An Overview of JPEG 2000 Encoder and Its Simulation on MATLAB

*Niket Shah niket_ec90@yahoo.com
*Pinkesh Shah pinkesh_92@yahoo.com
**Y.N.Trivedi trivediyn@yahoo.co.in
*Student, ** Assistant Professor,
Electronics & Communication,
Institute of Technology, Nirma University

Abstract: With the increasing use of multimedia technologies, image compression requires higher performance as well as new features. In 1996, the JPEG committee began to investigate possibilities for a new still image compression standard to serve current and future applications, this initiative was named JPEG2000. This paper provides a brief history of the JPEG-2000 standardization process, an overview of the standard, and some description of the capabilities provided by the standard as well as supportive simulations of JPEG encoder using MATLAB. Lossless and lossy compression, embedded lossy to lossless coding, progressive transmission by pixel accuracy and by resolution, robustness to the presence of bit-errors and region-of-interest coding, are some representative features.
Keywords -- JPEG, JPEG2000, color image coding, data compression, source coding, subband coding, wavelet transform.

I. 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 lead to the standard known as JPEG (Joint Photographic Experts Group)[7]. This standard uses DCT (Discrete Cosine Transform) for the separation of high and low frequency data from a still image. But when high compression ratio is applied on image when required, retrieved image at receiver end contains artifacts. As digital imagery becomes more commonplace and of higher quality, there is the need to manipulate more and more data. Thus, image compression must not only reduce the necessary storage and bandwidth requirements, but also allow extraction for editing, processing, and targeting particular devices and applications. The JPEG-2000 image compression system has a rate-distortion advantage over the original JPEG. More importantly, it also allows extraction of different resolutions, pixel fidelities, and Regions of Interest (ROI), components, and more, all from a single compressed bit stream. JPEG-2000 has a long list of features, a subset of which are[1]:

- State-of-the-art low bit-rate compression performance
- Progressive transmission by quality, resolution, component, or spatial locality
- Lossy and lossless compression
- Random (spatial) access to the bit stream
- Pan and zoom (with decompression of only a subset of the compressed data)
- Compressed domain processing (e.g., rotation and cropping)
- Region of interest coding by progression
- Limited memory implementations.

In this paper section II and section III gives the basics of JPEG 2000 encoder block while section IV contains simulated results obtained by the authors and section V gives information of future planning of the project.

II. APPLICATIONS-REQUIREMENTS-FEATURES

- Superior low bit-rate performance: This standard should offer performance superior to the current standards at low bit-rates (e.g. below 0.25 bpp for highly detailed gray-scale images).
- Lossless and lossy compression: It is also desired that the standard should have the property of creating embedded bitstream and allow progressive lossy to lossless build-up.
- Region-of-Interest Coding: Often there are parts of an image that are more important than others. This feature allows users to define certain ROI's in the image to be coded and transmitted with better quality and less distortion than the rest of the image.
- Compress once, Decompress many ways: The compress once: Decompress many ways is the new concept brought by the JPEG-2000 standard in field of image compression standards. The JPEG-2000 standard tightly integrates the four modes (sequential, progressive, hierarchical and lossless) of the JPEG standard. The encoder decides maximum quality up to and including lossless. The same way encoder also decides maximum resolution or size. At the decoder quality or size that can be decompressed is up to and maximum decided at the time of encoding. This is also known as SNR and resolution scalability.

III. Functional Block Diagram

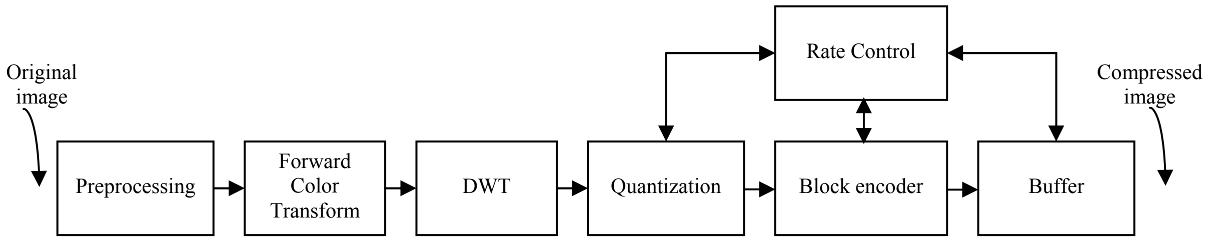

Fig. 3.1: Block diagram of a representative JPEG2000 encoder[2]

The block diagram of the JPEG2000 encoder is illustrated in Figure 3.1. The discrete transform is first applied on the source image data. The transform coefficients are then quantized and entropy coded, before forming the output codestream (bitstream).

A. Preprocessing

The first step in pre-processing is to partition the input image into rectangular and non-overlapping tiles of equal size. Tiling is particularly useful for applications where the amount of available memory is limited compared to the image size. While encoding the dynamic range of the sample data is approximately centered about zero. As the JPEG-2000 accepts both signed and unsigned sample data, the dynamic range of the sample data in case of signed data is $[-2^{B-1}, 2^{B-1}-1]$ and in case of unsigned data is $[0, 2^{B}-1]$ where B is sample width in bits. If the sample values of a component are already signed, then no processing is required. For the unsigned case, the dynamic range of the sample data values of a component is made approximately centered about zero by subtracting 2^{B}-1 from each sample data value. This is done for each component of the image. A grey image contains only one component while RGB color image contains three components i.e. red, green and blue component. As an example, from each sample data value of an 8-bit gray image whose range is [0, 255], 128 is subtracted to make it [-128, 127].

B. Forward Color Transform

This transform operates on all the components red, green and blue at the same time in order to reduce the correlation between them, which leads to improved coding efficiency. The JPEG-2000 codec define two different types of inter-component transforms: the Reversible Color Transform (RCT) and Irreversible Color Transform (ICT). The RCT is integer-to-integer and reversible while ICT is real-to-real and non-reversible. Both the transforms map the image data from RGB to YCrCb color space. As ICT is non- reversible, it is only used in lossy compression. The RCT is reversible and useful in both lossless and lossy compression. After performing inter-component transform, the data in all three components are treated separately.

The forward RCT is given as:

$$V_0(x,y) = \left\lfloor \frac{1}{4}\left(U_0(x,y) + 2U_1(x,y) + U_2(x,y)\right) \right\rfloor$$

$$V_1(x,y) = U_2(x,y) - U_1(x,y)$$

$$V_2(x,y) = U_0(x,y) - U_1(x,y)$$

The forward ICT is given as:

$$\begin{bmatrix} V_0(x,y) \\ V_1(x,y) \\ V_2(x,y) \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} U_0(x,y) \\ U_1(x,y) \\ U_2(x,y) \end{bmatrix}$$

Where, $U_0(x,y)$:Red, $U_1(x,y)$:Green, $U_2(x,y)$: Blue are input components. $V_0(x,y)$:Y[luminance component], $V_1(x,y)$:Cr, $V_2(x,y)$:Cb [Chrominance Components] are output components.

C. Discrete Wavelet Transform (DWT)

The DWT is applied separately to each component. The application of wavelet splits a component into number of frequency bands, called sub bands. Because of statistical properties of the sub band signals, the transformed data can be coded more efficiently than the original sample data.

C.1The 1-D DWT

To perform the forward DWT the standard uses a 1-D sub band decomposition of a 1-D set of samples into low-pass samples and high-pass samples. Low pass samples represent a down sampled low-resolution version of the original set. High-pass samples represent a down sampled residual version of the original set, needed for the perfect reconstruction of the original set from the low-pass set. As shown in the figure 3.2, input signal is x(n) and the low-pass and high-pass filters are designated as $h_0(n)$ and $h_1(n)$. The input signal x(n) is assumed as one row of the image

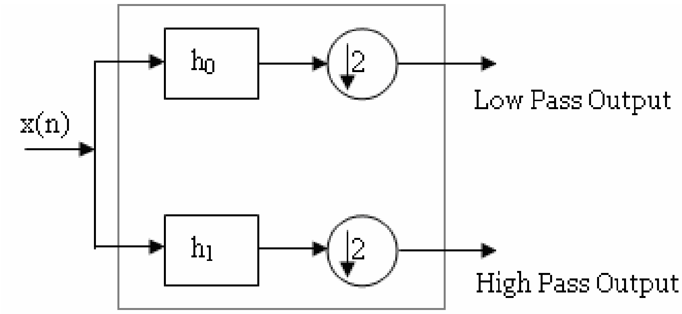

Fig. 3.2: Analysis Filter Bank[6]

The coefficients of the low-pass and high-pass filters are[6]:

$$h_0(n) = \left(-\frac{1}{8}, \frac{1}{4}, \frac{3}{4}, \frac{1}{4}, -\frac{1}{8}\right)$$

$$h_1(n) = \left(-\frac{1}{2}, 1, -\frac{1}{2}\right)$$

This bank is referred as 5x3 filter bank meaning that low-pass filter has five and high-pass filter has three coefficients.

C.2The 2-D DWT:

The 1-D DWT is extended to 2-D by applying the filter bank in separable manner. At each level of wavelet decomposition, each row of 2-D image is first transformed using a 1-D horizontal analysis filter bank. The same filter bank is then applied vertically to each column of the transformed and sub sampled data.

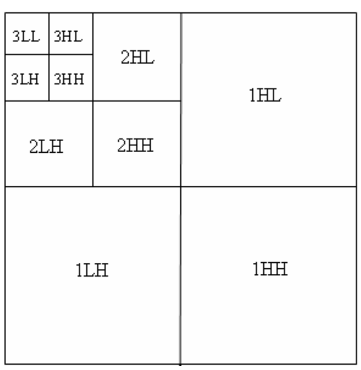

Fig. 3.3: 2-D 3-level wavelet decomposition[2]

1. Horizontally & Vertically low-pass band (LL) subband
2. Horizontally low-pass & vertically high-pass (HL) subband
3. Horizontally high-pass & vertically low -pass (LH) subband
4. Horizontally & Vertically high-pass band (HH) subband

D. QUANTIZATION

After transformation, all coefficients are quantized. Scalar quantization is used in Part I of the standard. Quantization is the process by which the coefficients are reduced in precision. This operation is lossy, unless the quantization step is 1 and the coefficients are integers, as produced by the reversible integer 5/3 wavelet. Each of the transform coefficients $y_b(u,v)$ of the subband b is quantized to the value $q_b(u,v)$ according to the formula[2]:

$$q_b(u,v) = \text{sign}(y_b(u,v)) \left\lfloor \frac{|y_b(u,v)|}{\Delta_b} \right\rfloor$$

Where, Δb: Step size consists of 11-bit mantissa μb and a 5 bit exponent εb, according to the relationship given below:

$$\Delta b = 2-\varepsilon b(1+\mu b/211)$$

When reversible 5x3 filter bank is used, the quantization step-size Δb is set to 1 by choosing the εb = 0 and μb = 0. It is same as bypassing the quantization process in the encoder.

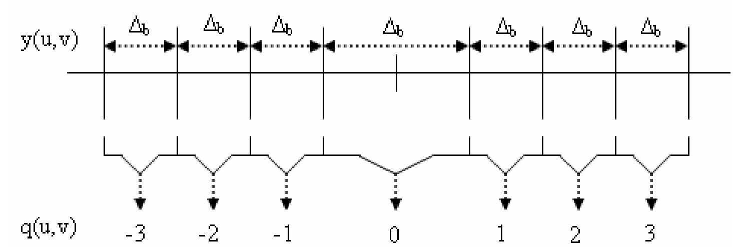

Fig. 3.4: Uniform quantizer[2]

E. BLOCK CODING

E.1Entropy Coding

After quantization, each subband is divided into rectangular blocks, i.e. non-overlapping rectangles. Three spatially consistent rectangles (one from each subband at each resolution level) comprise a packet partition location or precinct. Each packet partition location is further divided into non-overlapping rectangles, called "code-blocks", which form the input to the entropy coder[1].

E.2 Bit-plane Coding

The individual bit-planes of the coefficients in a code-block are coded within three coding passes. Each of these coding passes collects contextual information about the bit-plane data. An arithmetic coder uses this contextual information and its internal state to decode a compressed bit-stream. Different termination mechanisms allow different levels of independent extraction of this coding pass data. The coded data of each code-block is distributed across one or more layers in the codestream. Each layer consists of a number of consecutive bit-plane coding passes from each code-block in the tile, including all subbands of all components for that tile[4]. The number of coding passes in the layer may vary from code-block to code-block and may be as little as zero for any or all code-blocks. Each layer successively and monotonically improves the image quality, so that the decoder is able to decode the codeblock contributions contained in each layer in sequence.

The data representing a specific tile, layer, component, resolution and precinct appears in the codestream in a contiguous segment called a packet. The data in a packet is ordered in such a way that the contribution from the LL, HL, LH and HH subbands appear in that order. Within each subband, the code-block contributions appear in raster order, confined to the bounds established by the relevant precinct.

Each bit-plane of a code-block is scanned in a particular order. Starting from the top left, the first four bits of the first column are scanned. Then the first four bits of the second column, until the width of the code-block is covered. Then the second four bits of the first column are scanned and so on. A similar vertical scan is continued for any leftover rows on the lowest code-blocks in the sub band.

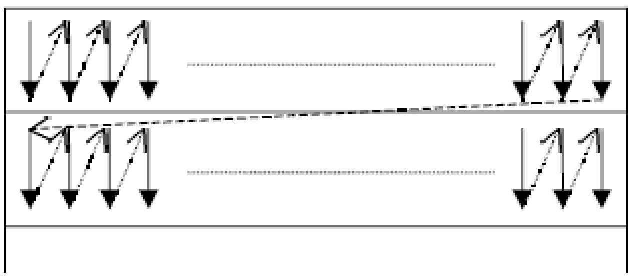

Fig.3.5 Scanning Pattern of Bit Plane Coding[2]

Code-blocks are then coded a bit-plane at a time starting from the most significant bit-plane with a non-zero element to the least significant bit-plane. For each bit-plane in a code-block, a special code-block scan pattern is used for each of three coding passes. Each coefficient bit in the bit-plane is coded in only one of the three coding passes. The three coding passes are: *significance propagation*, *magnitude refinement*, and *cleanup[3]*.

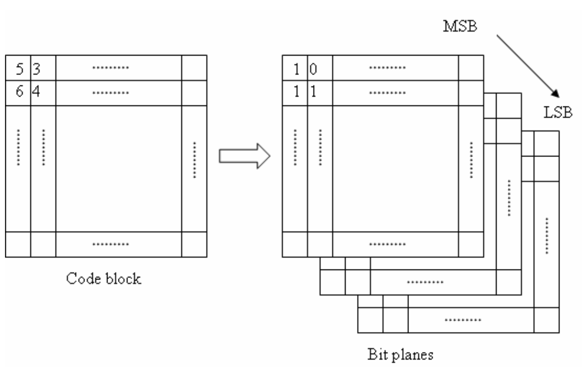

Fig. 3.6 Bit Planes of a code block

Each coefficient in a code block is assigned a binary state variable called its *significance state* variable. The 0 value of variable indicates that coefficient is insignificant and 1 value indicates that coefficient is significant. Initially when encoding of the block starts it is 0 (insignificant) and it changes to 1 (significant) when the first non-zero bit of the coefficient is found. During the first pass known as *significance propagation pass*, the coefficients having highest probability to become significant as determined by their neighbors are coded. The second pass known as *magnitude refinement pass* codes the refinement bits of the significant coefficients. The bits of a coefficient after first non-zero bits from MSB to LSB order are called refinement bits. During the third pass known as *cleanups pass* all the remaining coefficients having lowest probability to become significant are coded.

E.3 Adaptive Arithmetic coding

As described above, all coding is done using context dependent binary arithmetic decoding (the MQ coder is adopted in JPEG2000). The recursive probability interval subdivision of Elias coding is the basis for the binary *arithmetic coding* process. With each binary decision, the current probability interval is subdivided into two sub-intervals, and the codestream is modified (if necessary) so that points to the base (the lower bound) of the probability sub-interval assigned to the symbol, which occurred. Since the coding process involves addition of binary fractions rather than concatenation of integer codewords, the more probable binary decisions can often be coded at a cost of much less than one bit per decision[1].

E.4 Layered Bit-Stream Formation

In JPEG2000, the bit-stream is organized as a succession of layers. Each layer contains the additional contributions from each code-block (some contributions may be empty and in general the number of bits contributed by a code block). For each code-block, a separate bit-stream is generated. No information from other blocks is utilized during the generation of the bitstream for a particular block. Truncation points to each code block are allocated using rate distortion optimization[2].

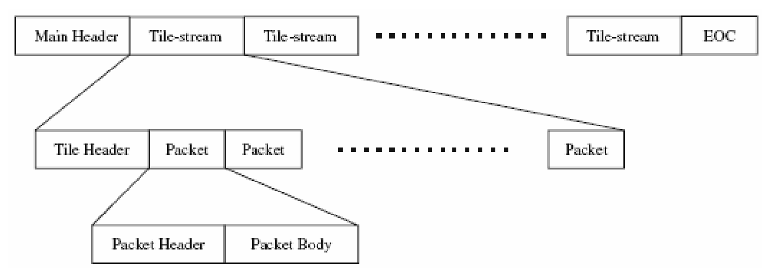

Fig. 3.7 Embedded Bit Stream[5]

IV. PROJECT PROGRESS

The authors have successfully implemented Pre-processing, Forward Color Transform, DWT, Quantization, Entropy Coding, Bit Plane Coding as a part of Embedded Bit Stream formation followed by the Arithmetic Coder block of the JPEG 2000 Encoder. Here authors have used RCT in FCT block as it is supporting one of the features that is lossy & lossless compression. The use of 5X3 filter over 9X7 filter in DWT block as it is reversible and reduces the complexity at both ends. The Embedded bit stream with multiple truncation points helps receiver to decide how much data is it required to reconstruct an image and can truncate stream any of truncation points.

SIMULATION RESULTS:

1. Pre-processing:

Fig.4.1: Original Image
192 KB

Fig 4.2: Signed Image

2. Forward Color Transform

Fig 4.3: Y (Luminance Component)

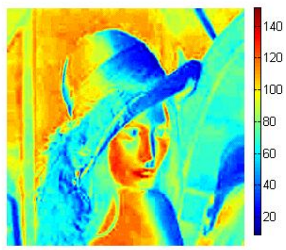

Fig 4.4: U (R-G)

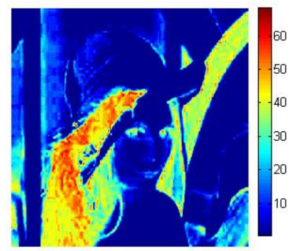

Fig 4.5:V (B-G)

.3. Discrete Wavelet Transform (DWT)

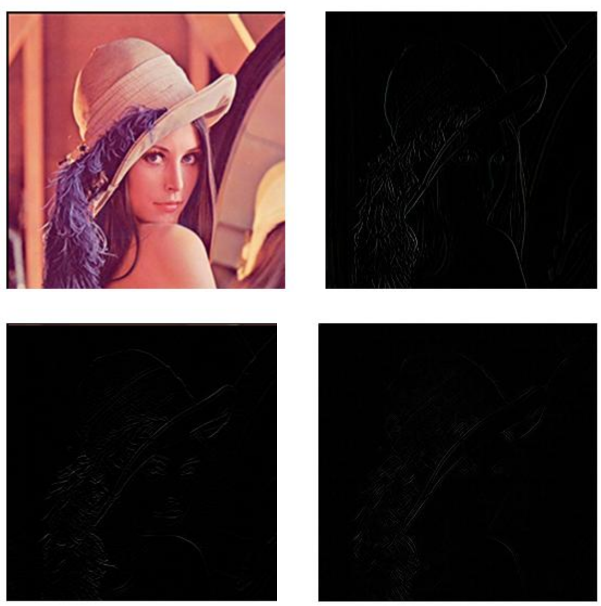

Fig 4.6: 2-D level 1 DWT
(Best View at 300% zoom)
Size of LL subband : 102 KB)

V. Future Scope

The JPEG 2000 standard has been divided in 6 parts, mentioned below:

Part	Title	Purpose
1	Core coding system	Specify the core(minimum functionality) codec for the JPEG-2000 family of standards
2	Extensions	Specifies additional functionalities that are useful in some applications but need not be supported by all the codecs.
3	Motion JPEG-2000	Specifies extensions to JPEG-2000 for intraframe style video compression
4	Conformance testing	Specifies the procedures to be employed for compliance testing
5	Reference software	Provides sample software implementations of the standard to serve as a guide for implementers
6	Compound image file format	Specifies compound image file format for document and fax applications

Having completed the core coding system, the authors have started working on ROI and would like to implement motion JPEG –2000.

Acknowledgement

The work is supported and sponsored by Institute of Technology, Nirma University and the authors are thankful to the Institute for providing required resources during the entire project.

References

[1] THE JPEG2000 STILL IMAGE CODING SYSTEM: AN OVERVIEW by:Charilaos Christopoulos Athanassios Skodras, and Touradj Ebrahimi,Published in IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp. 1103-1110, November 2000.

[2] AN OVERVIEW OF THE JPEG2000 STILL IMAGE COMPRESSION STANDARD:by Majid Rabbani,Rajan Joshi,published in ELSEVIER ,Signal Processing Image Communication ,pp 3-10,15-20

[3] A NEW ALGORITHM FOR THE EMBEDDING OF A PREDICTION MECHANISM INTO THE JPEG2000 CODING CHAIN by Marco Aguzzi., Maria Grazia Albanesi, Marco Martelli ,published in ELSEVIER Pattern Recognition 39 (2006) 1492-1494

[4] AN OVERVIEW OF JPEG-2000 by Michael W. Marcellin1, Michael J. Gormish2, Ali Bilgin1, Martin P. Boliek, This paper appeared in *Proc. of IEEE Data Compression Conference*, pp. 530-535, 2000. When JPEG2000 Part I went from CD to FCD the term "packet partition location" was changed to "precinct."

[5] D. Saloman, *Data Compression – The Complete Reference*. New York City, NY: Springer, 2004.

[6] Digital Signal Processing by Steve White for IIR & FIR Digital Filter.

[7] A book on INTRODUCTION TO DATA COMPRESSION by K. Sayood.