ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 5, No. 3, 2010, *pp*. 163-172

JPEG and Sequential Search Algorithm applied on Low-Frequency Sub- Band for Image Compression (JSS)

Mohammed Mustafa Siddeq [+]

Software Engineering Depart. Technical College Kirkuk Iraq

(*Received July 11, 2009, accepted October 22, 2009*)

Abstract. In this research we introduce and idea for image compression, based on JPEG technique and Sequential Search Algorithm, at first using Discrete Wavelet Transform (DWT) to obtain LL sub-band, then apply JPEG technique on the LL sub-band. JPEG technique consist of; JPEG Transformation, and JPEG Coding. The LL sub-band transformed by JPEG transformation for obtaining more compression ratio. Before apply JPEG coding, our algorithm used feedbacks system for inverse JPEG transformation to get decoded LL sub-band, and then the difference between decoded LL and original LL are stored in a new matrix called D-Matrix, finally the D-Matrix compressed by Sequential Search Algorithm. The Decompression algorithm consists of; Inverse JPEG transformation to get decoded LL, Sequential Search Algorithm to find D-Matrix, Add D-Matrix with the decoded LL and apply inverse DWT to get decompressed image. Our algorithm tested on two different gray level images and compared with JPEG and JPEG2000.

Keywords: Discrete Wavelet Transform, JPEG Technique, Minimize Algorithm, Sequential Search Algorithm

1. Introduction

Since the mid-80s, members from both the International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) have been working together to establish a joint international standard for the compression of grayscale and color still images. This effort has been known as JPEG, the Joint Photographic Experts Group the "joint" in JPEG refers to the collaboration between ITU and ISO) [1,2,4,5]. Officially, JPEG corresponds to the ISO/IEC international standard 10928-1, digital compression and coding of continuous-tone (multilevel) still images or to the ITU-T Recommendation T.81. The text in both these ISO and ITU-T documents is identical. The process was such that, after evaluating a number of coding schemes, the JPEG members selected a DCT1-based method in 1988. From 1988 to 1990, the JPEG group continued its work by simulating, testing and documenting the algorithm. JPEG became a Draft International Standard (DIS) in 1991 and an International Standard (IS) in 1992 [5,6]. Lossy compression is compression in which some of the information from the original message sequence is lost. This means the original sequences cannot be regenerated from the compressed sequence. Just because information is lost doesn't mean the quality of the output is reduced [8]. For example, random noise has very high information content, but when present in an image or a sound file, we would typically be perfectly happy to drop it. Also certain losses in images or sound might be completely imperceptible to a human viewer (e.g. the loss of very high frequencies). For this reason, lossy compression algorithms on images can often get a factor of 2 better compressions than lossless algorithms with an imperceptible loss in quality. However, when quality does start degrading in a noticeable way, it is important to make sure it degrades in a way that is least objectionable to the viewer (e.g., dropping random pixels is probably more objectionable than dropping some color information). For these reasons, the ways most lossy compression techniques are used are highly dependent on the media that is being compressed [11]. Lossy compression for sound, for example, is very different than lossy compression for images.

2. Compression Algorithm

In this research we introduce an idea for image compression depends on JPEG technique and Sequential

[+] Corresponding author. Tel.: +9647701256324.
E-mail address: - mamadmmx76@yahoo.com

Search algorithm. We will explain Sequential Search Algorithm in section 4, while in this section we describe how the JPEG technique applied on the low-frequency coefficients.

2.1. Discrete Wavelet Transform (DWT)

A single-stage wavelet transformation consists of four frequency bands as shown in Fig. 1. The top-left corner of the transformed image "LL" is the original image, low-frequency coefficients and the top-right corner "HL" consists of residual vertical frequencies (i.e. the vertical component of the difference between the "LL" image and the original image) [8-11]. The bottom-left comer "LH" contains residual horizontal frequencies, whilst the bottom-right corner "HH" contains residual diagonal frequencies.

The main reason for use DWT, for reduce an image dimensions, and the high-frequencies coefficients are ignored (i.e. not used in this research), this will effects on the image quality, while this process increasing compression ratio.

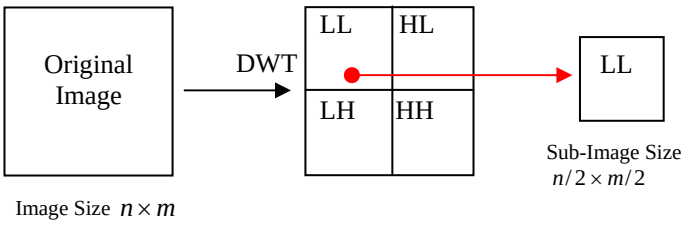

Fig .1: Original image dimensions reduced by DWT

2.2. JPEG Technique

JPEG is a sophisticated lossy compression method for color or grayscale images. An important feature of JPEG is its use the quality parameter "*Quality*", allowing the user to adjust the amount of the data lost over a very wide range [1]. The problem of this technique, the compressed images, degraded when the user select low quality parameter for obtains for higher compression ratio [1,3]. In this research we apply this technique on the sub-image LL, and the image quality problem are solved in the Section 2.3. The JPEG technique illustrated in the following steps:

1. The pixels of an image are organized in groups of 8×8 pixels called *data units*, and each data unit is compressed separately. If the number of image rows or columns is not a multiple of 8, the bottom row and the rightmost column are padded with zeros as many times as necessary [4].

2. The discrete cosine transform (DCT) is then applied on the each group of 8×8 pixels to create an 8×8 map of frequency domain. The frequency domain consists of; DC coefficient at the first location in 8×8 frequency component, which is represents average pixels value. The components are called AC, which is representing high-frequency coefficients [5].

3. Each of the 64 frequency components in an 8×8 map are divided by a separate numbers its called *Quantization Coefficient* (QC), and then rounded to an integer. This is where information is irretrievably lost. A simple quantization table *Q* is computed, based on one parameter *R* specified by the user. A simple expression such as $Qij = 1+(i + j) \times R$ guarantees that QCs start small at the upper-left corner and get bigger toward the lower-right corner [6].

4. Each 8×8 map frequency component scanned by using zigzag scan, to create a one dimensional array contains 64 quantized frequency coefficients. The one-dimensional array encoded using a combination of Run-Length-Encoding (RLE) and Huffman coding [1,2,6].

The Fig-2 illustrates sub-image compressed by JPEG technique

The parameter "R" in QC computed by selecting maximum coefficient from an 8x8 frequency components and then multiply by ratio=75%. This process removes small high-frequency coefficients to increase zeros and convert large coefficients to small coefficients, for gaining high compression ratio[6]. But in the decompression part, we obtain low quality image (degraded image).

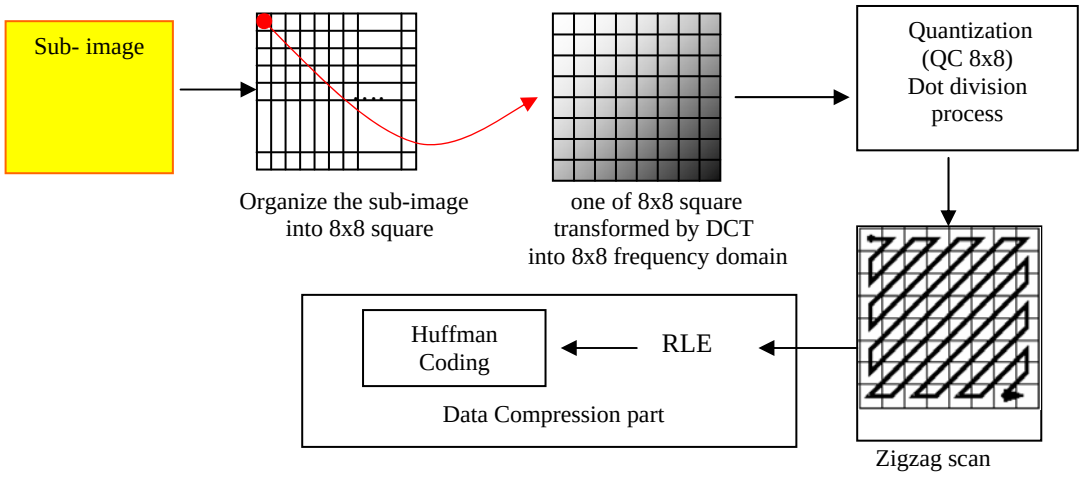

Fig. 2: JPEG technique applied on sub-image $n/2 \times m/2$

2.3. Image Quality and Coding

Before compress an image, we need a method for save image quality. The image quality obtains from the difference between original sub-image (LL) and decoded sub-image (LL_{Decode}), as shown in the following figure:

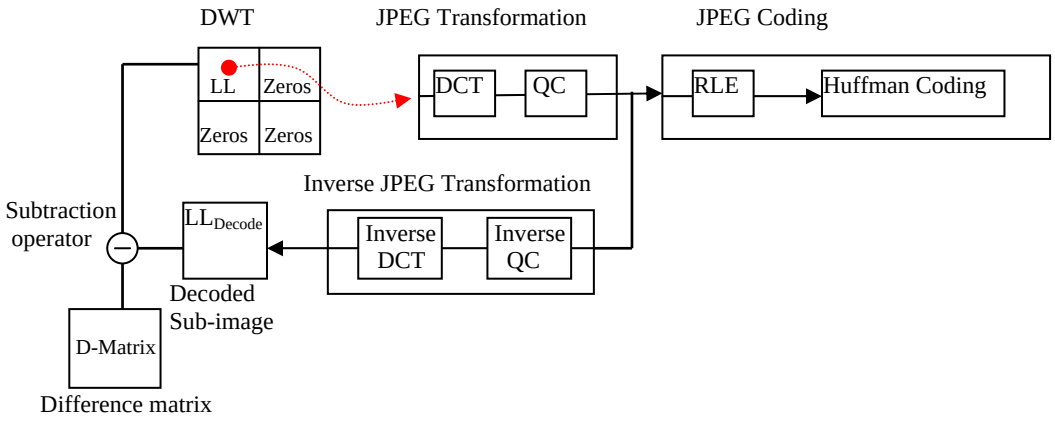

Fig. 3: Difference matrix obtained from subtraction between original LL and LL_{Decode}

The difference matrix in the above figure called *D-Matrix*. To compress D-Matrix, we divide the matrix by a value called *Q-Value*. This value is used to change an image quality for obtains compression ratio, and this value is adjusts by the user to change an image quality.

The range of the "Q-Value" between $\{1 \ldots m\}$, where m represents maximum value in D-Matrix. To compress the D-Matrix, at first we minimize the size for the matrix and then compress each row by RLE and arithmetic coding, the following figure shows compression process for D-Matrix:

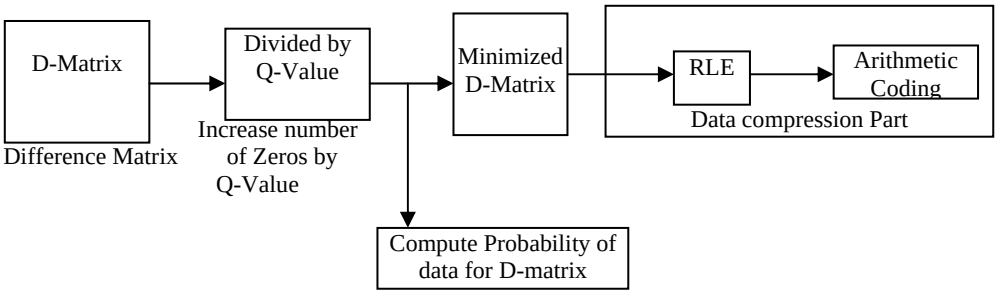

Fig. 4: Minimized D-Matrix steps

The Minimize algorithm reduces size of each row from D-matrix, by the following equation:-

$$m(i,t) = \sum_{j=1}^{4} w(j) * D\_Matrix(i, j+k) \quad (1)$$

Where \ t=1, 2, 3, …, n

k=0, 4, 8, 12, …row size

i=1,2,3,…Column Size

In equation (3) each four data converts to single floating point value. This conversion depends on weight values are generated randomly between {0...1}. The Minimize Algorithm can be illustrates in List-1:

List-1: Minimize Algorithm

```
W=Generate_Random_Wieghts( [0…1] );
P=Compute_Probability_Data();
For (i=1; i<=Column size; ++i)
{ // Begin …
 t=1, k=0;
   while (k <= Row size) Do
   {  // Begin..
         S=0;
         For (j=1; j<=4; ++j)
          S=S+W(j) * D_Matrix(i, k+j);

         Minimized_D_Matrix(i,t)=S;
          t=t+1;
   }  // End..
   k=k+4
}// End…
```

In above List-1, the function "Generate_Random_Wieghts {0..1}" is used to generate four random numbers, and also the function "Compute_Probability_Data()" is used to compute the probability of data for D-Matrix before applying equation(1). The List-1 produced "Minimized D-Matrix" contains floating point values, because the weights values are floating points, and in this research we keep three-digits after decimal point, for obtaining more frequent values in Minimized D-Matrix. The following figure illustrates Minimized D-Matrix:-

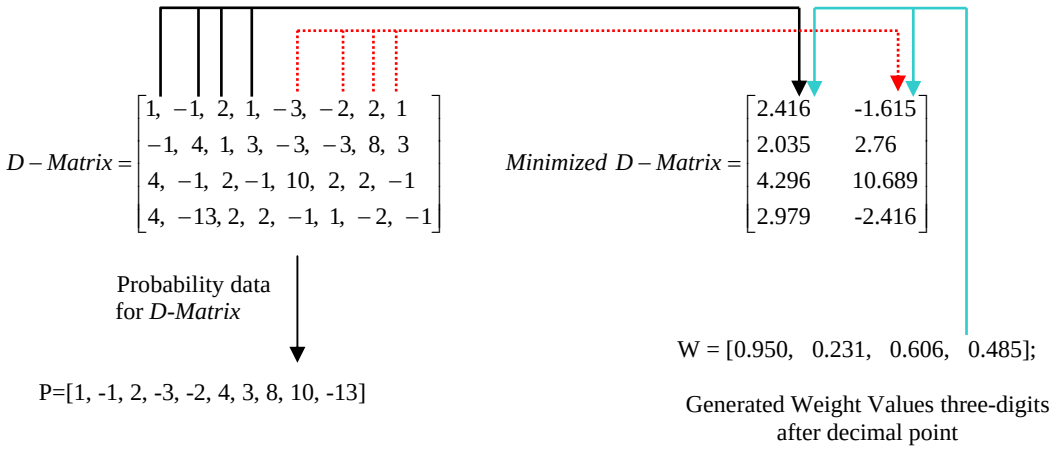

Fig. 5: Example illustrates converting D-Matrix into Minimized D-Matrix

The advantages of "Minimized D-Matrix" to help the RLE and Arithmetic coding for compress data as much as possible. The RLE and Arithmetic Coding used in this research together to compress each row independently from Minimized D-Matrix.

The RLE applied on the stream of a continuous consecutive data, and then replace a number rather than a continuous data. This number is represents number of continuous data, while un-continuous data never changed. For more information about RLE finds in the references[1].

The Arithmetic coding takes stream of data and convert it to single of floating point value. These output values in the range less than one and greater than zero[2], when decoded this value getting exact stream of data. The arithmetic coding need to compute the probability of all data and assign range for each data, the

range value consist from Low and High value, information about Arithmetic coding in the references[2].

Before minimizing the D-matrix we need to compute the probability of data for the D-Matrix. This probability it is used in decompression part (i.e. the probability used by Sequential Search algorithm).

3. Decompression Algorithm (JSS)

Before explaining our decompression algorithm, there is three main points for our compression algorithm, which consisted of:

A- Generate LL sub-band by applying DWT on grayscale image

B-Compress LL (sub-band) by JPEG technique

C-Apply Minimize algorithm on the D-Matrix.

From the above three main points we could understand the compression converts the LL sub-band to stream of bits by Huffman coding and the Minimized D-Matrix converted into another stream of bits by arithmetic coding. In this section we explain the decompression algorithm which it is represents inverse for our compression algorithm, as shown in the following figure:-

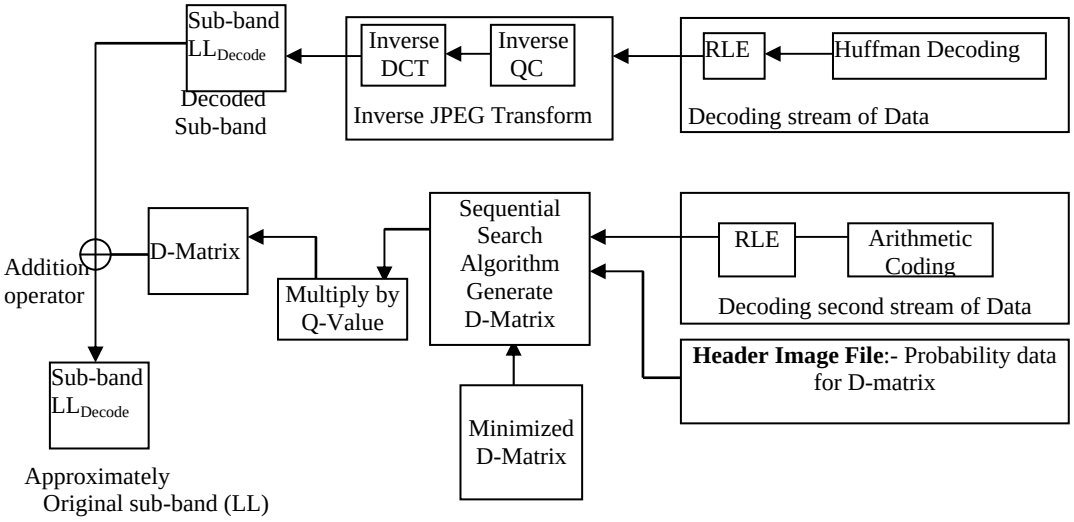

Fig. 6: Our Decompression algorithm (JSS)

In our decompression algorithm the first stream of bits decoded by inverse JPEG to constructs LL_{Decode} (sub-image), but the quality of this sub-image it is low quality (i.e. degraded sub-image). Second stream of bits decoded by arithmetic decoding and inverse RLE, to construct Minimized D-Matrix. Minimized D-Matrix cannot be added with LL_{Decode}. This is because the Minimized D-Matrix contains floating point values, and not compatible with LL_{Decode} to increasing the quality.

In this research we designed a new algorithm used for decoding the Minimized D-Matrix this algorithm called *Sequential Search Algorithm*. The new algorithm used to constructs D-Matrix depending on:-

1) Probabilities of data for the D-Matrix computed previously (See Fig. 4).

2) Minimized D-matrix contains floating point values computed previously (See equation (1)).

4. Sequential Search Algorithm

This technique used in this research for search for the missing data and the data must be inside limited values. We designed this technique for search of D-Matrix values; the probabilities of data for the D-Matrix is represent the limited values. Sequential search algorithm finds the values of D-Matrix by looking for inside the probabilities of D-Matrix.

To illustrate sequential search algorithm, assume we have Minimized D-Matrix as showed previously in Fig. 5 and used sequential search algorithm for return D-Matrix. This algorithm start to initialize four pointers; S1= S2=S3=S4=0; these pointers are refers to array locations, at each iteration the S4 increment by

one. After S4 finished S3 start increment after that S2 and finally S1. This means the S1 is represents outer loop, while S2, S3 and S4 represent inner loop. This operation called Sequential Search. At each iteration our algorithm computes the following equation:

$$C = P(S1)*W(1) + P(S2)*W(2) + P(S3)*W(3) + P(S4)*W(4) \quad (2)$$

In equation(2) the array "P" is represented the probabilities data of D-Matrix, and "W" represents weights values previously used to compress D-Matrix (See Fig. 5). The output "C" compared with each value in Minimized D-Matrix, if equaled, it means; P(S1), P(S2), P(S3) and P(S4) represents original values in the D-Matrix. For example: the first value in Minimized D-Matrix is "2.416" (See Fig. 5), sequential search algorithm used to find the original data inside the probability array "P". after 120 iterations finds the data: P(S1)= 1, P(S2)= -1, P(S3)= 2 and P(S4)= 1 these values represented first four data from D-Matrix. The following figure shows how sequential search algorithm approached to the result:

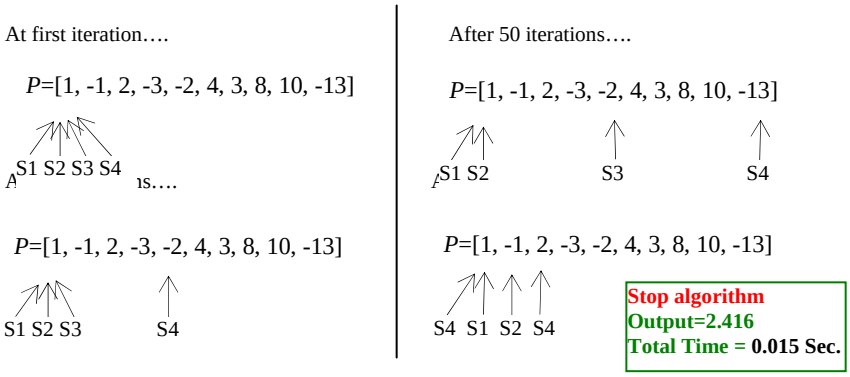

Fig. 7: Samples from some iteration for sequential search algorithm fins 2.416

After sequential search algorithm finds solution, then jump to the second value in Minimized D-Matrix for searching another four values, this process will continue until reached to end of Minimized D-Matrix, and all data for D-Matrix are generated. List-2 illustrates sequential search algorithm applied on the Minimized D-Matrix:-

List-2. Sequential Search Algorithm

```
n → size of "P"
 k=0;
For (i=1;i<=Column Size; ++i )
{// BEGIN
  For (j=1; j<=Row Size; ++j)
  { // BEGIN
    Set_Pointer_to_initilization([S1,S2,S3,S4]);
     Sum=0;
    While (Sum not equal to Minimized D-Matrix (i,j) ) Do
     { // BEGIN
        IF (S1> n) {S1=1}
        IF (S2> n) {S2=1, ++S1}
        IF (S3> n) {S3=1, ++S2}
        IF (S4> n) {S4=1, ++S4} ELSE {++S4};
        Sum=Compute_Equation (W, [S1,S2,S3,S4] );
     } // END
    D-Matrix( i,k )=P(S1), D-Matrix( i,k+1 )=P(S2), D-Matrix( i,k+2 )=P(S3), D-Matrix( i,k+3 )=P(S4);
    k=k+4
  } // END
} // END
```

After generating "D-Matrix", then added with LL_{Decode} for obtains good quality sub-band. Finally the high frequencies values=0 then combined with LL_{Decode}, then using inverse DWT to obtains decompressed image.

5. Computer Results

JSS algorithm tested on two different gray level images as shown in Fig.-8. This algorithm used in laptop, which has the following specification; 1) Processor AMD Turion Dual-Core-2.2GHz, 2) RAM -2GBytes. The language is used for applying our algorithm is MATLAB R2006a.

(a) Original Lena Image with dimensions (500 x 500) Size =244 Kbytes

(b) Original Animal Image with dimensions (960 x 720) Size = 675 Kbytes

Fig. 8 Original Bitmap images used for the compression

JSS algorithm starts with decompose the above two images by single stage DWT type Daubechies Wavelets 'db4' , and ignore all high-frequencies domains (i.e. ignore HL,LH and HH sub-bands). Then apply JPEG technique with reducing image quality by ratio=75%. Also JSS algorithm computes the probability for D-Matrix and stores it as header file, and computes maximum value in the D-Matrix for selecting different Q-Values between 1 and the maximum value, for obtains different compressed size. The Q-Value range for Lena image ={1..48}, and the Q-Value range for Animal image={1..39}.

Finally convert the D-Matrix into Minimized D-Matrix by using the weight values= {0.950, 0.231, 0.606, 0.485}. Table-1 and Table-2 shows the compression ratio for Lena image and Animal image respectively, by JSS algorithm. The compression ratio defined as [1,5]:

$$Compression\ Rate = \frac{Size\ after\ compression}{Size\ before\ compression} \tag{3}$$

Table-1 JSS result for Lena image

Q-Value	Compressed Image size	Compression Ratio
1	16.95 Kbytes	0.0694
5	14.88 Kbytes	0.0609
10	11.88 Kbytes	0.0486
20	6.24 Kbytes	0.0255

Table-2 JSS result for Animal image

Q-Value	Compressed Image size	Compression Ratio
1	47.5 Kbytes	0.0703
5	28.46 Kbytes	0.0421
10	13.78 Kbytes	0.0204
20	12.4 Kbytes	0.0183

JSS decompression algorithm applied on the compressed images as shown in Fig. 9. The decompression algorithm consist from; inverse JPEG technique and sequential search algorithm.

(a) Q-Value=1, PSNR=32.25 dB

(b) Q-Value=1, PSNR=31.74 dB

(c) Q-Value=5, PSNR=32.06 dB

(d) Q-Value=5, PSNR=31.7 dB

(e) Q-Value=10, PSNR=31.47 dB

(f) Q-Value=10, PSNR=30.85 dB

(g) Q-Value=20, PSNR=30.69 dB

(h) Q-Value=20, PSNR=29.84 dB

Fig. 9: (a-h) Decompressed images by JSS according to the Q-Values.

Table-3 Comparison with JPEG and JPEG2000 for Lena image

Algorithm	Parameter	After Compression	PSNR for Decompressed image	Compression Ratio
JSS	Q-Value=1	16.95 Kbytes	PSNR=32.25 dB	0.0694
	Q-Value=5	14.88 Kbytes	PSNR=32.06 dB	0.0609
	Q-Value=10	11.88 Kbytes	PSNR=31.47 dB	0.0486
	Q-Value=20	6.24 Kbytes	PSNR=30.69 dB	0.0255
JPEG	Quality=90%	65.8 Kbytes	PSNR=40.10 dB	0.269
	Quality=75%	37.1 Kbytes	PSNR=36.19 dB	0.152
	Quality =50%	23.7 Kbytes	PSNR=34.00 dB	0.0971
	Quality=25%	14.6 Kbytes	PSNR=31.95 dB	0.0598
JPEG2000	Quality=90%	117 Kbytes	PSNR=49.22 dB	0.479
	Quality=75%	46.3 Kbytes	PSNR=39.14 dB	0.189
	Quality =50%	17.9 Kbytes	PSNR=34.41 dB	0.0733
	Quality=25%	9.7 Kbytes	PSNR=31.9 dB	0.039

Table-4 Comparison with JPEG and JPEG2000 for Animal image

Algorithm	Parameter	After Compression	PSNR for Decompressed image	Compression Ratio
JSS	Q-Value=1	47.5 Kbytes	PSNR=31.74 dB	0.0703
	Q-Value=5	28.46 Kbytes	PSNR=31.70 dB	0.0421
	Q-Value=10	13.78 Kbytes	PSNR=30.85 dB	0.0204
	Q-Value=20	12.4 Kbytes	PSNR=29.84 dB	0.0183
JPEG	Quality=90%	223 Kbytes	PSNR=43.44 dB	0.337
	Quality=75%	131 Kbytes	PSNR=38.15 dB	0.194
	Quality =50%	84.8 Kbytes	PSNR=34.65 dB	0.125
	Quality=25%	65.3 Kbytes	PSNR=32.99 dB	0.0967
JPEG2000	Quality=90%	324 Kbytes	PSNR=51.9 dB	0.48
	Quality=75%	127 Kbytes	PSNR=41.45 dB	0.188
	Quality =50%	48.7 Kbytes	PSNR=33.88 dB	0.0721
	Quality=25%	25.7 Kbytes	PSNR=30.65 dB	0.0380

In above figures, the PSNR (Peak Signal to Noise Ratio) can be calculated very easily and is therefore a very popular quality measurement. It is widely used as a method of comparing the "Quality" of original and Decompressed images [10,13]. PSNR calculated using equation (3) it is measured on a logarithmic scale and is based on the mean squared error (MSE) between an original image and decompressed image, relative to $(255)^2$ (i.e. the square of the highest possible signal value in the image).

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \tag{4}$$

JSS compared with two important image compression methods, JPEG and JPEG2000. JPEG explained previously in section 2.2. JPEG has "Quality" parameter to change the quality of an image [7,6]. While JPEG2000 technique consist of; 1) decompose an image into three-level decomposition. 2) Quantization. 3) Arithmetic decoding. This technique has ability to compress an image as much as possible more than JPEG [8,9,10], also this technique has "Quality" parameter used by the user to select a value between {1%..100%} for compress an image[12,14]. Table-3 and Table-4 shows the comparison JSS with two compression methods for Lena image and Animal image respectively.

6. Conclusion

In this research the image compression algorithm depends on following steps:

1. Reducing image size by the Daubechies DWT. The properties of DWT help our algorithm to get good image quality, and good image compression. JPEG technique associated with our algorithm for obtains higher compression ratio.

2. The Minimize algorithm used to reduce D-Matrix size, this process will help arithmetic coding to compress as much as possible. For this reason our algorithm compresses image size more than JPEG and JPEG2000.

3. The decompressed images has less blurring, because our algorithm used "D-Matrix" with decoded LL(Sub-band). Some regions for the decompressed images by JPEG2000 are blurred[11], while our algorithm eliminates these blurring from decompressed images by using D-Matrix.

Also our algorithm has disadvantages illustrated in the following steps:

1. All the decompressed images has low quality than JPEG and JPEG2000.

2. The sequential search algorithm used to find the D-Matrix, this process takes more execution time for large images than JPEG and JPEG2000.

7. References

[1] K. Sayood. *Introduction to Data Compression*(2nd). San Francisco: Morgan Kaufmann Publishers, Inc. 2000.

[2] Witten, Ian H., Neal, Radford M., and Cleary, John G. Arithmetic Coding for Data Compression. *Communications of the ACM*. 1987, pp. 520-540.

[3] N. Ahmed, T. Natarajan and K. R. Rao. Discrete Cosine Transforms. *IEEE Transactions Computer*. 1974, **C-23**: 90-93.

[4] K. R. Rao and P. Yip. *Discrete Cosine Transform.* Academic Press, 1990.

[5] Rafael C. Gonzalez, Richard E. Woods. *Digital Image Processing.* Addison Wesley publishing company, 2001.

[6] Pennebaker, William B., and Joan L. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, 1993.

[7] S. A. Martucci. Symmetric convolution and the discrete sine and cosine transforms. *IEEE Trans. Sig. Processing*. 1994, **SP-42**: 1038-1051.

[8] Raghuveer, Rao and Ajit. B. Bopadikar. *Introduction to Theory and Applications- Wavelet Transforms.* New Delhi: Pearson Education Asia, 2004.

[9] James. E.Fowler and Beatrice Pesuet-Popescu. An Overview on Wavelets in Source Coding , Communication, and Networks. *EURSIP Journal on Image and Video Processing*, 2007.

[10] D. S. Taubman and M. W. Marcellin. *JPEG2000 Image Compression Fundamentals, Standards and Practice.* The Netherlands: Kluwer Academic, Dordrecht, 2002.

[11] T. Acharya and P. S. Tsai. *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures.* New York: John Wiley & Sons, 2005.

[12] Z. Arnavut and H. Out. Lossless compression of color mapped images with Burrows-Wheeler transformation. *in Proceedings of the IASTED Signal Processing Conference.* 2001, pp. 185-189.

[13] Castelli. V., Robinson j. and Turek j.j., Multiresolution lossless/lossy compression and storage of data for efficient processing thereof, U.S. Patent No. 6,141,445. Octobar, 2000.

[14] S. S. Pradhan, K. Ramchandran. Enhancing analog image transmission systems using digital side information: A new wavelet-based image coding paradigm. *in:Proc. IEEE Data Compression Conf.* (DCC). 2001, pp. 63-72.