Image Compression Using Lifting Scheme

Dr. Hadeel N. Abdullah & Yasir A. Ahmed Al-Obaidi*

Received on:30/9/ 2009
Accepted on:30/6/2010

Abstract

This paper introduces, firstly, a proposed method of computing one and two-dimensional wavelet transform .The proposed method reduces heavily processing time for decomposition of image keeping or overcoming the quality of reconstructed images. Also, the inverse procedures of all the transformations for multi- dimensional cases verified.

Secondly, computes quantization and run length encoder. Different types of quantization are presented in this paper with effects of these differences on Compression Ratio (CR).

Thirdly, compute PSNR, RMSE, CR, and size. The effect noted this difference in levels of FLWT on same picture, where PSNR, MSRE, SIZE, and CR different from one level to another.

Keywords: Image Compression, Lifting Wavelet Transform, Run Length Encoding.

ضغط الصورة بأستخدام مخطط الرفع (تحويل الموية)

الخلاصة

يُقدّم هذا البحث أولاً, طريقة مقترحة لحساب الموية ذو البعد الواحد وذو البعدين. الطريقةَ المُقتَرَحة تُخفضُ كثيراً من زمن المعالجة الصورة المتحللة وتَبقي أو تَتغلّبُ على بناء تركيب الصورِ المُعَادة.أيضاً التحويلات المعكوسة لكُلّ ما قيل سابقاً من التحويل في الإبعاد المتعددة للحالات المتحققة.

ثانياً، أنواع مختلفة مِن التكميم قدّمتْ في هذه الإطروحةِ ثم تطبيق ضغط الصورة باستخدام RLE وتُؤثّرَ هذه الإختلافاتِ على نسبةِ الضغطِ.

ثالثاً، يَحْسبُ PSNR و RMSE و CR، وحجم. لاحظَ التأثيرُ هذا الإختلافِ في مستويات FLWT على الصورةِ نفسهاِ، حيثPSNR، MSRE، حجم، وCR مختلف من المستوى الاول إلى آخرِ.

1. Introduction

Data compression is the process of converting data files into smaller files for efficiency of storage and transmission. It is the key in the rapid progress being made in information technology. Simply, it would not be practical to put images, audio, and/or video alone on websites without compression.

Image coding consists of mapping images to strings of binary digits. A good image coder is one that produces binary strings whose lengths

*Electrical & Electronic Engineering Department, University of Technology/ Baghdad

https://doi.org/10.30684/etj.28.17.5
2412-0758/University of Technology-Iraq, Baghdad, Iraq

are on average much smaller than the original canonical representation of the image. In many imaging applications, exact reproduction of the image bits is not necessary. In this case, one can perturb the image slightly to obtain a shorter representation [1].

Image compression is one of the most important and successful applications of the wavelet transform. Mature wavelet based image coders like the JPEG2000 standard is available, gaining in popularity, and easily to perform the traditional coders.

Unlike image compression, based the Discrete Cosine Transform (DCT) like JPEG, the performance of a wavelet-based image coder depends to a large degree on the choice of the wavelet. This problem is usually handled by using standard wavelets that are not specially adapted to a given image, but that are known to perform well on photographic images.

Application of digital images often is not viable due to high storage or transmission costs. Image compression technology offers a possible solution. The basic goal of image compression is to reduce the bit rate of an image then to minimize the communication channel capacity or digital storage memory requirements with maintaining necessary fidelity in the image, or, equivalently, to obtain the best possible fidelity for a given bit rate[2].

Two major features should be available for good and successful data compression, low complexity(e.g., ease of decoding) and the efficient implementation (in terms of memory requirements). In addition to these features, the process time delay and the quality after compression are among the major concerns for a good compression technique[3].

In order to be useful, a compression algorithm has a corresponding decompression algorithm that, given the compressed file, reproduces the original file. There have been many types of compression algorithms developed .These algorithms fall into two broad types, lossless algorithms and lossy algorithms. A lossless algorithm reproduces the original data exactly. A lossy algorithm, as its name implies, loses some redundant data. Data loss may be unacceptable in many applications. For example, text compression must be lossless because a very small difference can result in statements with totally different meanings. There are also many situations where loss may be either unnoticeable or acceptable. In image compression, for example, the exact reconstructed value of each sample of the image is not necessary. Depending on the quality requirements of the reconstructed image, some losses of information can be accepted[4].

2. Wavelet-based Compression System

The basic structure of a wavelet-based compression system is summarized in Fig. (1)[5]:

The basic functions of each block in the encoder are:

1. Forward Transform: Decorrelation of image pixels is performed here via a multiresolution wavelet transform. This block is equivalent to the forward DCT block in a classic DCT-based compression system.

2. Quantization: Quantization of wavelet coefficients can produce a

great compression rate in the lossy mode. In the lossless mode an error residual has to be stored in order to achieve perfect reconstruction of the image. Special coding schemes are generally used here to obtain an embedded bit stream output.

3. Entropy coding: Arithmetic coding or Huffman coding can be applied to the quantized coefficients to get a better compression ratio. Nevertheless this final step can be skipped in order to obtain a faster compression system.

Through the application of the wavelet transform a component is split into numerous frequency bands (i.e., subbands). Due to the statistical properties of these subband signals, the transformed data can usually be coded more efficiently than the original untransformed data.

3. Lifting Scheme

Wavelets based on dilations and translations of a mother wavelet are referred to as first generation wavelets or classical wavelets. Second generation wavelets, i.e., wavelets which are not necessarily translations and dilations of one function, are much more flexible and can be used to define wavelet bases for bound intervals, irregular sample grids or even for solving equations or analyzing data on curves or surfaces. Second generation wavelets retain the powerful properties of first generation wavelets, like fast transform, localization and good approximation. Lifting scheme is a rather new method for constructing wavelets. The main difference with the classical constructions is that; it does not rely on the Fourier transform. In this way, Lifting can be used to construct second generation wavelets Lifting scheme, and efficiently implement classical wavelet transforms. Existing classical wavelets can be implemented with Lifting Scheme by factorization them into Lifting steps[6].

3.1 Forward Transform:

The forward transform of Fast Lifting scheme is composed of three steps, depicted as follows [7]:

- Split.
- Predict.
- Update.

Split: This stage does not do much except for splitting the signal into two disjoint sets of samples. In this case one group consists of the even indexed samples s_{2l} and the other group consists of the odd indexed samples s_{2l+1}. Each group contains half as many samples as the original signal. The splitting into even and odds is called the Lazy wavelet transform. Thus an operator is built so that[7]

$$(even_{j-1}, odd_{j-1}) := \text{Split}(s_j) \quad \dots (1)$$

Predict: The even and odd subsets are interspersed. If the signal has a local correlation structure, the even and odd subsets will be highly correlated. In other words given one of the two sets, it should be possible to predict the other one with reasonable accuracy. In the Haar case the prediction is particularly simple. An odd sample $s_{j,2l+1}$ will use its left neighboring even sample $s_{j,2l}$ as its predictor. Then the detail $d_{j-1,l}$ be the difference between the odd sample and its prediction:

$$d_{j-1,l} = s_{j,2l+1} - s_{j,2l} \qquad \dots (2)$$

which defines an operator P such that

$$d_{j-1} = odd_{j-1} - P(even_{j-1}) \qquad \dots (3)$$

It should be possible to represent the detail more efficiently. Note that if the original signal is constant, then all details are exactly zero [7].

Update: One of the key properties of the coarse signals is that they have the same average value as the original signal, i.e., the quantity

$$S = 2^{-j} \sum_{l=0}^{2^j-1} s_{j,l} \qquad \dots (4)$$

is independent of j. This results in the fact that the last coefficients $s_{0,0}$ is the DC component or overall average of the signal. The update stage ensures this fact by letting

$$s_{j-1,l} = s_{j,2l} + d_{j-1,l}/2 \qquad \dots (5)$$

Substituting this definition:

$$(s_{j,2l} + d_{j-1,l}/2) =$$

$$\sum_{l=0}^{2^{j-1}} s_{j-1,l} = \sum_{l=0}^{2^{j-1}} 1/2 \sum_{l=0}^{2^{j-1}} (s_{j,2l} + s_{j,2l+1}) =$$

$$1/2 \sum_{l=0}^{2^j} s_{j,l}$$

$$\dots (6)$$

which defines an operator U of the form

$$s_{j-1} = even_{j-1} + U(d_{j-1}) \qquad \dots (7)$$

The even locations can be overwritten with the averages and the odd ones with the details. An abstract implementation is given by [7]:

$$(odd_{j-1}, even_{j-1}) := Split(s_j) \quad \dots (8)$$

$$odd_{j-1} -= P(even_{j-1}) \qquad \dots (9)$$

$$even_{j-1} += U(odd_{j-1}) \qquad \dots (10)$$

These three stages are depicted in a wiring diagram in Fig. (2)

3.1.1 Computation of Fast Lifting Wavelet Transform (FLWT) for 1-D Signal

To compute a signal level FLWT for 1-D signal, the following steps should be followed:

1. Checking input dimensions: Input vector should be of length N, where N must be power of two.
2. Split: divides the input data into odd and even elements.
3. Predict: predicts the odd elements from the even elements
[odd new = odd - even].
4. Update: replaces the even elements with an average
[even new = even + odd new/2].

3.1.2 Computation of Fast Lifting Wavelet Transform (FLWT) for 2-D Signal

To compute a signal level FLWT for 2-D signal, the following steps should be followed:

1. Checking input dimensions: Input vector should be of length NxN, where N must be power of two.
2. Split row: divides the input data into odd and even elements of each column.
3. Predict row: predicts the odd elements from the even elements
[odd new = odd - even].
4. Update row: replaces the even elements with an average
[even new = even + odd new/2].
5. Split column: divides the (odd new and even new) data into odd and even elements of each row.
6. Predict column: predicts the odd elements from the even elements
[odd1 new= odd1 – even1].
7. Update column: replaces the even elements with an average

[even1 new = even1 + odd1 new/2].

3.2 Inverse Transform:

It is easy to verify, that the perfect reconstruction in case of lossless compression is guaranteed, since that same value of the modified predictor and update operator is added and subtracted. The inverse transform of fast lifting scheme is composed of three steps, depicted as follows [7]:

1-Undo update.
2- Undo predict.
3- Merge.

Undo update: Given d_j and s_j, by subtracting the update information the even samples can be recovered

$$even_{j-1} = s_{j-1} - U(d_{j-1}) \quad \dots (11)$$

.In the case of Haar:

$$s_{j,2l} = s_{j-1,l} - d_{j-1,l}/2 \quad \dots (12)$$

Undo predict: Given $even_{j-1}$ and d_{j-1}, by adding the prediction information the odd samples can be recovered

$$odd_{j-1} = d_{j-1} + P(even_{j-1}) \quad \dots (13)$$

In the case of Haar:

$$s_{n,2l+1} = d_{n-1,l} + s_{n,2l} \quad \dots (14)$$

Merge: Now we have to zipper the even and odd samples together to recover the original signal. This is the inverse Lazy wavelet:

$$s_j = Merge(even_{j-1}, odd_{j-1}) \dots (15)$$

Assuming that the even slots contain the averages and the odd ones contain the difference, the implementation of the inverse transform is:

$$even_{j-1} -= U(odd_{j-1})$$
$$odd_{j-1} += P(even_{j-1}) \quad \dots (16)$$
$$s_j := Merge(odd_{j-1}, even_{j-1}$$

These three stages are depicted in a wiring diagram in Fig. (3)

3.2.1 Computation of Inverse Fast Lifting Wavelet Transform (IFLWT) for 1-D Signal

To compute the signal level IFLWT for 1-D signal, the following steps should be followed:

1. Undo update:
   [even = even new - odd new/2].
2. Undo predict:
[odd = odd new + even].
3. Merge: solder odd and even elements into input data.

3.2.2 Computation of Inverse Fast Lifting Wavelet Transform (IFLWT) for 2-D Signal

1. Undo update column:
   [even1 = even1 new - odd1 new/2].
2. Undo predict column:
   [odd1 = odd1 new + even1].
3. Merge column: solder the (odd1 and even1) of each row into odd and even elements.
4. Undo update row:
   [even = even new - odd new/2].
5. Undo predict row:
   [odd = odd new + even].
6. Merge row: solder odd and even elements of each column into the input data.

3.3 A Computer Test

A general program computing a single-level FLWT using Haar filter is written using Matlab V.R2008a for a general NxN 2-D signal (or image). As shown in Fig.(4a), the original “Lena512.png” image dimensions are 512x512

(NxN). After a single-level of wavelet decomposition using Haar filter, image dimensions will be a matrix of 512x512 (NxN), as shown in Fig. (4b). The upper-left most, LL, subband of 256x256 dimensions, is zoomed in as in Fig. (4c)

An example test is applied to the decomposed Lena image to reconstruct the original "Lena" image by using a general computer program computing a single-level 2-D IFLWT and the result is shown in Fig. (4d).

4. Quantization

Quantization is the process of selecting the discarded visual information without a significant loss in the visual effect. Quantization reduces the number of bits needed to store an integer value by reducing the precision of the integer. A uniform quantizer does not depend on the data. It divides the range of the values into quantization intervals of equal length [8].

Subjective experiments involving the human visual system have resulted in the JPEG standard quantization matrix. With a quality level of 50, this matrix renders both high compression and excellent decompressed image quality [9].

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad \ldots (17)$$

If, another level of quality and compression is desired, scalar multiples of the JPEG standard quantization matrix may be used. For a quality level greater than 50 (less compression, higher image quality), the standard quantization matrix is multiplied by (100-quality level)/50. For a quality level less than 50 (more compression, lower image quality), the standard quantization matrix is multiplied by 50/quality level. The scaled quantization matrix is then rounded and clipped to have positive integer values ranging from 1 to 255. For example, the following quantization matrices yield quality levels of 10 and 90 [9].

$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix} \quad \ldots (18)$$

$$Q_{90} = \begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 12 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 21 \\ 14 & 18 & 19 & 20 & 22 & 20 & 20 & 20 \end{bmatrix} \quad \ldots (19)$$

Quantization is achieved by dividing each element in the transformed image matrix D by the corresponding element in the quantization matrix, and then rounding to the nearest integer value [9].

$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right) \quad \ldots (20)$$

5. Run Length Encoding (RLE)

RLE works by reducing the physical size of a repeating string of characters. This repeating string, called a run, is typically encoded into two bytes. The first byte represents the number of characters in the run and is called the run count. In practice, an encoded run may contain 1 to 128 or 256 characters; the run count usually contains the number of characters minus one (a value in the range of 0 to 127 or 255). The second byte is the value of the character in the run, which is in the range of 0 to 255, and is called the run value. Uncompressed, a character run of 15A characters would normally require 15 bytes to store: AAAAAAAAAAAAAAA. The same string after RLE encoding would require only two bytes: 15A.The 15A code generated to represent the character string is called an RLE packet. Here, the first byte, 15, is the run count and contains the number of repetitions. The second byte, A, is the run value and contains the actual repeated value in the run [10]. The flow chart shown in fig. (5) explains the implementation of Run Length Encoder.

6. Compression Ratio

It is defined as the ratio between he size of the original image data (file) and the size of overall compressed data file [11].

$$C.R. = \frac{Original\ File\ Size}{Compressed\ File\ Size} \quad \ldots (21)$$

7. Fidelity Criteria

Fidelity means "how accurately the signal recovered in the received represents the one sampled in the sender".

Quality means "a composition of the fidelity and the delay used to replay the samples" [12].

Methods for image quality evaluation can be classified as objective and subjective measures:

Objective measure: By objective measures some statistical indices are calculated to indicate the reconstructed image quality. it includes :

RMSE: is found by taking the square root of the error squared divided by the total number of pixels in the image ("mean").

The MSE in N*M decompressed image can be defined as:

$$MSE = \frac{1}{N*M}\sum_{x=1}^{x=N}\sum_{y=1}^{y=M}[f(x,y)-f^{'}(x,y)]^2 \quad \ldots (22)$$

The RMSE is computed as follow:

$$RMSE = \sqrt{MSE} \quad \ldots (23)$$

Where f (x, y) is the original image data and f '(x, y) is the reconstructed image value [13].

PSNR: Another quantitative measure is the peak signal-to-noise ratio (PSNR), based on the root mean square error of the reconstructed image. The formula for PSNR is given as follow:

$$PSNR = 10\log\left(\frac{(255)^2}{MSE}\right) \ldots (24)$$

Subjective measure: subjective measure viewers read images directly to determine their quality. Subjective evaluation by viewers is still a method commonly used in measuring image quality. The subjective test emphatically examines fidelity and at the same time considers image intelligibility. When taking subjective test, viewer's focus on the difference between reconstructed image and the original image, they notice such details where information loss cannot be accepted [12].

8. Result

This section presents the results that were performed on the design of the wavelet based Image Compression using MATLAB R2008a program. Table (1) shows the effect of quantization on compression ratio. Table (2) shows computed PSNR, RMSE, and CR.

The effect of quantization shows the difference in levels of FLWT on same picture, where PSNR, MSRE, SIZE, and CR differs from one level to another. For example pepper256.png (40KB) in level 1 has PSNR=26.5335db, RMSE=12.0190, SIZE=2KB, and CR=4.6895 while in level 2 PSNR=32.2241db, RMSE=6.2422, SIZE=1KB, and CR=20.1836.

9. Conclusions

Wavelet based image compression is ideal for adaptive compression since it is inherently a multi-resolution scheme. Variable levels of compression can be easily achieved. The number of wavelet stages can be varied, resulting in different number of sub bands.

Fast Lifting Wavelet Transform for hardware acceleration of the Discrete Wavelet Transform is based on the Lifting scheme, which is used to enhance the performance of the multimedia tools as JPEG2000 and MPEG-4 by hardware acceleration.

References

[1] Davis G. M. and Nosratinia A.:" Wavelet-Based Image Coding: An Overview",1999. Site:www.geoffdavis.net/pape/**accsc**.pdf.

[2] Majeed A. S.: "Image Compression Using Hybrid Transform", M. Sc. Thesis, University of Baghdad, 2005.

[3] Steinmetz R. and Wolf L.: " Multimedia Systems: Compression Techniques", Darmstadt University of Technology, Industrial Process and System Communications, Merckstraße 25 – 64283 Darmstadt – Germany.

[4] Xiao P.: "Image Compression By Wavelet Transform", M. Sc. Thesis, East Tennessee State University, August 2001.

[5] Hassen W. M: "Image Compression Using Wavelet Transform", M.Sc. Thesis, University of Al-Mustansiriyah, Nov.2000.

[6] Zafairfar B.: "Micro-Codable Discrete Wavelet Transform", M.Sc. Thesis, Delft University of Technology, 2002.

[7] Schröder P. and Sweldens W.: "Wavelet in Computer Graphics", IndustrialMathematics Initiative,Mathematics Department, University of South Carolina, 1996.

[8] Quinteros J. and Ruedin A.: "Quantization of Wavelet Coefficients", JAIIO 32 Journadas Argentinas de Informatica e Investigacion Operativa, 2003.

[9] Cabeen K. and Gent P.: "Image Compression and The Discrete Cosine Transform", College of the Redwoods, 1998.

[10] O'Reilly :" Run Length Encoding (RLE)", Encyclopedia of Graphics File Formats Book, Second Edition, 2007. Site:http://www.fileformat.info/mirror/egff/ch09_03.htm

[11] Rosenfeld A. and Kak A., "Digital Picture Processing", 2nd Ed.,Academic Press, New York, 1982.

[12] Saffor A., Rahman R. A., Hoon N. K., Dowsett D., "Objective and Subjective Evaluation of Compressed Computed Tomography (CT) Images" , The Internet Journal of Radiology, 2002.
Site: http://www.ispub.com/ostia/index.php?xmlFilePath=index.xml

[13] Umbaugh S. E.: "Computer Vision and Image Processing: A Practical Approach Using CVIP Tools", Prentice Hall PTR, 1998.

Table (1) The Effect of Quantization on Compression Ratio

Name Image	Quantization	Compression Ratio
Lena512.png (level 1)	Q90	4.5574
Lena512.png (level 1)	Q50	131072
Papers256.png (level 1)	Q90	4.6895
Papers256.png (level 1)	Q50	32768
Lena.tif (level 1)	Q90	4.5502
Lena.tif (level 1)	Q50	131072
Papers256.png (level 2)	Q90	20.1836

Table (2) Size, Compression Ratio, REMSE, and PSNR Results of the Filtered Image with Wavelet Transform.

Name Image	Original Image Size (KB)	Compression Image Size (KB)	Compression Ratio	RMSE	PSNR (db)
Lena512.png (level 1)	**148**	**5**	**4.5574**	**25.4753**	**20.0084**
Pepper256.png (level 1)	**40**	**2**	**4.6895**	**12.0190**	**26.5335**
Cameraman256. png (level 1)	**37**	**2**	**4.6972**	**9.2187**	**28.8374**
Lena.tif (level 1)	**257**	**113**	**4.5502**	**25.4576**	**20.0145**
Pepper256.png (level2)	**40**	**1**	**20.1836**	**6.2422**	**32.2241**

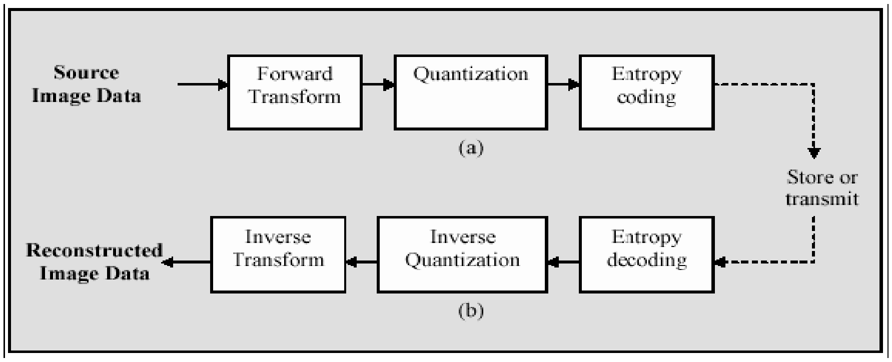

Figure (1) General Block Diagram of a Wavelet-based Compression Scheme (a) Encoder and (b) Decoder

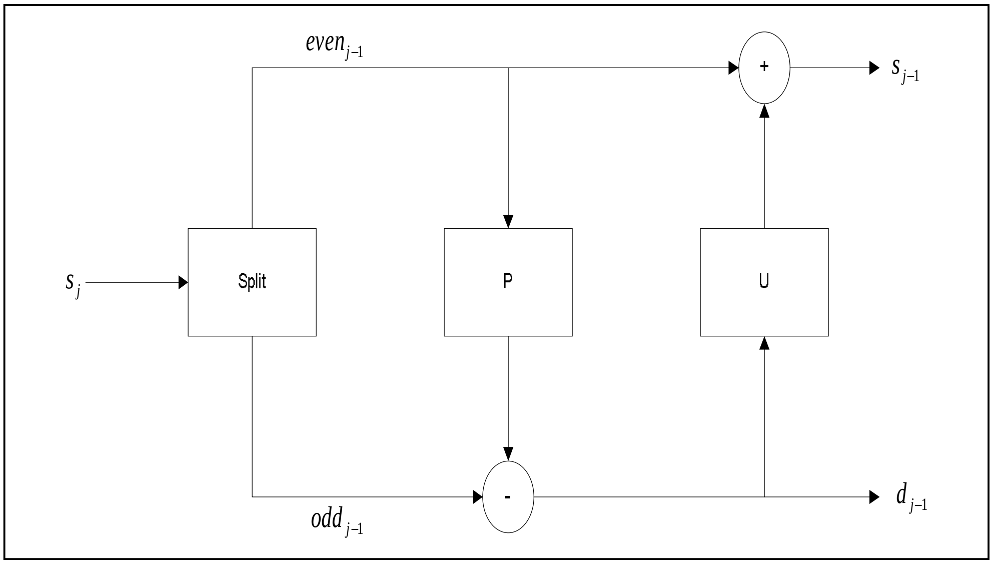

Figure (2) The Lifting Scheme ,Forward Transform

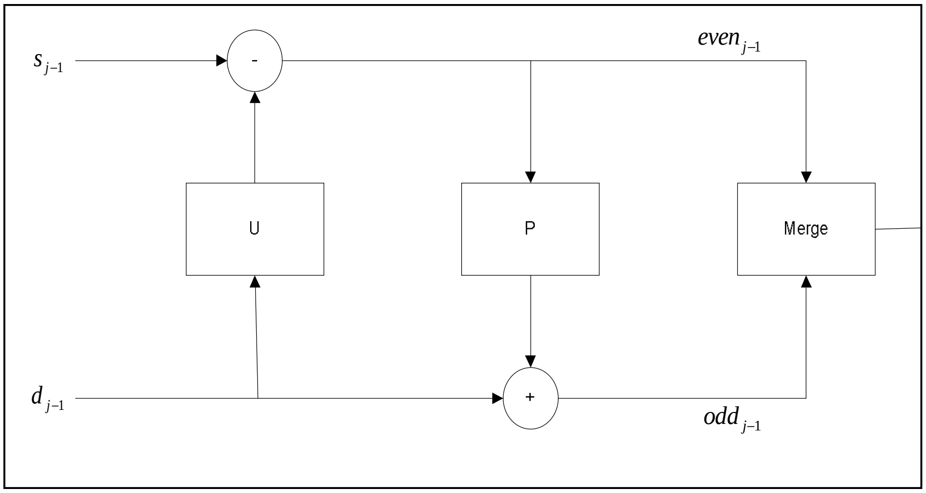

Figure (3) The Lifting Scheme, Inverse Transform

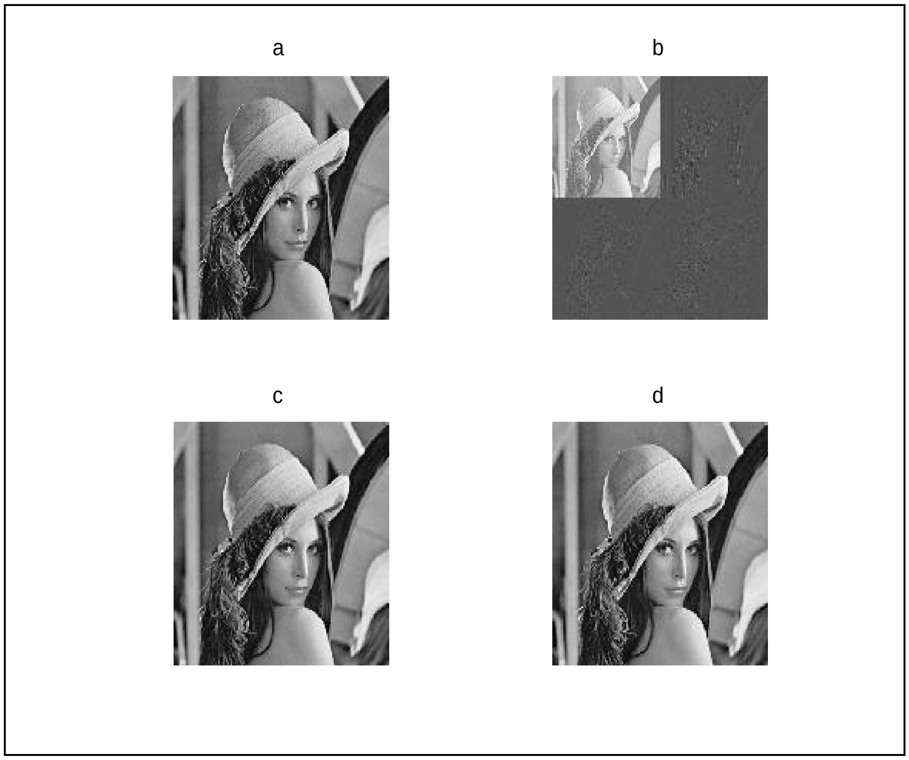

Figure (4) Lena Image, (a) Original, (b) After Single Level of 2-D FLWT Using Haar, (c) Zoomed in Upper-Left Front Most Subband After 2-D FLWT and (d) Reconstructed Image.

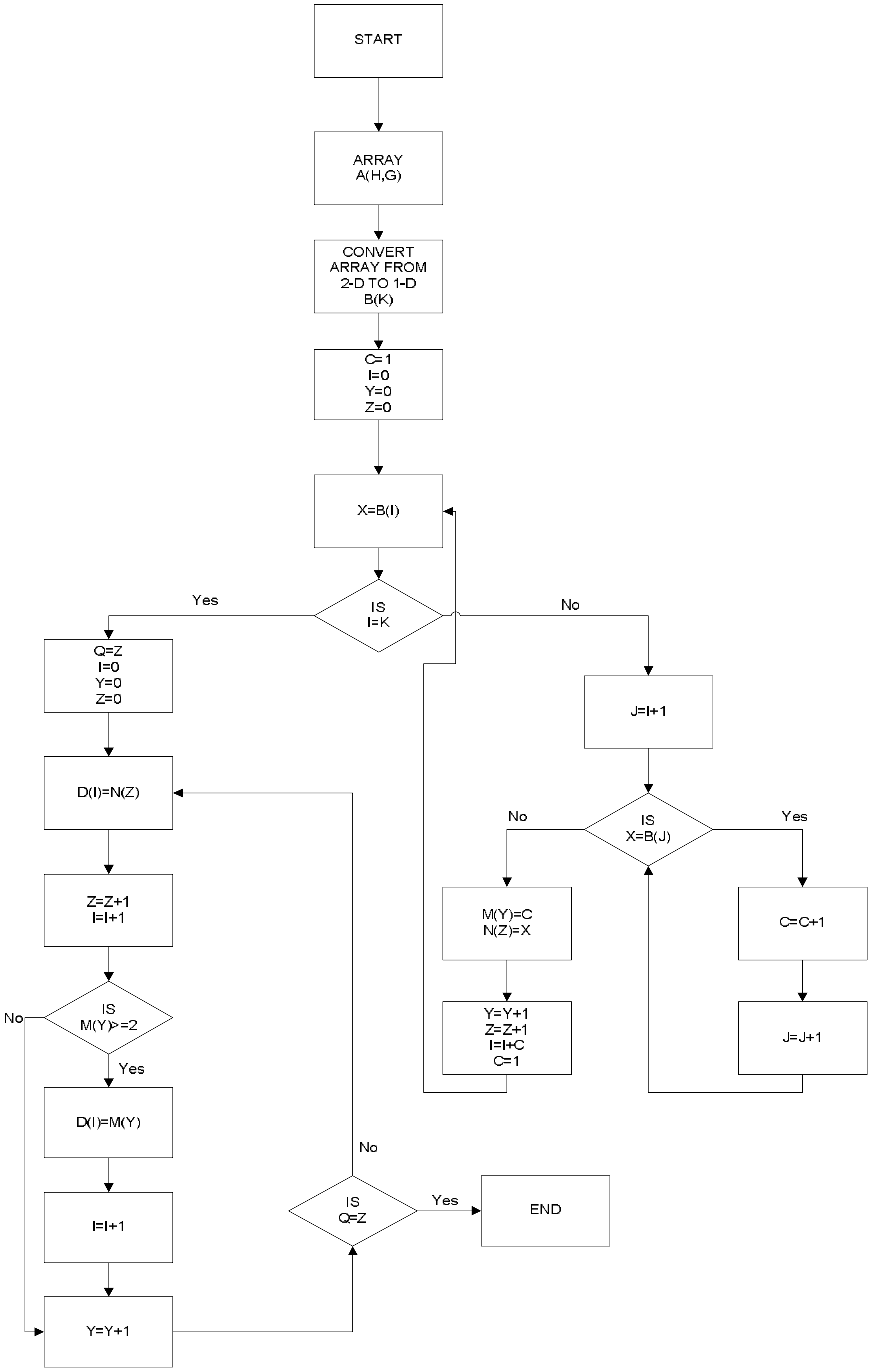

Figure (5) Flow Chart of the Run Length Encoding