

Panoramic Dental X-Ray Image Compression using Wavelet Filters

J.Nancy¹, J.Wilson², K.Vinoth Kumar³

PG Scholar, P.S.N.A College of Engineering and Technology, Tamil Nadu, India¹

Assistant Professor, M.A.R College of Engineering and Technology, Tamil Nadu, India^{2,3}

Abstract—Recently, wavelet transform has proven to be very effective in medical image compression. So, the choice of wavelet filters is a vital factor that could determine the compression performance. This paper considered the effect of applying different types of wavelet filters belonging to orthogonal and biorthogonal wavelet families with different orders on the panoramic dental x-ray images. In addition to commonly use of PSNR and compression ratio measurement is adopted for objective image quality evaluation. SPIHT coding is used to encode and decode the image. The wavelet filters used are Haar, Daubichies8, 9/7 and 5/3 filters. The performance evaluation of the image quality is measured objectively using peak signal to noise ratio and compression ratio. The simulated results reveal that 9/7 irreversible wavelet transform is almost the best choice for lossy panoramic dental x-ray image compression. And it can be described by using a MATLAB software which is a high performance language for technical computing. It integrates computation, visualization, and programming in an easy to use environment where problems and solutions expressed in familiar mathematical notation.

Keywords— Medical image compression, panoramic dental x-ray images, wavelet filters, PSNR, Compression ratio.

1. Introduction

According to the constrained bandwidth and storage capacity, medical image have to be compressed before storage and transmission. Storing digital medical image is standardized by the DICOM (Digital Imaging and Communications in Medicine). It is based on JPEG (Joint Photographic Experts Group) compression algorithm, which used the discrete cosine transform for the main feature [1]. However, this transform is unacceptable due to the poor resolution of blocking artefacts and low compression ratio that is sometimes inadequate for bandwidth limited transmission.

At present, it has been known that the lossy compression is acceptable if the compression will not affect the diagnosis accuracy of the digital medical images. Recently, DICOM has approved the lossy medical image compression by a JPEG2000 baseline system where the wavelet transform has been used. In wavelet domain, the

image is represented by a set of basis functions. If the basis functions are different, the image displayed will also be different and the coding performance will change. In fact there is no single wavelet, which will always provide the best performance.

In this project image compression is done by using the discrete wavelet transformation (DWT). Since there are many wavelet filters available, each with the different set of basis functions, the choice of wavelet filters is very crucial factor to gain a good coding performance. Therefore, the main purpose of this thesis is to investigate the effect of applying different types of wavelet filters belonging to orthogonal and biorthogonal wavelets with different orders.

The wavelet filters used are Haar, Daubichies8, 9/7 and 5/3 filters. The compression results are done on four panoramic dental x-ray images. The size of images is 256x256, 8 bits per pixels. The performance evaluation of the image quality is measured using peak signal to noise ratio and compression ratio.

2. Related Works

In the past decades, the discrete cosine transform (DCT) has been the most popular for compression because it provides optimal performance and is implemented at a reasonable cost. Several compression algorithms, such as the JPEG standard [2] for still images and the Moving Picture Experts Group (MPEG) standard [3] for video images are based on DCT.

Embedded Zero tree Wavelet (EZW) [4], Set Partitioned Embedded block coder (SPECK) [5], Wavelet Difference Reduction (WDR)[6], Space Frequency Quantization (SFQ) [7], Compression with Reversible Embedded Wavelet (CREW) [8], Embedded Predictive Wavelet Image Coder (EPWIC) [9], Embedded Block Coding with Optimized Truncation (EBCOT) [10], and Stack- Run (SR) [11] standard are based on the discrete wavelet transform (DWT) [12]. DWT has the ability to solve the blocking effect introduced by DCT, it also reduces the correlation between the neighbouring pixels and gives multi scale sparse representation of the image. For instance, in medical image compression applications, diagnosis is effective only when

compression techniques preserve all the relevant and important image information needed. This is the case with lossless compression techniques. Lossy compression techniques, on the other hand, are more efficient in terms of storage and transmission needs but there is no warranty that it preserves the characteristics needed in medical image processing and diagnosis. In this latter case, of lossy compression, image characteristics are usually preserved in the coefficients of the domain space in which the original image are transformed.

That is, for instance, in the DWT based medical image compression, the wavelet coefficients keep all the information needed for reconstructing the medical image. The goal of such a compression methodology that aims at maximization of the compression ratio should be to discard only their relevant wavelet coefficients according to a criterion, like the magnitude of their values. This is done either by applying the same constant threshold to the whole transform domain or by applying different thresholds to uniquely defined regions of the transform domain, depending on the significance of these regions in the preservation of image characteristics.

Embedded Zero Tree Wavelet

The EZTW [4] algorithm is one of the first algorithms to show the full power of wavelet-based image compression. It is introduced in the groundbreaking paper of Shapiro. An EZW encoder is an encoder specially designed to use with wavelet transforms. The EZW encoder is originally designed to operate on images (2D-signals) but it is also used on other dimensional signals. The EZW encoder is based on progressive encoding to compress an image into a bit stream with increasing accuracy. This means that when more bits are added to the stream, the decoded image will contain more detail. Progressive encoding is also known as embedded encoding, which explains the E in EZW.

The EZW encoder is based on two important observations¹. Natural images in general have a low pass spectrum. When an image is wavelet transformed the energy in the sub bands decreases as the scale decreases (low scale means high resolution), so the wavelet coefficients will, on average, be smaller in the higher sub bands than in the lower sub bands. This shows that progressive encoding is a very natural choice for compressing wavelet transformed images, since the higher sub bands only add detail. 2. Large wavelet coefficients are more important than small wavelet coefficients.

These two observations are exploited by encoding the wavelet coefficients in decreasing order, in several passes. For every pass a threshold is chosen against which all the wavelet coefficients are measured. If a wavelet coefficient is larger than the threshold it is encoded and removed from the image, if it is smaller it is left for the next pass. When all the wavelet coefficients have been visited the threshold is lowered and the image is scanned again to add more detail to the already encoded image. This process is repeated until all the wavelet

coefficients have been encoded completely or another criterion has been satisfied (maximum bit rate for instance).

Set Partitioned Embedded Block Coder

The SPEBC [5], is different from some of the above-mentioned schemes in that it does not use trees which span, and exploit the similarity, across different sub bands; rather, it makes use of sets in the form of blocks. The main idea is to exploit the clustering of energy in frequency and space in hierarchical structures of transformed images. The SPECK algorithm is said to belong to the class of scalar quantized significance testing schemes. It has its roots primarily in the ideas developed in the SPIHT, and few block coding image coding algorithms. An image which has been adequately transformed using an appropriate sub band transformation. The transformed image is said to exhibit a hierarchical pyramidal structure defined by the levels of decomposition, with the topmost level being the root.

The finest pixels lie at the bottom level of the pyramid while the coarsest pixels lie at the top (root) level. The SPECK algorithm makes use of rectangular regions of image. These regions are hence forth referred to as sets, is of varying dimensions. The dimension of a set depends on the dimension of the original image and the sub band level of the pyramidal structure at which the set lies. During the course of the algorithm, sets of various sizes will be formed, depending on the characteristics of pixels in the original set. A set of size 1 consists of just one pixel. The other type of sets used in the SPECK algorithm is referred to as sets of type 1. These sets are obtained by chopping off a small square region from the top left portion of a larger square region. The former contains sets of type of varying sizes which have not yet been found significant against a threshold while the latter obviously contains those pixels which have tested significant. Two types of set partitioning are used in SPECK. They are quad tree partitioning and octave band partitioning.

The motivation for quad tree partitioning of sets is to zoom in quickly to areas of high energy in the set and code them first. The idea behind octave band partitioning scheme is to exploit the hierarchical pyramidal structure of the sub band de-composition, where it is more likely that energy is concentrated at the top most levels of the pyramid and as one goes down the pyramid, the energy content decreases gradually.

Embedded Block Coding With Optimized Truncation

The EBCOT [6] algorithm uses a wavelet transform to generate the sub band coefficients which are then quantized and coded. Although the usual dyadic wavelet decomposition is typical, other "packet" decompositions are also supported and occasionally preferable. The original image is represented in terms of a collection of sub bands, which may be organized into increasing resolution levels. The lowest resolution level consists of the single LL sub band. Each successive resolution level contains the additional sub bands, which are required to

reconstruct the image with twice the horizontal and vertical resolution.

The EBCOT algorithm is related in various degrees to much earlier work on scalable image compression. A key of scalable compression is that the target bit-rate or reconstruction resolution needs not be known at the time of compression. Another advantage of practical significance is that the image need not be compressed multiple times in order to achieve a target bit-rate, as is common with the existing JPEG compression standard. EBCOT partitions each sub band into relatively small blocks of samples and generates a separate highly scalable bit-stream to represent each so called code-block. The algorithm exhibits state-of-the-art compression performance while producing a bit-stream with an unprecedented feature set, including resolution and SNR scalability together with a random access property.

The algorithm has modest complexity and is extremely well suited to applications involving remote browsing of large compressed images. The EBCOT bit-stream is composed of a collection of quality layers and that scalability is obtained by discarding unwanted layers. As might be expected, performance decreases as more layers are added to the bit-stream, because the overhead associated with identifying the contributions of each code-block to each layer grows.

Wavelet Difference Reduction (WDR)

This makes it difficult to perform operations which depend on the position of significant transform values, such as region selection on compressed data. Region selection, also known as region of interest (ROI) which means a portion of a compressed image that requires increased resolution. This will occur, for, example, with a portion of a low resolution medical image that has been sent at a low bpp rate in order to arrive quickly. Such compressed data operations are possible with the WDR [7] algorithm. The term difference reduction refers to the way in which WDR encodes the locations of significant wavelet transform values. Although WDR will not produce higher PSNR, as observed, it will produce perceptually superior images, especially at high compression rates. The only difference between WDR and bit plane encoding is the significant pass. In WDR, the output from the significance pass consists of the signs of significant values along with sequences of bits which concisely describe the precise locations of significant values.

Space Frequency Quantization (SFQ)

SFQ [8] for Wavelet Image Coding belongs to a new class of image coding algorithms coupling standard scalar quantization of frequency coefficients with tree structured quantization. Its good performance appears to confirm the promised efficiencies of hierarchical representation. This technique exploits both spatial and frequency compaction property of the wavelet transform through the use of two simple quantization modes. To exploit the spatial compaction property, a symbol is defined, that indicates that a spatial

region of high frequency coefficients has zero value. Application of this symbol is referred to as zero-tree quantization, because it will involve setting to zero a tree-structured set of wavelet coefficients.

This is done in the first phase called Tree Pruning Algorithm. In the next phase called Predicting the tree, the relation between a spatial region in image and the tree-structured set of coefficients is exploited. Zero tree quantization is viewed as a mechanism for pointing to the location where high frequency coefficients are clustered. Thus, this quantization mode directly exploits the spatial clustering of high frequency coefficients predicted. For coefficients that are not set to zero by zero tree quantization, a common uniform scalar quantization, independent of coefficient frequency band is applied. Uniform quantization followed by entropy coding provides nearly optimal coding efficiency.

Compression with Reversible Embedded Wavelet (CREW)

CREW [9] is a new form of still image compression developed at the Ricoh California Research Centre in Menlo Park, California, is a new type of image compression system and is lossy and lossless, bi-level and continuous-tone, progressive by resolution and pixel depth, and will preserve the source image at encode and quantize for the target device at decode or transmission. It uses a new form of wavelet transform technology. It is pyramidal (similar to hierarchical) and progressive by nature. CREW was the stimulus for a new JPEG 2000 standard. CREW offers a number of features that should be expected of the compression standards of the next century.

The features make CREW an ideal choice for applications that require high quality and flexibility for multiple input and output environments, such as, medical imagery, fixed-rate and fixed-size applications pre-press images, continuous-tone facsimile documents, image archival, World Wide Web (WWW) image or graphic type, satellite images. Many of these applications have never used compression because the quality will not be assured, the compression rate is not high enough, or the data rate is not controllable. Three new technologies combine to make CREW possible when the reversible wavelet transform on-linear filters that have exact reconstruction implemented in minimal integer arithmetic.

The embedded code stream of a method of implying quantization in the code stream and a high-speed, high-compression binary entropy coder the same CREW code stream is used for both lossless and lossy applications due to embedded quantization. The wavelet transform produces pyramid ally ordered data and a natural means for interpolation. The bit-significance coding allows for bit plane progressive transmission.

Embedded Predictive Wavelet Image Coder (EPWIC)

EPWIC [10] is an embedded image coder based on a statistical characterization of natural images in the wavelet

transform domain. The joint distribution between pairs of coefficients at adjacent spatial locations, orientations, and scales are defined. Although the raw coefficients are nearly uncorrelated, their magnitudes are highly correlated. A linear magnitude predictor coupled with both multiplicative and additive uncertainties, provides a reasonable description of the conditional probability densities. In EPWIC, sub band coefficients are encoded one bitplane at a time using a non-adaptive arithmetic encoder. Bit-planes are ordered using an algorithm that considers the MSE reduction per encoded bit. The overall ordering of bit planes is determined by the ratio of their encoded variance to compressed size. The coder is inherently embedded, and should prove useful in applications requiring progressive transmission.

Stack- Run (SR)

SR [11] image coding is a new approach in which a 4-ary arithmetic coder is used to represent significant coefficient values and the length of zero runs between coefficients. This algorithm works by raster scanning within sub bands and therefore involves much lower addressing complexity than other algorithms such as zero tree coding which requires creation and maintenance of lists of dependencies across different decomposition levels. Despite its simplicity and the fact that these dependencies are not explicitly used, this algorithm is competitive with best enhancements of zero tree coding.

The stack-run representation of partitions the quantized transform coefficients into two groups containing zero valued and nonzero valued, referred to as significant coefficients. Each significant coefficient is represented in binary notation as a stack or a column of bits with the MSB at the top and the LSB at the bottom. A sub band is encoded by starting in one corner of the sub band and performing a raster scan described in terms of stack and run representations, where 'stack' is the magnitude and sign of the significant coefficient and 'run' is the number of zero valued coefficients encountered before the next significant coefficient.

3. Proposed System

The architecture of the proposed system is shown in fig3.1. The size of images is 256x256, 8 bits per pixels. Five types of wavelet families (Haar, Daubichies8, 9/7 and 5/3, Integer wavelet) are used to investigate the performance of image compression. The images are initially transformed by wavelet transform at 4 levels. The wavelet transform coefficients then sorted in decreasing order of values in order to determine a threshold. Next, wavelet coefficients below the threshold level are set to zeros where the threshold is determined by the remaining rate in percents. Then, image is restored by the renewed wavelet coefficients. Finally, the inverse wavelet transform is performed to get the compressed image results. Important properties of wavelet filters in digital image compression are compact support (lead to efficient

Implementation), symmetry (useful in avoid ingarti facts at the borders of the wavelet sub bands and lead to linear phase filter), orthogonality (allow fast algorithm), regularity, and degree of smoothness (related to filter order or filter lengths).

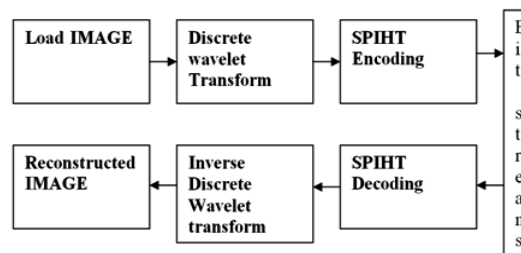


Fig 3.1 Proposed systems Architecture

The different wavelet filters make different trade-off between how compactly the basis functions are localized in space and how smooth they are [Grgic, 2001]. However, regularity alone is not sufficient criterion to determine the quality of image compression neither [Villas nor, 1995]. Moreover, the even length filters have significantly less shift variance than odd length filters. In this thesis, four types of wavelet families are used to investigate the performance of image compression. The orthogonal wavelet [Daubichies, 1998] that used in this study is Haar wavelet and Daubichie8. In biorthogonal wavelet transform, the 9/7 and 5/3 filters are used in order to obtain the linear phase wavelet filters.

Wavelet transform provides a compact multi-resolution representation of the image [Mallet, 1998]. It has excellent energy compaction property which is suitable for exploiting redundancy in an image to achieve compression. DWT is implemented using two-channel wavelet filter bank in a recursive fashion. For an image, 2D-DWT is calculated using a separable approach. Input image is first scanned in the horizontal direction and passed through low pass and high pass decomposition filters producing low-frequency and high-frequency data in horizontal direction.

3.3.1 Wavelet Filters Used For Compression

In two ways the computation of the discrete wavelet transform has been implemented. The first is convolution with appropriate boundary handling; the second is a fast lifting approach, which is used in this project, a refined system of very short filters which are applied in a way that produces the same result as the first approach. Lifting scheme is derived from a poly phase matrix representation of the wavelet filters, a representation that is distinguishing between even and odd samples. The wavelet filters that used in this project are given below

- a) Haar wavelet
- b) Daubichies8
- c) Daubichies9/7
- d) Daubichies5/3
- e) Integer wavelet

Daubichies9/7

Forward transform

$$F_{high}(j,k+1) = \text{odd}(j,k) - \text{even}(j,k+1) \quad (3.1)$$

$$F_{Low}(j,k) = \text{even}(j,k+1) + \text{round}[f_{high}(j,k)/2] \quad (3.2)$$

Inverse transform

$$\text{Even}(j,k+1) = \text{low}(j,k) - \text{round}[f_{high}(j,k)/2] \quad (3.3)$$

$$\text{Odd}(j,k) = f_{high}(j,k) + \text{even}(j,k+1) \quad (3.4)$$

Daubichies5/3

Forward transform

$$\begin{aligned} lo(i,j) = & (2*x(i,2*j-1) + 6*x(i,2*j) + 2*x(i,2*j+1) - \\ & x(i,2*j+2))/4; \end{aligned} \quad (3.5)$$

$$hi(i,j) = (2*x(i,2*k+1) - x(i,2*k+2))/2; \quad (3.6)$$

Inverse transform

$$x(i,2*j-1) = (6*w(i,2*j-1) + 2*w(i,2*j) - w(i,2*j+1) + 4)/8; \quad (3.7)$$

$$x(i,2*j) = (-w(i,2*j-1) + 2*w(i,2*j) - w(i,2*j+1))/4; \quad (3.8)$$

Integer Wavelet

Forward transform

$$\text{pix_diff} = (\text{im}(i,j+1) - \text{im}(i,j)); \quad (3.9)$$

$$\text{pix_sum} = (\text{im}(i,j) - [\text{pix_diff}/2]); \quad (3.10)$$

Inverse transform

$$s1 = (\text{pix_sum} - [\text{pix_diff}/2]); \quad (3.11)$$

$$s2 = (\text{pix_diff} + s1); \quad (3.12)$$

To calculate the transform of an array of pixels

1. Find the average of each pair of samples.
2. Find the difference between the average and the samples.
3. Fill the first half of the array with averages.
4. Fill the second half of the array with differences.
5. Repeat the process on the first half of the array.

Set Partitioning In Hierarchical Trees (SPIHT) Coding

The SPIHT coder is a highly refined version of the EZW algorithm and is a powerful image compression algorithm that produces an embedded bit stream from the best reconstructed images using mean square error at various bit rates. Some of the best results of highest PSNR values for given compression ratios for a wide variety of images have been obtained with SPIHT. Hence, it has become the benchmark state-of-the-art algorithm for image compression. The SPIHT algorithm [Amir, 1996] uses a partitioning of the spatial orientation trees in a manner that tends to keep insignificant coefficients together in larger subsets. The

partitioning decisions are binary decisions that are transmitted to the decoder, providing a significance map encoding. The thresholds used for checking significance are powers of two. So in essence the SPIHT algorithm sends the binary representation of the integer value of the wavelet coefficients.

The significance map encoding or set partitioning and ordering step is followed by a refinement step in which the representations of the significant coefficients are refined. In SPIHT algorithm, the wavelet coefficients [Assad, 1999] are divided into trees originating from the lowest resolution band. This algorithm is applied to both grey-scale and colour images. SPIHT displays exceptional characteristics over several properties like good image quality, fast coding and decoding, a fully progressive bit stream, application in lossless compression, error protection and ability to code for exact bit rate.

The SPIHT process represents a very effective form of coding. A straightforward consequence of the compression simplicity is the greater coding/decoding speed. The SPIHT algorithm is nearly symmetric, i.e., the time to encode is nearly equal to the time to decode. SPIHT codes the individual bits of the image wavelet transform coefficients following a bit plane sequence. It is capable of recovering the image perfectly by decoding all these bits. In practice it is possible to recover the image perfectly using rounding after recovery. SPIHT generates two types of data. The first is sorting information, which needs error protection. The second consists of uncompressed sign and refinement bits, which do not need special protection because they affect only one pixel. Coding Algorithm

Since the order in which the subsets are tested for significance is important, in a practical implementation the significance information is stored in three ordered lists, called list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP). In all lists each entry is identified by a coordinate (i, j), which in the LIP and LSP represents individual pixels, and in the LIS represents either the set D(i, j) or L(i, j). To differentiate between them, we say that a LIS entry is of type A if it represents D(i, j), and of type B if it represents L(i, j).

During the sorting pass (see Algorithm I), the pixels in the LIP-which were insignificant in the previous pass-are tested, and those that become significant are moved to the LSP. Similarly, sets are sequentially evaluated following the LIS order, and when a set is found to be significant it is removed from the list and partitioned. The new subsets with more than one element are added back to the LIS, while the single-coordinate sets are added to the end of the LIP or the LSP, depending whether they are insignificant or significant, respectively. The LSP contains the coordinates of the pixels that are visited in the refinement pass. Below we present the new encoding algorithm in its entirety. It is essentially equal to Algorithm I, but uses the set partitioning approach in its sorting pass.

Algorithm II:

1) Initialization: output $n = \lfloor \log_2(\max_{(i,j)} \{|c_{i,j}|\}) \rfloor$; set the LSP as an empty list, and add the coordinates (i, j) $\in H_{to}$

the LIP, and only those with descendants also to the LIS, as type Entries.

2) Sorting Pass:

2.1) for each entry (i,j) in the LIP do:

2.1.1) output $S_n(i, j)$;

2.1.2) if $S_n(i, j) = 1$ then move (i, j) to the LSP and Output the sign of $c_{i,j}$;

2.2) for each entry (i, j) in the LIS do:

2.2.1) if the entry is of type A then Output $S_n(D(i,j))$;if $S_n(D(z, 1)) = 1$ then for each $(k, I) \in O(i,j)$ do:output $S_n(k, I)$; if $S_n(k, I) = 1$ then add (k, I) to the LSB and sign of $c_{i,j}$ if $S_n(k, I) = 0$ then add (k, I) to the end of LIP.if $l(i, j) \neq 0$ then move (i, j) to the end of the LIS, as an entry of type B, and go to Step 2.2.2); otherwise, remove entry (i, j) from the LIS;

2.2.2) if the entry is of type B then output $S_n(L(2, J))$; if $S_n(L(I, j)) = 1$ then add each $(k,I) \in O(z, j)$ to the end of the LIS as an entry of type A; remove (i, j) from the LIS.

3) Refinement Pass: for each entry (i, j) in the LSP, except those included in the last sorting pass (i.e., with same n), output the nth most significant bit of $|c_{i,j}|$;

4) Quantization-Step Update: decrement n by 1 and go to Step 2.

One important characteristic of the algorithm is that the entries are added to the end of the LIS in Step 2.2) are evaluated before the same sorting pass ends. So, when it says "for each entry in the LIS" it also means those that are being added to its end. With Algorithm II, the rate is precisely controlled because the transmitted information is formed of single bits. The encoder will also use the property, to estimate the progressive distortion reduction and stop at a desired distortion value.

Note that in Algorithm II, all branching conditions based on the significance data S_n which will be calculated with the knowledge of $C_{i,j}$ and the output given by the encoder. Thus, to obtain the desired decoder's algorithm, which duplicates the encoder's execution path as it sorts the significant coefficients, it simply has to replace the words output by input in Algorithm II. Comparing the algorithm above to Algorithm I, it will show that the ordering information $\eta(k)$ recovered when the coordinates of the significant coefficients are added to the end of the LSP; that is, the coefficients pointed by the coordinates in the LSP are sorted as in (5).

But note that whenever the decoder inputs data, its three control lists (LIS, LIP, and LSP) are identical to the ones

used by the encoder at the moment it outputs that data, which means that the decoder indeed recovers the ordering from the execution path. It is easy to see that with this scheme, coding and decoding have the same computational complexity.

An additional task done by decoder is to update the reconstructed image. For the value of n when a coordinate is moved to the LSP, it is known that $2^n \leq |c_{i,j}| < 2^{n+1}$. So, the decoder uses that information, plus the sign bit that is input just after the insertion in the LSP, to set $\hat{c}_{i,j} = 1.5 \times 2^n$. Similarly, during the refinement pass, the decoder adds or subtracts 2^{n-1} to $\hat{c}_{i,j}$ when it inputs the bits of the binary representation of $|c_{i,j}|$. In this manner, the distortion gradually decreases during both the sorting and refinement passes.

As with any other coding method, the efficiency of Algorithm II is be improved by entropy-coding its output, but at the expense of a larger coding/ decoding time. Practical experiments have shown that normally there is little to be gained by entropy-coding the coefficient signs or the bits put out during the refinement pass. On the other hand, the significance values are not equally probable, and there is a statistical dependence between $S_n(i, j)$ and $S_n[D(i, j)]$ and also between the significance of adjacent pixels.

To increase the coding efficiency, groups of 2×2 coordinates were kept together in the lists, and their significance values were coded as a single symbol by the arithmetic coding algorithm. Since the decoder only needs to know the transition from insignificant to significant (the inverse is impossible), the amount of information that needs to be coded changes according to the number m of insignificant pixels in that group, and in each case it is conveyed by an entropy-coding alphabet with 2^m symbols. With arithmetic coding it is straight forward to use several adaptive models, each with 2^m symbols, $m \in \{1, 2, 3, 4\}$, to code the information in a group of four pixels.

By coding the significance information together, the average bit rate corresponds to an m^{th} order entropy. At the same time, by using different models for the different number of insignificant pixels, each adaptive model contains probabilities conditioned to the fact that a certain number of adjacent pixels are significant or insignificant. This way the dependence between magnitudes of adjacent pixels is fully exploited. The scheme above was also used to code the significance of trees rooted in groups of 2×2 pixels.

Image Quality

SPIHT wins in the test of finding the minimum rate required to obtain a reproduction indistinguishable from the original. The SPIHT advantage is even more pronounced in encoding color images, because the bits are allocated automatically for local optimality among the color

components, unlike other algorithms that encode the color components separately based on global statistics of the individual components. The lossless color compression is obtained with some images at compression ratios from 100-200:1.

Progressive Image Transmission

In some systems with progressive image transmission like WWW browsers the quality of the displayed images is inefficient. The problem is that such widely used schemes employ a very primitive progressive image transmission method. On the other extreme, SPIHT is a state-of-the-art method that is designed for optimal progressive transmission and still beats most non-progressive methods. It does so by producing a fully embedded coded file, at any moment the quality of the displayed image is the best available for the number of bits received up to that moment. So, SPIHT is very useful for applications where the user will quickly inspect the image and decide if it should be really downloaded, or is good enough to be saved, or need refinement.

Optimized Embedded Coding

A strict definition of the embedded coding scheme is if two files produced by the encoder have size M and N bits, with $M > N$, then the file with size N is identical to the first N bits of the file with size M. If it want to compress an image for three remote users means. Each one have different needs of image reproduction quality, and it find that those qualities are obtained with the image compressed to at least 8 Kbp, 30 Kbp, and 80 Kbp, respectively. It uses a non-embedded encoder like JPEG, to save in transmission costs (or time). On the other hand, if it use an embedded encoder like SPIHT then it will compress the image to a single 80 Kbp file, and then send the first 8 Kbp of the file to the first user, the first 30p Kb to the second user, and the whole file to the third user.

Surprisingly, with SPIHT all three users would get for the same file size an image quality comparable or superior to the most sophisticated non-embedded encoders available today. SPIHT achieves this feat by optimizing the embedded coding process and always coding the most important information first.

Lossless Compression

SPIHT codes the individual bits of the image wavelet transform coefficients following a bit-plane sequence. Thus, it is capable of recovering the image perfectly every single bit of it by coding all bits of the transform. However, the wavelet transform yields perfect reconstruction only if its numbers are stored as infinite-precision numbers. In practice it is frequently possible to recover the image perfectly using rounding after recovery, but this is not the most efficient approach. A code that uses this transformation to yield efficient

Progressive transmission up to lossless recovery is among the SPIHT. A surprising result obtained with this code is that for lossless compression it is as efficient as the most effective

lossless encoders lossless JPEG is definitely not among them. In other words, the property that SPIHT yields progressive transmission with practically no penalty in compression efficiency applies to lossless compression too.

Rate or Distortion Specification

Almost all image compression methods developed so far do not have precise rate control. For some methods you specify a target rate, and the program tries to give something that is not too far from what you wanted. For others it specify a "quality factor" and wait to see if the size of the file fits the need. The embedded property of SPIHT allows exact bit rate control, without any penalty in performance (no bits wasted with padding or whatever). The same property also allows exact mean squared-error (MSE) distortion control. Even though the MSE is not the best measure of image quality, it is far superior to other criteria used for quality specification.

Encoding/Decoding Speed

A straightforward consequence of the compression simplicity is the greater coding/decoding speed. The SPIHT algorithm is nearly symmetric, i.e., the time to encode is nearly equal to the time to decode. Complex compression algorithms tend to have encoding times much larger than the decoding times.

4. Results and Comparison

The results show that the best filter for lossless panoramic dental x-ray image compression wavelet transform and it is described by using MATLAB software which is a high performance language for technical computing. It integrates computation, visualization, and programming in an easy to use environment where problems and solutions expressed in familiar mathematical notation. The Quality of the reconstructed image is measured interns of mean square error (MSE), peak signal to noise ratio (PSNR) ratio and compression ratio [12]. The MSE is often called reconstruction error variance and it is defined as,

$$MSE = 1/MxN \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x(m,n) - x^*(m,n)] \quad (4.1)$$

Where $x(m, n)$ is the original $M \times N$ pixel image and $x^*(m,n)$ is the reconstructed image. Where the sum over j, k denotes the sum over all pixels in the image and N is the number of pixels in each image. From that the peak signal-to-noise ratio is defined as the ratio between signal variance and reconstruction error variance. The PSNR between two images having 8 bits per pixel in terms of decibels (dBs) [12] is given by:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (4.2)$$

Generally when PSNR is 40 dB or greater, then the original and the reconstructed images are virtually indistinguishable by human eyes.

The compression ratio gives an indication of how much compression is achieved for a particular image. Most algorithms have a typical range of compression ratios that they can achieve over a variety of images. Because of this, it is usually more useful to look at an average compression ratio for a particular method. The compression ratio (CR) is given by,

$$CR = \frac{\text{Size of the Original Image}}{\text{Size of compressed Image}} \quad (4.3)$$

Simulation Results

The simulated results reveal that 9/7 irreversible wavelet transform is almost the best choice for lossy panoramic dental x-ray image compression. And it can be described by using a MATLAB software which is a high performance language for technical computing.

Table4.1: The average peak signal to noise ratio and the average compression ratio measurement of a panoramic dental X-ray image1

Filters	PSNR	CR	Enc_time	Dec_time	MSE
Haar	54.4606	2.42	4.03686	1.50749	0.232
Db8	33.6443	2.54	3.28109	0.88742	28.09
9/7	26.8536	2.45	3.64064	1.46464	134.1
5/3	42.3471	2.18	3.51335	1.43949	3.787
IW	26.9219	1.23	4.20618	1.71613	132.1

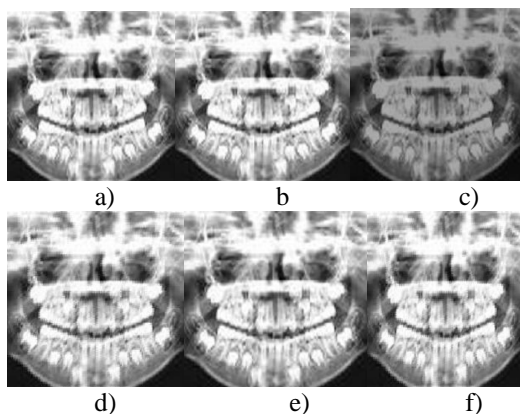


Figure 4.1 a) Original image b) Reconstructed Image 1 Using Haar filter c) Daubichies filter d) Daubichies 9/7 filter e) Daubichies 5/3 filter f) Integer Wavelet

Table4.2: The average peak signal to noise ratio and the average compression ratio measurement of a panoramic dental X-ray image 2

Filters	PSNR	CR	Enc_time	Dec_time	MSE
Haar	55.0643	1.34	3.22634	1.17898	0.202
Db8	33.3867	1.38	3.39299	0.73936	29.81
9/7	29.5224	2.70	3.20762	1.41176	72.58
5/3	41.8973	1.17	3.66379	1.31651	4.200
IW	29.6012	1.36	3.52453	1.29255	71.27

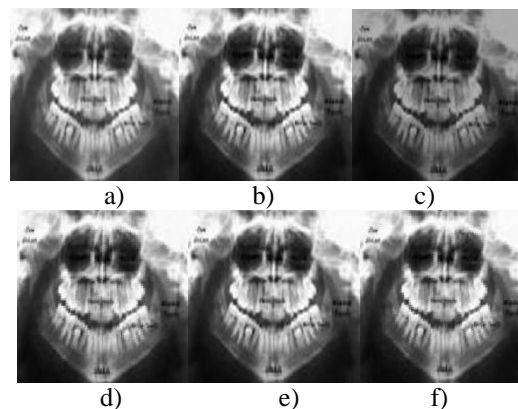


Figure 4.2 a) Original image b) Reconstructed Image 1 Using Haar filter c) Daubichies filter d) Daubichies 9/7 filter e) Daubichies 5/3 filter f) Integer Wavelet.

Table4.3: The average peak signal to noise ratio and the average compression ratio measurement of a panoramic dental X-ray image3

Filters	PSNR	CR	Enc_time	Enc_time	MSE
Haar	55.0502	1.33	3.21301	1.10009	0.203
Db8	28.0055	1.20	3.47936	1.15984	102.9
9/7	27.2258	2.67	3.47043	1.23941	123.1
5/3	37.3022	1.10	3.3388	1.32233	12.10
IW	27.2609	1.35	3.68595	1.29464	122.1

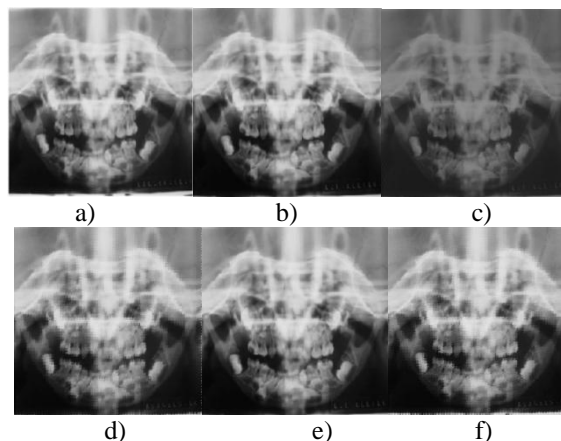


Figure 4.3 a) Original image b) Reconstructed Image 3 Using Haar filter c) Daubichies filter d) Daubichies 9/7 filter e) Daubichies 5/3 filter f) Integer Wavelet

Table4.4: The average peak signal to noise ratio and the average compression ratio measurement of a panoramic dental X-ray image 4

Filters	PSNR	CR	Enc_time	Enc_time	MSE
Haar	54.2384	1.15	4.26494	1.66357	0.245
Db8	34.7083	1.04	3.80531	1.65159	21.99
9/7	27.9406	2.34	3.58715	1.77584	104.4
5/3	44.2247	0.99	3.85134	1.78208	2.458
IW	28.0117	1.17	3.95266	1.48453	102.7

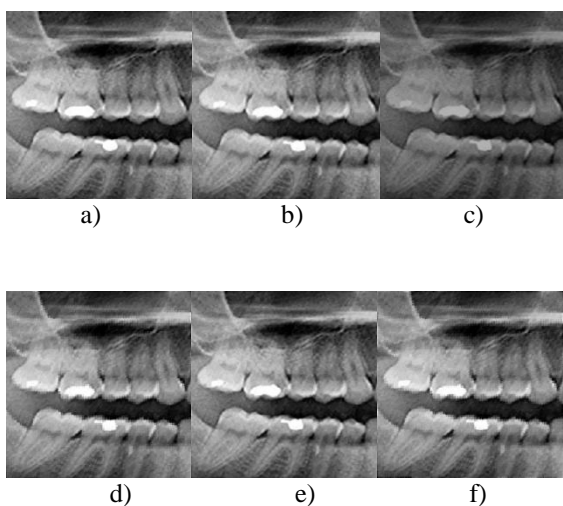


Figure 4.4 a) Original image b) Reconstructed Image4 Using Haar filter c) Daubichies filter d) Daubichies 9/7 filter e) Daubichies 5/3 filter f) Integer Wavelet

5. Conclusion

This paper considered the effect of applying different types of wavelet filters belonging to orthogonal and biorthogonal wavelet families with different orders on the panoramic dental x-ray images. In addition to commonly use of PSNR and compression ratio measurement is adopted for objective image quality evaluation. SPIHT coding is used to encode and decode the image. From the experimental results, it can be seen that PSNR alone is not suitable objective scale to evaluate compressed images. Also, the best choice of wavelet filters depends on the contents of image to be compressed. The results show that the best filter for lossless panoramic dental x-ray image compression is 9/7 irreversible wavelet transform. And it is described by using MATLAB software which is a high performance language for technical computing. It integrates computation, visualization, and

programming in an easy to use environment where problems and solutions expressed in familiar mathematical notation. The proposed algorithm will be an efficient and fast algorithm for image compression. It will be useful to the applications in electronic equipments such as Digital Multimedia Broadcasting (DMB), electronic dictionary, navigation, and etc.

References

- [1] K. Hammonton, "Ultrasonic Echo Image Compression by JPEG2000 - optimization and the subjective assessment of the quality," Engineering in Medicine and Biology Society, Proceedings of 25th Annual International Conference of the IEEE, vol.1, 2003.
- [2] Wallace, G.K., The JPEG still picture compression standard, Comm. of the ACM, 34(4), 30 44, 1999.
- [3] Rafa_ Mantiuk, Alexander Efremov, Karol Myszkowski, "Backward Compatible High Dynamic Range MPEG Video Compression" c ACM, 2006.
- [4] Jerome M Shapiro Embedded Image coding using zerotrees of wavelets coefficients", IEEE Transactions on signal processing, "Vol 41 No. 12 , pp. 3445-3462, Dec 1993.
- [5] Asad Islam & Pearlman, "An embedded and efficient low-complexity, hierarchical image coder", Visual Communication and Image processing '99 proceedings of SPIE., Vol 3653, pp294-305, Jan 1999
- [6] Walker, J.S., Alossy image codec based on adaptively scanned wavelet difference reduction, Optical Engineering, 2005.
- [7] Vladan Velisavljević, Baltasar Beferull-Lozano "Space-Frequency Quantization for Image Compression with Direction lets" IEEE Transactions on Image Processing March 15, 2007.
- [8] Boliek, M., Gormish, M. J., Schwartz, E. L., and Keith, A. Next Generation Image Compression and Manipulation Using CREW, Proc. IEEE ICIP, 1997, <http://www.crc.ricoh.com/CREW>.
- [9] Buccigrossi, R., and Simoncelli, E. P. EPWIC: Embedded Predictive Wavelet Image Coder, GRASPLaboratory, TR#414, 2008
- [10] Tsai, M. J., Villasenor, J. D., and Chen, F. Stack- Run Image Coding, IEEE Trans. Circuit and systems for video technology, vol. 6, no. 5, Oct. 1996, pp.519- 521.
- [11] Vetterli, M. and Kovacevic, J. Wavelets and Subband Coding, Englewood Cliffs, NJ, Prentice Hall, 1995, <http://cm.bell-labs.com/who/jelena/Book/home.html>
- [12] N. S. Jayant and P. Nool, Digital Coding of Waveforms: Principles and Applications to Speech and Video, Printice-Hall, 1994.
- [13] D. S. Taubman, M. Marcellin and M. W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2001.
- [14] Mallet, S. G. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. PAMI, vol. 11, no. 7, pp. 674-693. , July 1998.

- [15] Rafael C.Gonzalez and Richard E. Woods “Digital Image Processing”, Second edition.
- [16] M. Grgic, M. Ravnjak, and B. Zovko-Cihlar, “Filter comparison in wavelet transform of still images,” Proceedings of IEEE International Symposium Industrial Electronics, 1999.
- [17] S. Grgic, M. Grgic, and B. Zovko-Cihlar, “Performance Analysis of Image Compression Using Wavelets,” IEEE Transactions on Industrial Electronics, vol.48, no. 3, 2001.
- [18] J. D. Villasenor, B. Belzer and J. Liao, “Wavelet Filter Evaluation for Image Compression,” IEEE Transactions on Image Processing, vol.4, no. 8, pp. 1053-1060, 1995..
- [19] I. Daubechies, “Orthogonal bases of compactly supported wavelets,” Common, On Pure and Applied Mathematics, vol.12, pp. 909-996. Sept, 1998.
- [20] Amir Said, and Pearlman, W. A. A New, Fast, and Efficient Image Codec Based on SPIHT, IEEE Trans. Circuit and systems for Video Technology, vol. 6, no. 3, pp. 243-250, June 2008.
- [21] Assad Islam & Pearlman, “An embedded and efficient low-complexity, hierarchical image coder”, Visual Communication and Image processing’99 proceedings of SPIE., Vol 3653, pp294-305, Jan 1999.

He received the **M.E.** degree in Applied Electronics from the J.J College of engineering and technology, Trichirappalli, India, in 2011. Currently doing **Ph.D.** in Communication and Networking in Karpagam University Coimbatore. He published six international journals and his research interest includes wireless communication, Mobile Ad hoc networks, Sensor Networks, Communication networks.

Biographies



J.Nancy received the **B.E.** degree in Electronics and Communication Engineering from the P.S.N.A College of engineering and technology, Dindigul, Anna University, Trichirappalli, India, in 2012. Currently doing **M.E.** VLSI Design in P.S.N.A College of engineering and technology, Dindigul, India. His research interest includes Low Power VLSI, CMOS Technology, and Image Processing.



J.Wilson received the **B.E.** degree in Electronics and Communication Engineering from the Kurinji College of engineering and technology, Manapparai, Anna University, Chennai, India, in 2008. He received the **M.E.** degree in Applied Electronics from the P.S.N.A College of engineering and technology, Dindigul, India, in 2010. Currently doing **Ph.D.** in Communication and Networking in Karpagam University Coimbatore. He published two international journals and his research interest includes Mobile Ad hoc networks Low Power VLSI, CMOS Technology, Image Processing.



K.Vinoth Kumar received the **B.E.** degree in Electronics and Communication Engineering from the Kurinji College of engineering and technology, Manapparai, Anna University, Chennai, India, in 2009.