compression is an important area of research which aims at producing algorithms that reduce file size and at the same time maintain relevant diagnostic information. The main focus of this paper is to analyze techniques and find a compression scheme that can compress medical images quickly and reduce compression rate while maintaining a good level of visual quality. A Region Of Interest (ROI)-based approach is used to separate the important medical data and background data. The Block Truncation Coding (BTC) and Discrete Cosine Transformation (DCT) algorithms are used to code background region, while Embedded Zero-tree Wavelet (EZW) coding, Set Partition In Hierarchical Tree (SPIHT) Algorithm, Zero-Tree Entropy (ZTE) Coding Algorithm and Singular Value Decomposition (SVD) are used to compress the ROI region. Several experiments were conducted to analyze the algorithms based on compression ratio, decompressed image quality and speed.

**Index terms**— Medical image compression, Discrete Cosine Transformation, Wavelet-based coding, Singular Value Decomposition Region Of Interest.

## I. INTRODUCTION

Medical imaging is the use of the algorithms and procedures for operations such as general enhancement, compression, analysis and mapping on images obtained from medical devices. Examples of such devices include x-ray devices, CT / MRI scanners, electron microscope, etc. Medical image compression is considered as an important task in the field of medical image processing and has gained more attention as hospitals and clinics move towards filmless imaging and go completely digital. According to [9], the motivation behind using compression for medical images comes from the amount of storage required to store raw image produced by the medical imaging devices. A typical 12-bit medical X-ray may be 2048 pixels by 2560 pixels in dimension. This translates to a file size of 10,485,760 bytes.

Manuscript received July 16, 2011; revised August 12, 2011.

V. Radha is with the Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women University, Coimbatore, Tamilnadu, PIN 641043 India. (phone: +91422-2404983; e-mail:radhasrimail@gmail.com).

A typical 12-bit medical X-ray may be 2048 pixels by 2560 pixels in dimension. This translates to a file size of 10,485,760 bytes. A typical 16-bit mammogram image may be 4500 pixels by 4500 pixels in dimension for a file size of 40,500,000 (40 megabytes). This has consequences for disk storage and image transmission time. Even though disk storage has been increasing steadily, the volume of digital imagery produced by hospitals and their new filmless radiology departments has been increasing even faster. Even if there were infinite storage, there is still the problem of transmitting the images.

According to ESG White Paper of ESG Strategy Group, the database archiving and storage volumes is going to increase by more than 11 per cent each year [8], which stresses the need to rethink the approaches available to accommodate future storage and information management. Compression is a technology that is proposed as a solution to such situations. Moreover, according to [2], the computer processing power doubles every 18 months and the disk storage capacity doubles every nine months. This trend shows that the ability to capture data has far outpaced the ability to process and store it. If the amount of information in the world doubles every 18 months, then the techniques needed to store and transmit these captured data are to be improved accordingly. Thus, the demand for compression techniques grows tremendously and is considered to be very important in medical environment.

Compression techniques in medical domain work with dual goals of reducing the file size and at the same time maintain relevant diagnostic information. Reduced file size has a direct impact on transmission speed, which plays a vital role in fields like telemedicine. Several researchers have proposed lossy compression algorithms for the natural images which contain large low frequency components, few components of high frequency and higher derivatives. In the domain of medical images, compression research is still at its peak. The reason behind this is that medical imaging devices are constantly changing and many new innovative techniques are being used in producing medical images, which help the healthcare professionals in better diagnosis.

The existing solutions focus either on developing a new technique or enhance the existing techniques. The medical community has been reluctant to adopt lossless methods for image compression. The main goal has been to produce an exact replica of the original image, suffering high file size. Only recently, attention to use lossy image compression which maximizes compression while maintaining clinical relevance data has been probed ([3], [5]).

This demand for fast and efficient coding algorithms in medical field has led to the development of several techniques which have revolutionized the area of image compression. This increase in the number of techniques has given rise to the dilemma of deciding which of these methods possess the best properties and potentials for effective coding. This problem is of particular importance in the medical field where the distortion of information may lead to inaccurate diagnosis. Thus, there exists a need to compare the different coding algorithms in order to see the advantages of each technique. This paper, in continuation with this thought, analyzes the suitability of lossy techniques for medical image compression. The research tries to answer the question “Which of the existing lossy techniques works better for medical images in terms of compression rate and quality?” and answering this question is the main goal of this work. The aim is to analyze techniques and find a compression scheme that can compress medical images quickly and at the same time reduce compression rate while maintaining a good level of visual quality. The various algorithms considered are Block Truncation Coding (BTC), Discrete Cosine Transformation (DCT), Embedded Zero-tree Wavelet (EZW) coding, Set Partition In Hierarchical Tree (SPIHT) Algorithm, Zero-Tree Entropy (ZTE) Coding Algorithm and Singular Value Decomposition (SVD).

To increase the efficiency of compression algorithm, two approaches have been used. The first technique uses coding images based on feature extraction. In these techniques, the image pixels are classified as significant and non-significant and the coding concentrates more on significant regions. The second method is Region of Interest (ROI) coding. The main idea behind these schemes is to preserve medically important regions while the rest of the regions are highly compressed. The ROI schemes reduce transmission time and storage cost to a significant extent and therefore are mostly preferred. In this paper, the BTC and DCT algorithms are used to code background region, while the rest of the algorithms are used for ROI region. The rest of the paper is organized as below. Section II a brief discussion on the selected algorithms. Section III presents the general methodology used. The experimental results and discussion is presented in Section IV, while Section V concludes the work with future research direction.

## II. COMPRESSION TECHNIQUES

The various compression techniques considered in this paper are discussed in this section.

### A. BTC Algorithm

Block Truncation Coding (BTC) is a lossy moment preserving quantization method for compressing digital gray-level images. Its advantages are simplicity, fault tolerance, the relatively high compression efficiency and good image quality of the decoded image. The BTC

algorithm is a lossy fixed length compression method that uses a Q level quantizer to quantize a local region of the image. The quantizer levels are chosen such that a number of the moments of a local region in the image are preserved in the quantized output. In its simplest form, the objective of BTC is to preserve the sample mean and sample standard deviation of a grayscale image. Additional constraints can be added to preserve higher order moments. For this reason BTC is called as a block adaptive moment preserving quantizer. The principle used by the block truncation coding (BTC) method and its variants is to quantize pixels in an image while preserving the first two or three statistical moments. The algorithm begins by dividing an image into blocks (4×4 or 8×8 pixels). Assuming that a block contains n pixels with intensities  $p_1$  through  $p_n$ , the first two moments are the mean and variance, can be calculated using Equations (1) and (2), from which the standard deviation of the block can be calculated (Equation 3).

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i \quad (1)$$

$$\overline{p^2} = \frac{1}{n} \sum_{i=1}^n p_i^2 \quad (2)$$

$$\sigma = \sqrt{\overline{p^2} - \bar{p}^2} \quad (3)$$

The principle of the quantization is to select three values, a threshold  $p_{thr}$ , a high value  $p^+$ , and a low value  $p^-$ . Each pixel is replaced by either  $p^+$  or  $p^-$ , such that the first two moments of the new pixels (i.e., their mean and variance) will be identical to the original moments of the pixel block. The rule of quantization is that a pixel  $p_i$  is quantized to  $p^+$  if it is greater than the threshold, and is quantized to  $p^-$  if it is less than the threshold (if  $p_i$  equals the threshold, it can be quantized to either value). Thus,

$$p_i \leftarrow \begin{cases} p^+ & \text{if } p_i \geq p_{thr} \\ p^- & \text{if } p_i < p_{thr} \end{cases} \quad (4)$$

Intuitively, it is clear that the mean  $\bar{p}$  is a good choice for the threshold. The high and low values can be determined by writing equations that preserve the first two moments, and solving them. The number of pixels in the current block that are greater than or equal to the threshold is denoted by  $n^+$ . Similarly,  $n^-$  stands for the number of pixels that are smaller than the threshold. The sum  $n^+ + n^-$  equal the number of pixels  $n$  in the block. Once the mean  $\bar{p}$  has been computed, both  $n^+$  and  $n^-$  are easy to calculate. Preserving the first two moments is expressed by the two equations

$$\bar{p} n = n^- \bar{p}^- + n^+ \bar{p}^+, \quad \overline{p^2} n = n^- (\bar{p}^-)^2 + n^+ (\bar{p}^+)^2 \dots \dots \quad (5)$$

and the solutions are

$$\bar{p}^- = \bar{p} - \sigma \sqrt{\frac{n^+}{n^-}}, \quad \bar{p}^+ = \bar{p} + \sigma \sqrt{\frac{n^-}{n^+}} \quad (6)$$

As these solutions are generally real numbers, they are rounded to the nearest integer, which implies that the mean and variance of the quantized block may be somewhat different from those of the original block. The solutions located on the two sides of the mean  $\bar{p}$  at distances are proportional to the standard deviation  $\sigma$  of the pixel block.

**B. DCT Algorithm**

The Discrete Cosine Transform (DCT) is a mathematical transformation technique that is used to convert a spatial representation of data into a frequency representation. A data in the frequency domain contains the same information as that in the spatial domain. The order of values obtained by applying the DCT is coincidentally from lowest to highest frequency. This feature and the psychological observation that the human eye and ear are less sensitive to recognizing the higher-order frequencies leads to the possibility of compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-order ones. When reconstructing the data and transforming it back to the spatial domain, the results are remarkably similar to the original signal. The DCT method can be used to compress both color and gray scale images. DCT is a method is most frequently used in several areas including WWW, industries, etc. and this popularity has made the author choose DCT as a format to be analyzed and compared. The step by step procedure is given below.

1. The image is divided into 8 x 8 blocks of pixels.
2. Working from left to right, top to bottom, apply DCT to each block.
3. Compress each block through a process called quantization
4. The resulting array of blocks that constitute the image is highly compressed and occupy very small amount of space.
5. When desired, the image can be reconstructed through Inverse Discrete Cosine Transform (IDCT), which is a reverse process of compression.

**C. EZW Algorithm**

This algorithm laid the foundation of modern wavelet coders and provides excellent performance for the compression of still images as compared to block based DCT algorithm. Introduced by Shapiro in 1993 [7], this algorithm uses the multi-resolution properties of wavelet transform. The EZW algorithm first uses DWT for the decomposition of an image where at each level  $i$ , the lowest spatial frequency subband is split into 4 more subbands for next higher level  $i+1$ , i.e.,  $LL_{i+1}$ ,  $LH_{i+1}$ ,  $HL_{i+1}$  and  $HH_{i+1}$  and then decimated. The algorithm uses the idea of significance map as an indication of whether a particular coefficient is zero or nonzero (i.e., significant) relative to a given quantization level. This means that if a wavelet coefficient at a coarse scale or highest level is insignificant (quantized to zero) with respect to a given threshold  $T$ , then all wavelet coefficients of the same orientation at the same spatial location at next finer scales (i.e., lower level) are likely to be zero with respect to  $T$ . The coefficient at coarse scale is called parent while the coefficients at the next fine scales in the same spatial orientation are called children (Figure 1).

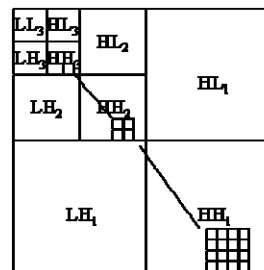


Figure 1. Parent-child dependencies of subbands in EZW

One can use this principle and code such a parent as a zero-tree root (ztr), thereby avoiding coding all of its children. This gives considerable compression as compared to block based coding algorithms such as DCT.

EZW scans wavelet coefficients subband by subband in a zigzag manner. Parents are scanned before any of their children by first scanning all neighboring parents. Each coefficient is compared against the current threshold  $T$ . A coefficient is significant if its amplitude is greater than  $T$ ; such a coefficient is then encoded using one of the symbols negative significant (ns) or positive significant (ps). The zero-tree root (ztr) symbol is used to signify a coefficient below  $T$ , with all its children in the zero-tree data structure also below  $T$ . The isolated zero (iz) symbol signifies a coefficient below  $T$ , but with one of its child not below  $T$ . For significant coefficients, EZW further encodes coefficient values using successive approximation quantization (SAQ) scheme. Coding is done bit-plane by bit-plane. The successive approximation approach to quantization of the wavelet coefficients leads to the embedded nature of EZW coded bit-stream. Finally the coefficients in the bit-stream are coded losslessly using adaptive arithmetic coding.

**D. SPIHT Algorithm**

In 1996, Pearlman and Said [6] improved the embedded zerotree wavelet (EZW) algorithm and developed a faster and more efficient image coding technology called set partitioning in hierarchical trees (SPIHT). SPIHT represents a step toward realizing lower costs with respect to compression complexity and prediction, as proposed in JPEG and JPEG 2000, to achieve higher compression performances [1]. In the SPIHT algorithm, the image is first decomposed into a number of subbands by means of hierarchical wavelet decomposition. For example, the subbands obtained for two-level decomposition are shown in Figure 2.

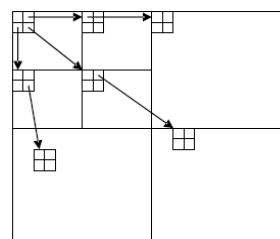


Figure 2. Two-Level Wavelet Decomposition

The subband coefficients are then grouped into sets known as spatial-orientation trees, which efficiently exploit the correlation between the frequency bands. The coefficients in each spatial orientation tree are then progressively coded from the most significant bit-planes (MSB) to the least significant bit-planes (LSB), starting with the coefficients with the highest magnitude and at the lowest pyramid levels. The SPIHT multistage encoding process employs three lists and a set:

1. The list of insignificant pixels (LIP) contains individual coefficients that have magnitudes smaller than the threshold.
2. The list of insignificant sets (LIS) contains sets of wavelet coefficients that are defined by tree structures and are found to have magnitudes smaller than the threshold (insignificant). The sets exclude the coefficients corresponding to the tree and all subtree roots and they have at least four elements.
3. The list of significant pixels (LSP) is a list of pixels found to have magnitudes larger than the threshold (significant).
4. The set of offspring (direct descendants) of a tree node,  $O(i, j)$ , in the tree structures is defined by pixel location  $(i, j)$ . The set of descendants,  $D(i, j)$ , of a node is defined by pixel location  $(i, j)$ .  $L(i, j)$  is defined as  $L(i, j) = D(i, j) - O(i, j)$ .

The threshold,  $T$ , for the first bit-plane is equal to  $2n$ , and  $n = \lfloor \log_2(\max_{(i,j)} \{ |c_{(i,j)}| \}) \rfloor$  (7)

where  $c_{(i,j)}$  represents the  $(i, j)^{th}$  wavelet coefficient. All the wavelet coefficients are searched in order to obtain the maximum  $c_{(i,j)}$  after executing the discrete wavelet transform. For operations in the subsequent bit-planes of threshold  $T$ ,  $n$  is reduced by 1. For each pixel in the LIP, one bit is used to describe its significance. If it is not significant, the pixel remains in the LIP and no more bits are generated; otherwise, a sign bit is produced and the pixel is moved to the LSP. Similarly, each set in the LIS requires one bit for the significance information. The insignificant sets remain in the LIS; the significant sets are partitioned into subsets, which are processed in the same manner and at the same resolution until each significant subset has exactly one coefficient. Finally, each pixel in the LSP is refined with one bit. The above mentioned procedure is then repeated for the subsequent resolution.

*E.Zero-Tree Entropy (ZTE) Coding Algorithm*

ZTE coding is a new efficient technique for coding wavelet transform coefficients and is based on, but differs significantly from, the EZW algorithm [4]. Like EZW, this new ZTE algorithm exploits the self-similarity inherent in the wavelet transform of images and video residuals to predict the location of information across wavelet scales. ZTE coding organizes quantized wavelet coefficients into wavelet trees and then uses zero-trees to reduce the number of bits required to represent those trees. ZTE differs from EZW in four major ways:

1. quantization is explicit instead of implicit and can be performed distinct from the zero-tree growing process or can be incorporated into the process, thereby making it possible to adjust the quantization according to where the transform coefficient lies and what it represents in the frame;
2. coefficient scanning, tree growing, and coding are done in one pass instead of bit-plane-by-bit-plane;

3. coefficient scanning is changed from subband by subband to a depth-first traversal of each tree; and
4. the alphabet of symbols for classifying the tree nodes is changed to one that performs significantly better for low bit-rate encoding of video. The ZTE algorithm does not produce an embedded bit-stream as EZW does, but by sacrificing the embedding property, this scheme gains flexibility and other advantages over EZW coding, including substantial improvement in coding efficiency.

In ZTE coding, the coefficients of each wavelet tree are reorganized to form a wavelet block as depicted in Figure 3. Each wavelet block comprises those coefficients at all scales and orientations that correspond to the frame at the spatial location of that block. The concept of the wavelet block provides an association between wavelet coefficients and what they represent spatially in the frame.

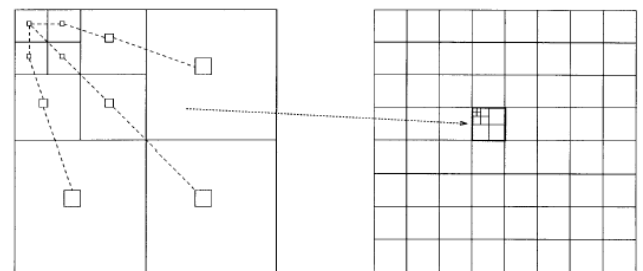


Figure 3. Reorganization of a wavelet tree into a wavelet block

EZW scans coefficients from subband by subband in a zigzag manner. In ZTE, all wavelet coefficients that represent a given spatial block are scanned, in ascending frequency order from parent to child, to grandchild, and so on, before the coefficients of the next adjacent spatial location are scanned.

*F.SVD*

Singular Value Decomposition (SVD) is considered to be one of the significant topics in linear algebra by many renowned mathematicians. SVD has many practical and theoretical values, other than image compression. One special feature of SVD is that it can be performed on any real  $(m,n)$  matrix. It factors  $A$  into three matrices  $U, S, V$ , such that,  $A = USV^T$ . Where  $U$  and  $V$  are orthogonal matrices and  $S$  is a diagonal matrix. In this research work, the values of SVD are used to perform medical image compression and the process is explained in this section. The main purpose of (SVD) is to factor a image matrix  $A$  into  $USV^T$ . The matrix  $U$  contains the left singular vectors, the matrix  $V$  contains the right singular vectors, and the diagonal matrix  $S$  contains the singular values. The singular values to attain this goal are arranged on the main diagonal as given in Equation (8)

$$\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > \sigma_{r+1} = \sigma_p = 0 \tag{8}$$

where  $r$  is the rank of matrix  $A$ , and where  $p$  is the smallest of the dimensions  $m$  or  $n$ . There are many properties and attributes of SVD, some of the properties used with medical image compression are listed below.

1. The singular value  $\sigma_1, \sigma_2, \dots, \sigma_n$ , are unique, however, the matrices  $U$  and  $V$  are not unique;
2. Since  $A^T A = V S^T S V^T$  and  $V$  diagonalizes  $A^T A$ , it follows that the  $v_j$ s are the eigenvector of  $A^T A$ .

3. Since  $AA^T = USS^T U^T$ , so it follows that  $U$  diagonalizes  $AA^T$  and that the  $u_i$ 's are the eigenvectors of  $AA^T$ .
4. If  $A$  has rank of  $r$  then  $v_j$  and  $v_j, \dots, v_r$  form an orthonormal basis for range space of  $A^T$ ,  $R(A^T)$ , and  $u_j$  and  $u_j, \dots, u_r$  form an orthonormal basis for range space  $A$ ,  $R(A)$ .
5. The rank of matrix  $A$  is equal to the number of its nonzero singular values.

According to the property 5 of SVD in the above section, "the rank of matrix  $A$  is equal to the number of its nonzero singular values". In many applications, the singular values of a matrix decrease quickly with increasing rank. This propriety allows to reduce the noise or compresses the matrix data by eliminating the small singular values or the higher ranks.

When an image is SVD transformed, it is not compressed, but the data take a form in which the first singular value has a great amount of the image information. With this, it is possible to use only a few singular values to represent the image with little differences from the original. To illustrate the SVD image compression process, the following detail procedures are given below.

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (9)$$

That is  $A$  can be represented by the outer product expansion:

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T \quad (10)$$

When compressing the image, the sum is not performed to the very last SVs and the SVs with small enough values are dropped, as they are ordered on the diagonal fashion. The closet matrix of rank  $k$  is obtained by truncating those sums after the first  $k$  terms:

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_k u_k v_k^T \quad (11)$$

The total storage for  $k$   $A$  will be  $k(m + n + 1)$ . The integer  $k$  can be chosen confidently less then  $n$  and the digital image corresponding to  $kA$  will still be very close the original image. However, the selection of different  $k$  values will produce different corresponding image and with different memory usage. For typical choices of the  $k$ , the storage required for  $kA$  will be less the 20 percentage.

### III. METHODOLOGY

In the proposed methodology, the input medical image is first segmented into two regions. The first is the ROI, which has the area of interest of the medical practitioner and the second is the background, which has less important data. In the present work, a background / foreground separation algorithm is used to identify ROI and background regions. A Harris Corner algorithm (Figure ---) is used for this purpose. After segmenting the image into foreground and background, the BTC and DCT algorithms are used to compress the background, while the EZE, SPIHT, ZTE and SVD algorithms are used on the foreground. Thus, eight models, as listed below are designed and used for compressing medical images.

1. BTC - background and EZE - foreground(BTC-EZE)
2. BTC - background and SPIHT - foreground(BTC-SPIHT)
3. BTC - background and ZTE - foreground(BTC-ZTE)
4. BTC - background and SVD - foreground(BTC-SVD)

5. DCT - background and EZE - foreground(DCT-EZE)
6. DCT - background and SPIHT - foreground(DCT-SPIHT)
7. DCT - background and ZTE - foreground(DCT-ZTE)
8. DCT - background and SVD - foreground(DCT-SVD)

### IV. EXPERIMENTAL RESULTS

The proposed system was vigorously tested with test images to analyze its performance on compressing medical images. The results obtained are discussed in this section. The images used for this research were 512x512, 8 bits per pixel (bpp) images (Figure 4). All the experiments were conducted using Pentium IV machine with 512MB RAM. The system is evaluated using the performance parameters, like, Compression Ratio, Peak Signal to Noise Ratio (PSNR), Compression Time and Decompression Time.

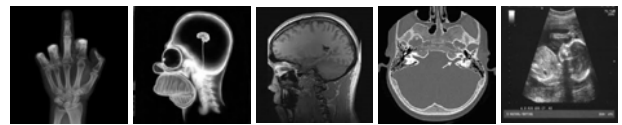


Figure 4. Test Images

#### A..Compression Ratio

The average compression ratio obtained during experimentation for the test images is shown in Figure 5.

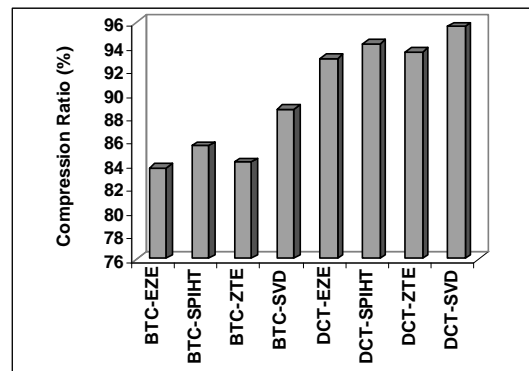


Figure 5. Average Compression Ratio

From the results, it is evident that the SVD-based models (BTC-SVD and DCT-SVD) produce better results when compared with all other models. While comparing between BTC and DCT combined with SVD, the DCT model achieved better compression ratio and achieved a 7.29 compression gain. This was followed by the SPIHT models, with SPIHT-DCT showing 9.21% efficiency gain in compression over SPHIT-BTC. The lowest performance was shown by EZE-based algorithms.

#### B. Peak Signal to Noise Ratio (PSNR)

The average PSNR values obtained are shown in Figure 6.

Again, with respect to decompressed image quality, the SVD-based algorithm showed significant increase in image quality than other models. This was followed by the SPIHT-based algorithms, SPIHT-based algorithms. As with compression ratio, the performance of the EZE-based algorithms, showed low quality decompressed image when compared with all other models. While comparing between BTC and DCT combination, the DCT showed better performance in terms of PSNR than BTC algorithm.



### C. Compression and Decompression Time

The average total compression time is shown in Figure 7. The total compression time is calculated as the sum of compression and decompression time taken by the algorithms.

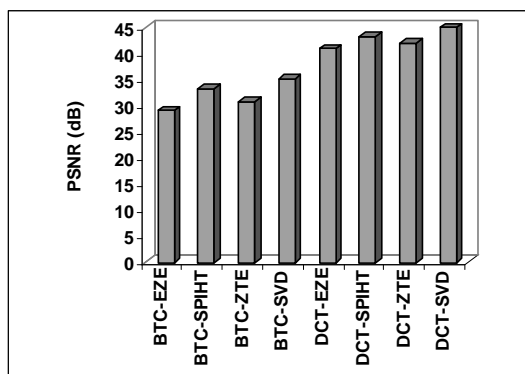


Figure 6. Average PSNR

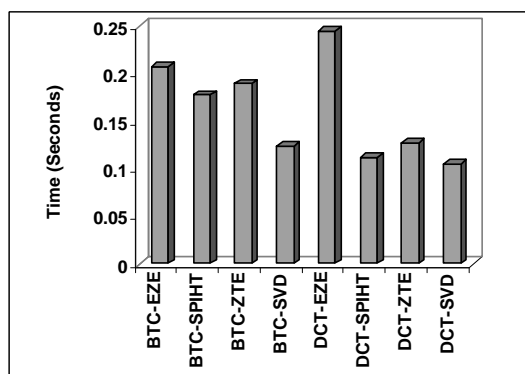


Figure 7. Speed of Algorithm

From the figure, it is clear that the DCT-SVD is the fastest algorithm, followed by DCT-SPIHT and BTC-ZTE algorithms. The EZE-based algorithms were the slowest of all the eight algorithms.

### V. CONCLUSION

The application of BTC, DCT, EZE, SPIHT, ZTE and SVD coding algorithms for medical images was analyzed and compared in this paper. The compression process began with identifying the ROI and background information of the medical image. BTC and DCT coding was applied to compress the background data, while SVD, ZTE, SPIHT and EZE algorithms were used to compress the more important ROI Data. Various experiments conducted showed that the coding approach that integrates SVD with DCT produce better results and shows significant improvement in terms of compression ratio, PSNR and speed. This was followed by SPIHT-based approaches. The EZE-based algorithms seem to be poor choice for compressing medical images, especially when combined with BTC algorithm. There are many possible directions for future investigation. In order to obtain better compression rates, the ROI algorithm can be improved, to take into account the edges of an image also, which has to be losslessly coded. Methods to handle noisy images can be considered and analyzed. A clinical case study with radiologists to observe the effect of compression on diagnostic performance can be conducted.

### REFERENCES

- [1] Adams, M.D. (2001) The JPEG-2000 Still Image Compression Standard, ISO/IEC JTC1/SC29/W0G1 (ITU-T SG8).
- [2] Fayyad, U. and Uthurusamy, R. (2002) Evolving data into mining solutions for insights, Communications of ACM, Vol. 45, Issue 8, Pp. 28-31.
- [3] Karras, D.A., Karkanis, S.A. and Maroulis, D.E. (2000) Efficient Image Compression of Medical Images Using the Wavelet Transform and Fuzzy c-means Clustering on Regions of Interest, Proceedings of the 26th EuroMicro Conference, Vol. 2, Pp.2469-2469.
- [4] Matrucci, S.A. and Sodagar, I. (1997) A zerotree wavelet video coder, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 7, No.1, Pp. 109-118.
- [5] Palanisamy, G. and Samukutti, A. (2008) Medical image compression using a novel embedded set partitioning significant and zero block coding, The International Arab Journal of Information Technology, Vol. 5, No. 2, Pp. 132-139.
- [6] Said, A. and Pearlman, W. A. (1996) A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, IEEE Trans. CSVT, Vol. 6, No. 3, Pp. 243-250, [ftp://ipl.rpi.edu/pub/EW\\_Code/SPIHT.ps.gz](ftp://ipl.rpi.edu/pub/EW_Code/SPIHT.ps.gz).
- [7] Shapiro, J.M. (1993) Embedded image coding using zerotrees of wavelet coefficients, IEEE Transactions on Signal Processing, Vol. 41, No. 12, Pp. 3445-3462.
- [8] Turner, M.J. (2008) IBM Information Infrastructure Initiative Aims to Tame the Information Explosion, White Paper, Enterprise Strategy Group, Pp.1-10.
- [9] Zukoski, M.J. (2006) A novel approach to medical image compression, Int. J. Bioinformatics Research and Applications, Vol. 2, No. 1, pp.89-103.