

Image Compression using Space Adaptive Lifting Scheme

¹Ramu Satyabama and ²Annadurai

¹Department of Electronics, Government College of Technology,
Coimbatore, 641 013, India

²Department of Technical Education, Chennai, 600025, India

Abstract: Problem statement: Digital images play an important role both in daily life applications as well as in areas of research and technology. Due to the increasing traffic caused by multimedia information and digitized form of representation of images; image compression has become a necessity. **Approach:** Wavelet transform has demonstrated excellent image compression performance. New algorithms based on Lifting style implementation of wavelet transforms have been presented in this study. Adaptively is introduced in lifting by choosing the prediction operator based on the local properties of the image. The prediction filters are chosen based on the edge detection and the relative local variance. In regions where the image is locally smooth, we use higher order predictors and near edges we reduce the order and thus the length of the predictor. **Results:** We have applied the adaptive prediction algorithms to test images. The original image is transformed using adaptive lifting based wavelet transform and it is compressed using Set Partitioning In Hierarchical Tree algorithm (SPIHT) and the performance is compared with the popular 9/7 wavelet transform. The performance metric Peak Signal to Noise Ratio (PSNR) for the reconstructed image is computed. **Conclusion:** The proposed adaptive algorithms give better performance than 9/7 wavelet, the most popular wavelet transforms. Lifting allows us to incorporate adaptivity and nonlinear operators into the transform. The proposed methods efficiently represent the edges and appear promising for image compression. The proposed adaptive methods reduce edge artifacts and ringing and give improved PSNR for edge dominated images.

Key words: Adaptive lifting, prediction filters, edge detection, nonlinear wavelet transform, relative local variance, Cohen-Daubechies-Feauveau (CDF) wavelets

INTRODUCTION

An efficient way to implement Discrete Wavelet Transform (DWT) using filters was developed by (Mallat 1989). This very practical filtering algorithm, which is based on the theory of multiresolution analysis, yields a fast discrete wavelet transform. This original work focused on orthonormal systems where one set of basic functions was used for both analysis and synthesis. Due to its many advantages, such as multiresolution representation, good energy compaction and decorrelation, the DWT has become one of the most important techniques for image and video compression in the last decade and been adopted by Joint Photographic Experts Group (JPEG2000) standard (Taubman and Marcellin, 2002). The wavelet based JPEG2000 not only presents superior coding performance over the DCT (Discrete Cosine Transform) based JPEG but also provides scalabilities in rate, quality and resolution (Pennebaker and Mitchell, 1993; Ghrare *et al.*, 2009). Cohen *et al.*

(1992) developed the idea of biorthogonal wavelets where the analysis basis and the synthesis basis are different (Cohen *et al.*, 1992). The advantage of this approach is increased flexibility in wavelet design. For example, it is possible to design the associated filter bank with linear filters.

Sweldens (1996) demonstrated a simple lifting scheme which provided a common framework for the design of biorthogonal filter banks (Daubechies and Sweldens, 1996; Sweldens, 1996). The lifting scheme is a way of generating a new set of biorthogonal filters from a known biorthogonal set. In addition to the extra flexibility offered by biorthogonal wavelets, it transpires that all biorthogonal filters can be generated using lifting schemes. Cohen-Daubechies-Feauveau wavelets are the historically first family of biorthogonal wavelets, which was made popular by Ingrid Daubechies. These are not the same as the orthogonal Daubechies wavelets and also not very similar in shape and properties. However their construction idea is the same.

Corresponding Author: R. Satyabama, Department of Electronics, Government College of Technology, Coimbatore, 641 013, India

The JPEG 2000 compression standard uses the biorthogonal CDF 5/3 wavelet (also called the LeGall 5/3 wavelet) for lossless compression and a CDF 9/7 wavelet for lossy compression (Cohen *et al.*, 1992).

Conventionally, two dimensional DWT (2-D DWT) is carried out as a separable transform by cascading two one Dimensional (1-D) transforms in the horizontal and vertical direction. The wavelet transform can be efficiently implemented by the lifting scheme where the Finite Impulse Response (FIR) wavelet filter can be factored into several lifting stages (Daubechies and Sweldens, 1996; Sweldens, 1996).

A lifting stage is comprised of the four steps namely Split, Predict, Update and Normalize. The lifting scheme proposed by Sweldens (Daubechies and Sweldens, 1993; Sweldens, 1996) is an efficient tool for constructing second generation wavelets and has advantages such as faster implementation, fully in-place calculation, perfect reconstruction with low memory and low computational complexity (Daubechies and Sweldens, 1993). It can also be considered as an alternate implementation of the first generation classical wavelet transform.

In many applications it is desirable to have a filter bank that somehow determines how to shape itself according to the data that it analyzes. This can be achieved by allowing lifting scheme to adapt its update and prediction filters to the local properties of the signal. In this study adaptivity is introduced by choosing the prediction operator based on the local properties of the image.

MATERIALS AND METHODS

Lifting scheme: Each 1-D wavelet transform can be factored in to one or more lifting stages (Asamwar *et al.*, 2010). A typical lifting stage is comprised of four steps: Split, Predict, Update and Normalize.

Split: The signal $x[n]$ is first split into even subset $x_e[n]$ and the odd subset $x_o[n]$, where Eq. 1 and 2:

$$x_e[n] = x[2n] \tag{1}$$

and:

$$x_o[n] = x[2n + 1] \tag{2}$$

Predict: Then the odd subset $x_o[n]$ is predicted from the neighboring even subset $x_e[n]$. The Predictor $P(\cdot)$ is a linear combination of the neighboring even subset Eq. 3:

$$p(x_e)[n] = \sum_i p_i x_e[n + i] \tag{3}$$

where, p_i is the prediction filter coefficient which is a high pass filter. This leads to the detail coefficient Eq. 4:

$$d[n] = x_o[n] - p(x_e)[n] \tag{4}$$

If the signal is locally smooth, the prediction residual $d[n]$ will be small. Given the even subset $x_e[n]$ and the prediction residual $d[n]$, the odd subset $x_o[n]$ can be recovered by noting that Eq. 5:

$$x_o[n] = d[n] + p(x_e)[n] \tag{5}$$

Update: The Update step transforms the even subset $x_e[n]$ into a low-pass filtered version of $x[n]$. This coarse approximation is obtained by updating with a linear combination of the prediction residual $d[n]$. Then the approximation coefficients $c[n]$ are Eq. 6:

$$c[n] = x_e[n] + U(d)[n] \tag{6}$$

where, $U(\cdot)$ is a linear combination of neighboring d values given by Eq. 7:

$$U(d)[n] = \sum_j u_j d[n + j] \tag{7}$$

where, u_j is the low pass filter coefficient. The lifting construction guarantees perfect reconstruction for any Predict and Update filters. Given $d[n]$ and $c[n]$, we have Eq. 8:

$$x_e[n] = c[n] - U(d)[n] \tag{8}$$

Normalize: The outputs of the lifting are weighted by k_e and k_o . These values serve to normalize the energy of the underlying scaling and wavelet functions, respectively. The normalization factor k_e and k_o are $\sqrt{2}$ and $1/\sqrt{2}$ respectively (Daubechies and Sweldens, 1993). For 2D signals, upon the completion of the 1-D lifting based horizontal transform, the 1-D lifting based vertical transform is performed in the same way. The Forward and Inverse lifting is carried out as shown in Fig. 1. The four steps in Inverse lifting are:

- Undo Normalize
- Undo Update
- Undo Predict
- Merge

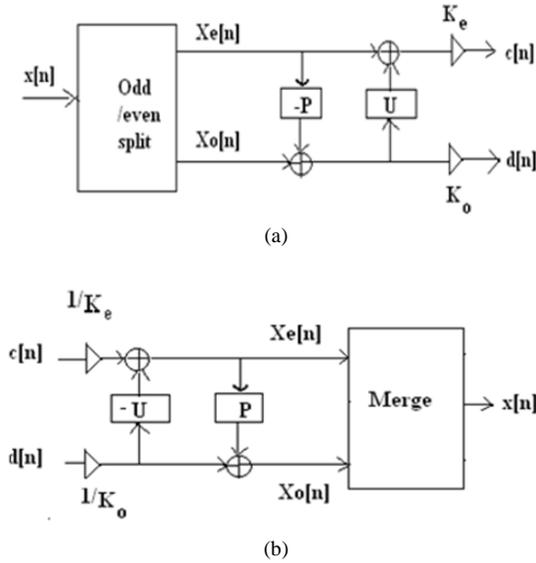


Fig. 1: The Lifting Scheme. (a) Forward Transform (b) Inverse Transform

The lifting framework allows us to incorporate non-linearities while retaining control over the properties of the wavelet transform. The nonlinearity comes from adaptively choosing from a set of linear predictors. Such nonlinear wavelet transforms provide added flexibility for image representations.

Adaptivity in wavelet transforms: Wavelet bases typically employed for image compression utilize smooth scaling and wavelet functions. Such bases can be easily constructed with the predict-then-update form of lifting as described above. Larger predictors that can exactly predict polynomials of higher degree correspond to smoother basis functions; these lifting predictors work well when the underlying signal is smooth. However, most of the images consist of regions of smoothness and texture separated by discontinuities (edges). These discontinuities cannot be well-represented by smooth basis functions. Since smooth basis functions correspond to lifting predictors with wide support, these predictors work poorly near edges, when the discontinuity is within the data that are used for prediction.

We introduce a mechanism that allows us to choose the prediction operator based on the local properties of the image. This makes the P operator data-dependent and thus the transform is nonlinear. However, lifting guarantees that the transform remains reversible. In regions where the image is locally smooth, we use higher order predictors. Near edges we reduce the order and thus the length of the predictor.

Such an adaptation would allow us to exploit the spatial structure that exists in edges. In this study, prediction filters are chosen based on the edge detection and the Relative Local Variance.

Adaptive prediction based on edge detection

(Method1): An edge detection algorithm analyzes the data in the 2-D prediction window to determine the location and the orientation of the edge. When an edge pixel is detected then we use a lower order predictor.

In this study an edge detection algorithm using Sobel operator is considered (Gonzalez *et al.*, 2004). The Sobel operator performs a 2-D spatial gradient measurement on an image. Typically it is used to find the approximate gradient magnitude at each point in an input grayscale image. The classical operator such as Sobel, which uses first derivative has very simple calculation to detect the edges and their orientations. It is easy to implement than the other operators. Sobel operator effectively highlights noise found in real world pictures as edges though the detected edges could be thick. Hence, Sobel operator is highly recommended in massive data communication found in image data transfer (Hafiz *et al.*, 2011).

The Sobel edge detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction and the other estimating the gradient in the y-direction. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The actual Sobel masks are shown Eq. 9 and 10:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (9)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (10)$$

At each point in the image the resulting gradient approximations can be combined to give the gradient magnitude using $G = \sqrt{G_x^2 + G_y^2}$ and using this information we can also calculate the gradient direction:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Thus we define a point in an image as an edge point if its two dimensional first order derivative is greater than the specified threshold.

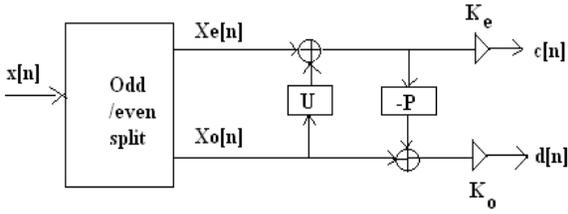


Fig. 2 Update First Lifting Scheme

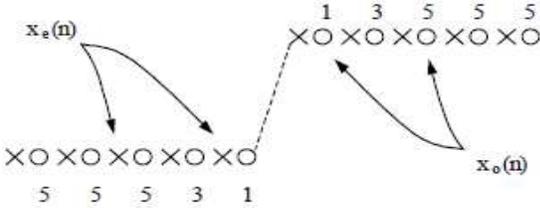


Fig. 3: Predictor Selection Near Edges. Number indicates the order of the predictor used

In lossy compression the decoder has only the quantized even coefficients rather than the original coefficients. If we use locally adapted filters, then quantization errors in coarse scales could cascade across scale and cause a series of incorrect filter choices leading to serious reconstruction errors. The simple modification that solves this problem is to reverse the order of the predict and update lifting steps in the wavelet transform as shown in Fig. 2 (Claypoole *et al.*, 2003). We first update the even samples based on the odd samples yielding the lowpass coefficients $c[n]$. We then reuse these lowpass coefficients to predict the odd samples, which gives the highpass coefficients $d(n)$. When updating first, the prediction operator is outside the loop. The coarse coefficients can be iterated to the lowest scale, quantized and reconstructed prior to the predictions. We use a linear update filter and let only the choice of predictor depend on the data.

When we do update first, the transform is only iterated on low pass coefficients and all $c(n)$ depend on data and are not affected by nonlinear prediction. Here we considered CDF(1,N) wavelets (Cohen-Daubechies-Feauveau) for adaptive lifting (Uytterhoeven *et al.*, 1997). The low pass coefficients are first computed using a Haar filter (one point update filter). We choose higher order predictors where the image is locally smooth, resulting in many negligible detail coefficients and near edges, lower order predictors are activated, resulting in large lifting detail coefficients for better image representation. Thus based on the gradient the prediction filters are chosen.

We choose $N=\{1,3,5\}$ point prediction. The prediction filters are represented as Eq. 10-13:

$$P(1) = [0, 0, 1, 0, 0] \text{ for order } N = 1 \quad (11)$$

$$P(2) = [0, -1, 8, 1, 0]/8 \text{ for order } N = 3 \quad (12)$$

$$P(3) = [-3, 22, 128, -22, 3] \text{ for order } N = 5 \quad (13)$$

and the update filter is Eq. 14:

$$U = [1, 1]/2 \quad (14)$$

The low pass coefficients are first computed using a Haar filter (one point update filter), where Eq. 15:

$$c(n) = \frac{x(n) + x(2n+1)}{2} \quad (15)$$

First order Haar prediction leading to (1,1) wavelet gives Eq. 16:

$$d(n) = x(2n+1) - c(n) \quad (16)$$

The third order predictor leading to (1,3) wavelet gives Eq. 17:

$$d(n) = x(2n+1) - \left(\frac{-c(n-1)}{8} + c(n) + \frac{c(n+1)}{8} \right)$$

The Fifth order predictor leading to (1,5) wavelet gives Eq. 18:

$$d(n) = x(2n+1) - \left(\frac{-3c(n-2)}{128} + \frac{22c(n-1)}{128} + c(n) - \frac{22(n+1)}{128} - \frac{3c(n+2)}{128} \right) \quad (18)$$

Figure 3 shows the predictor selection.

Adaptive prediction based on Relative Local Variance (Method 2):

The smoothness of the image can also be determined by measuring the Relative Local Variance (RLV). The Relative Local Variance of an image I is given by Eq. 19 and 20:

$$RLV[I](i,j) = \frac{\sum_{k=i-T}^{i+T} \sum_{l=j-T}^{j+T} (I(k,I) - \overline{\mu_{1,j}})^2}{\text{var}(I)}$$

With:

$$\overline{\mu_{i,j}} = \sum_{k=i-T}^{i+T} \sum_{l=j-T}^{j+T} \frac{I(k,l)}{(2T+1)^2} \quad (20)$$

For the window size we take T=5, since with this choice all I (k, l) used for the prediction of I (i, j) contribute to the RLV for (i, j). var(I) is the variance of the image I. For all pixels (i, j) to be predicted, we first compute RLV[I] (i, j). Then quantizing the values of the Relative Local Variance yields a decision map indicating which prediction filters should be used at different positions. The Relative Local Variance for all subsequent pixels (i, j) to be predicted is computed and suitable predictors are chosen. Two thresholds are chosen preliminarily according to practical situations. Quantization levels can be taken as multiples of the mean (μ) of the rlv. Test results have shown that [μ (rlv) 1.5 μ (rlv) 2 μ (rlv)] are the quantization levels that yield a good performance. The RLV value above the bigger threshold indicates that a lower order predictor namely P₁ should be selected. When RLV value is below the smaller threshold, a higher order predictor, namely P₃ should be activated. Otherwise P₂ is activated. Here also, we first update the even samples based on the odd samples. Then we reuse the low pass coefficients to predict odd samples which gives the high pass coefficients d(n). c(n) does not get affected and the choice of the predictor depends only on the data.

RESULTS

We have applied the adaptive prediction algorithms to 256×256 8 bit images. The original image is transformed using adaptive prediction algorithms and it is compressed using Set Partitioning In Hierarchical Trees (SPIHT) algorithm and the performance is compared with the performance of popular 9/7 transform (Said and Pearlman, 1996). The performance metric PSNR for N×N image (Peak Signal to Noise Ratio) at different bit rates in bits per pixel (bpp) is computed as Eq. 21:

$$\text{PSNR} = 10 \log_{10} \left[\frac{255^2}{\text{MSE}} \right] \text{db} \quad (21)$$

where, MSE is the mean square error given by Eq. 22:

$$\text{MSE} = \frac{1}{N^2} \sum_i \sum_j (X(i,j) - Y(i,j))^2 \quad (22)$$

taking X as the original image and Y as the reconstructed image.

Table 1: PSNR in db for the test images at different bit rates

Images	Bit rate in bpp	9/7 Wavelet	APM1*	APM2**
Circle	0.1	23.59	24.01	22.56
	0.2	26.01	26.80	26.02
	0.4	30.01	32.21	30.96
	0.6	33.83	36.79	34.02
	0.8	36.69	39.23	39.26
	1.0	38.02	42.69	41.86
	0.1	23.32	23.56	23.47
Cameraman	0.2	26.01	26.02	25.83
	0.4	29.02	29.35	28.75
	0.6	31.02	31.54	30.57
	0.8	32.83	32.86	32.43
	1.0	34.60	34.96	33.83
	0.1	24.02	18.60	19.05
	0.2	25.02	21.02	22.36
Lena	0.4	26.38	22.34	23.89
	0.6	28.78	25.74	25.87
	0.8	33.58	28.29	29.65
	1.0	35.07	30.56	30.88
	0.1	21.32	22.72	24.14
	0.2	22.69	23.61	25.21
	0.4	24.68	25.81	27.89
Baboon	0.6	26.41	27.91	29.96
	0.8	28.32	30.67	32.81
	1.0	29.11	31.98	34.21
	0.1	27.09	26.77	25.87
	0.2	28.60	27.77	26.76
	0.4	29.80	28.17	27.98
	0.6	31.82	29.37	29.04
Goldhill	0.8	33.20	31.77	31.09
	1.0	34.89	33.77	32.89

APM1*: Adaptive Prediction method; PM2**: Adaptive Prediction method

DISCUSSION

Table 1 gives the PSNR at different bitrates for images of different frequency distributions. The performance is compared for the test images circle, Lena and cameraman in Fig. 4-6 respectively. For the edge dominated circle image, we observe that the image transformed with the adaptive lifting has sharp edges as shown in Fig. 7.

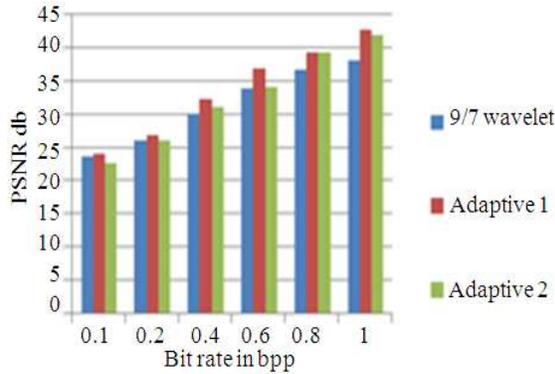


Fig 4: Bar chart for circle image

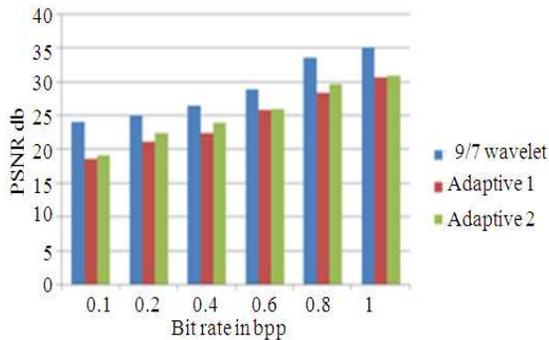


Fig. 5: Bar chart for Lena Image

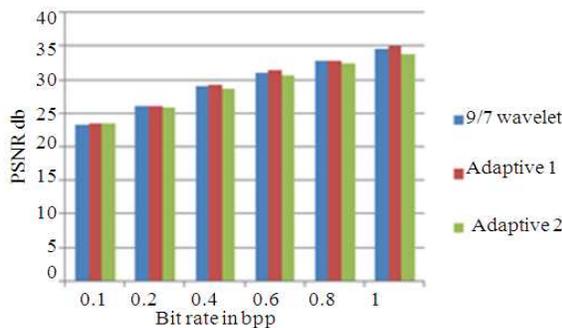


Fig. 6: Bar chart for Cameraman image

The PSNR values and the visual quality of the cameraman image using the proposed algorithms are comparable with that of 9/7 wavelet transforms as shown in Fig. 8. Since Lena image is a smooth image, 9/7 wavelet transform gives better PSNR and visual quality than the adaptive prediction methods as shown in Fig. 9. The proposed adaptive algorithms gives better performance than 9/7 wavelet for the Baboon image which contains a narrow range of luminance levels and a large number of details. The performance of the proposed algorithm is comparable with 9/7 wavelet for the Goldhill image.

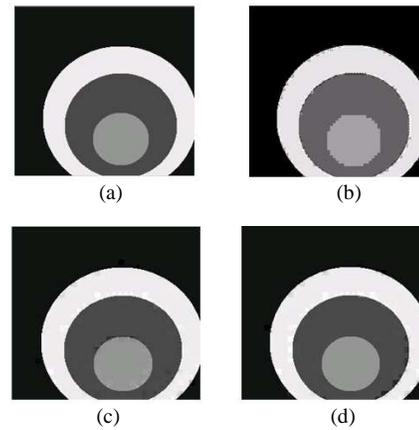


Fig. 7: Circle image compressed at the rate of 0.6 bpp

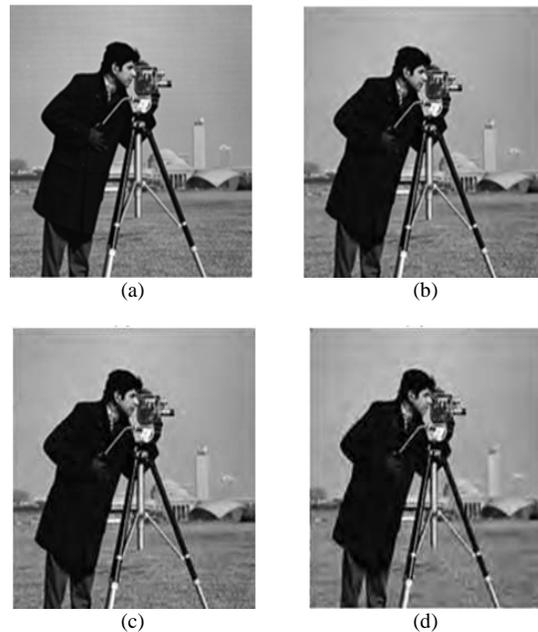


Fig. 8: Cameraman image compressed at the rate of 0.6 bpp



Fig. 9: Lena image compressed at the rate of 0.6 bp (a) Original Image (b) 9/7 Lifting (c) Adaptive method 1 (d) Adaptive method 2

CONCLUSION

Lifting allows us to incorporate adaptivity and nonlinear operators into the transform. The proposed methods efficiently represent the edges. These adaptive lifting transforms appear promising for image compression. These adaptive transforms reduce edge artifacts and ringing and give improved PSNR for edge dominated images like Circle and Baboon. The performance is comparable with 9/7 wavelet transform for images like Cameran and Goldhill with moderate frequency distributions. For smooth images like Lena, 9/7 transform gives much better results.

REFERENCES

Asamwar, R.S., K. Bhurchandi and A.S. Gandhi, 2010. Successive image interpolation using lifting scheme approach. *J. Comput. Sci.*, 6: 969-978. DOI: 10.3844/jcssp.2010.969.978

Claypoole, R.L. G.M. Davis, W. Sweldens, and R.G. Baraniuk, 2003. Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Image Process.*, 12: 1449-1459. DOI: 10.1109/TIP.2003.817237

Cohen, A. I. Daubechies and I.C. Feauveau, 1992. Biorthogonal bases of compactly supported wavelets. *Commun. Pure Applied Math.*, 45: 485-560. DOI: 10.1002/cpa.3160450502

Daubechies I. and W. Sweldens, 1996. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4: 247-269. DOI: 10.1007/BF02476026

Ghrare S.E., M.A.M. Ali, K. Jumari and M. Ismail, 2009. An efficient low complexity lossless coding algorithm for medical images. *Am. J. Applied Sci.*, 6: 1502-1508. DOI: 10.3844/ajassp.2009.1502.1508

Gonzalez, R.C., R.E. Woods and S.L. Eddins, 2004. *Digital Image Processing using MATLAB*. 1st Edn., Pearson Prentice Hall, Upper Saddle River, NJ., ISBN: 9780130085191, pp: 609.

Hafiz, D.A., W.M. Sheta, S. Bayoumi and B.A.B. Youssef, 2011. A New approach for 3D range image segmentation using gradient method. *J. Comput. Sci.*, 7: 475-487. DOI: 10.3844/jcssp.2011.475.487

Mallat, S.G., 1989. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. Acoust., Speech, Signal Process.*, 37: 2091-2110. DOI: 10.1109/29.45554

Pennebaker, W.B. and J.L. Mitchell, 1993. *JPEG Still Image Data Compression Standard*. 1st Edn., Springer, Van Nostrand Reinhold, New York, ISBN: 0-442-01272-1, pp: 638.

Said, A. and W.A. Pearlman, 1996. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circ. Syst. Video Technol.*, 6: 243-250. DOI: 10.1109/76.499834

Sweldens, W., 1996. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186-200, 1996. DOI: 1006/acha.1996.0015

Taubman, D.S. and M.W. Marcellin, 2002. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. 1st Edn., Springer, Kluwer Academic Publishers, Boston, ISBN: 0-7923-7519-x, pp: 773.

Uytterhoeven, G, D. Roose and A. Bultheel, 1997. "Wavelet transform using the lifting scheme", Report ITA-Wavelet WP1.1. Department of Computer Science, Katholieke University Leuven.