Block Based Discrete Wavelet Transform for Image Compression

K. Bhanu Rekha¹, S. Ramachandran²
¹ Sir MVIT, Bangalore
² SJBIT, Bangalore

I. INTRODUCTION

The two-dimensional Discrete Wavelet Transform (2D-DWT) is nowadays established as a key operation in image processing. In the area of image compression, the 2D DWT has clearly prevailed against its predecessor, the 2D Discrete Cosine Transform. This is mainly because it achieves higher compression ratios owing to the sub-band decomposition, while it eliminates the ‘blocking’ artefacts that deprive the reconstructed image of the desired smoothness and continuity. The high algorithmic performance of the 2D DWT in image compression justifies its use as the kernel of the JPEG-2000 still image compression standard. The disadvantage of DWT is that it requires more processing power. DCT overcomes this disadvantage since it needs less processing power, but it gives less compression ratio for the same quality [1-4].

For the execution of the multilevel 2D DWT, several computation schedules based on various input traversal patterns have been proposed. Among these, the most commonly used in practical designs are: the Direct and the Line Based methods [5, 6].

In this paper, a new non-overlapping block based technique is proposed which ensures faster processing when compared to the Direct and the Line Based methods without compromising on the quality of the reconstructed image. Test images of varying sizes ranging from 150x250 pixels to 600x912 pixels are divided into various sizes from 8x8 pixel blocks to 64x64 pixel macro-blocks in order to arrive at the optimum block size in terms of processing speed and quality. 2D-DWT is applied to each block, Quantized and Reconstructed. Simulation results show the impact of this structure in terms of image quality metrics.

The rest of the paper is organized as follows. Section II briefly reviews the lifting scheme. Section III describes the Direct and the Line Based 2D-DWT Structures followed by a description of the Proposed Technique in Section IV. The next section presents the experimental results and the conclusion is presented in the last section.

II. LIFTING SCHEME

Wim Sweldens developed a lifting scheme for the construction of bi-orthogonal wavelets. The main feature of the lifting scheme is that all constructions are derived in the spatial domain [7-9]. Lifting scheme is a simple and an efficient algorithm to calculate wavelet transforms as a sequence of lifting steps. Constructing wavelets using lifting scheme comprises three steps:
1. **Split step:** The original signal, $X(n)$, is split into odd and even samples.

2. **Lifting step:** This step is executed as N sub steps depending on the type of the filter, where the odd and even samples are filtered by the prediction and update filters, $p(z)$ and $u(z)$.

3. **Normalization or Scaling step:** After $N$ lifting steps, scaling coefficients $K$ and $1/K$ are applied respectively to the odd and even samples in order to obtain the low pass sub band $Y_L(i)$ and the high-pass sub band $Y_H(i)$.

Fig.1 shows the lifting scheme of wavelet filter computing one dimension signal. The inverse transform could easily be found by exchanging the sign of the predict step and the update step and applying all operations in reverse order as shown in Fig. 2. Lifting based

\[ P(z) = \prod_{i=1}^{3} \left( \begin{array}{cc} 1 & s_i(z) \\ 0 & 1 \end{array} \right) \left( \begin{array}{cc} 1 & 0 \\ t_i(z) & 1 \end{array} \right) \left( \begin{array}{cc} k & 0 \\ 0 & 1/k \end{array} \right) \]

inverse transform (IDWT) is simple and involves the reversal of the order of operations in DWT. Therefore the same resources can be reused to define a general programmable architecture for forward and inverse DWT [10-15].

Lifting scheme implements a filter bank as a multiplication of upper and lower triangular matrices, where each matrix constitutes a lifting step. For the 9/7 wavelet, four lifting steps and one scaling can be used; its polyphase analysis filter banks $P(z)$ can be written as

\[ P(z) = \prod_{i=1}^{3} \left( \begin{array}{cc} 1 & s_i(z) \\ 0 & 1 \end{array} \right) \left( \begin{array}{cc} 1 & 0 \\ t_i(z) & 1 \end{array} \right) \left( \begin{array}{cc} k & 0 \\ 0 & 1/k \end{array} \right) \]

where $s_1(z)=\alpha(1+z^{-1})$, $s_2(z)=\gamma(1+z^{-1})$, $t_1(z)=\beta(1+z^{-1})$ and $t_2(z)=\delta(1+z^{-1})$.

The parameters $\alpha$, $\gamma$, $\beta$ and $\delta$ are symmetric coefficients while $K$ and $1/K$ are scaling factors.

The lifting steps lead to the following equations:

\[ \text{Predict P1: } d_i^{1} = \alpha(x_{2i} + x_{2i+2}) + x_{2i+1} \]
\[ \text{Update U1: } a_i^{1} = \beta(d_i^{1} + a_{i-1}^{1}) + x_{2i} \]
\[ \text{Predict P2: } d_i^{2} = \gamma(a_i^{1} + a_{i-1}^{1}) + d_i^{1} \]
\[ \text{Update U2: } a_i^{2} = \delta(d_i^{2} + d_{i-1}^{2}) + a_i^{1} \]
\[ \text{Scale G1: } a_i = K \cdot a_i^{2} \]
\[ \text{Scale G2: } d_i = d_i^{2} / K \]

The original data to be filtered is denoted by $x_n$, and the outputs are the approximation coefficients $a_i$ and detail coefficients $d_i$. The superscripts on intermediate values show the lifting step number.
III. D DWT STRUCTURES

In this section, we will review 2 different structures for 2-D DWT, the direct method and the line based method.

Direct Method
In this method, the input image is stored in an external memory; scanned row by row and 1-D DWT is applied. The primary outputs are wavelet coefficients computed row-wise and are stored in the external memory. Thereafter, they are input to the 1-D DWT block column-wise. The outputs of the 1-D DWT block are the desired 2-D DWT coefficients of the input Image and are stored back in the external memory as shown in Fig. 3. If the computation of coefficients for one more decomposition level is needed, this procedure must be repeated for the LL part of the previous level, whose size is a quarter of the input image size. This routine will be repeated for higher levels. Although the method is simple, the resulting latency and the number of external memory accesses are unduly large [16-19].

Line-based method
The line-based method can be implemented by a 2-D DWT block. In this method, only internal memory is used to compute one level DWT for both the row and column directions. Therefore, there is no external memory access during the computation of one level 2-D DWT. The required internal memory is the sum of the data memory and the temporal memory for each line [20-22]. Once the DWT coefficients are computed, they are stored in the external memory. This line based structure is shown in Fig. 4.

IV. PROPOSED BLOCK BASED METHOD

The block diagram of the proposed structure is presented in Fig. 5. An image is transformed from RGB to Gray component. Then, the component is partitioned into non-overlapping 16x16 pixel Macro Blocks to form an array of “Macro-components”. In turn, every Macro-component is wavelet transformed into 4 sub-bands for every level of the wavelet transformation. Then, each Macro Block is Quantized and transformed back using Inverse Wavelet transform. The Image is reconstructed by merging all the Macro Blocks to form the Reconstructed Image.

The idea of Macro Block serves the same purpose as the partition of 8x8 blocks in the JPEG standard. All Macro Blocks are handled independently. Therefore, it reduces memory requirements as the entire bit-stream is not needed to process a portion of the image. Macro Block also makes extraction of a region of the image for editing easier by specifying the indexes of the corresponding Macro Block. All Macro Blocks are square and have same dimensions.

Each Macro Block is wavelet transformed into N_L decomposition level. Thereafter, N_L+1 numbers of different resolutions are provided for this macro block. We denote the resolutions by an index \( r \), ranging from 0 to N_L. The lowest resolution is \( r = 0 \), which is represented by the N_L numbers of LL sub-band coefficients while \( r = N_L \) is the highest resolution, which is reconstructed from the 1LL, 1HL, 1LH and 1HH sub-bands. For a specific resolution, \( r \) not equal to 0, it is reconstructed from nLL, nHL, nLH, and nHH sub-bands, where n is equal to N_L - r +1.
For a 5-tap or 3-tap wavelet transformation, no quantization is used to reduce the precision of the coefficients. That means the quantization step is one and the coefficients have integer values. On the other hand, for the 9-tap/7-tap wavelet transformation, each sub-band from a Macro Block can have its own quantization step value. The quantization step, $\Delta_b$, for sub-band b is specified by the following equation:

$$\Delta_b = 2^{R_b - e_b} \left( 1 + \frac{H_b}{2^{11}} \right)$$

(3)

where $R_b$ is the nominal dynamic range for sub-band b. It is the sum of the number of bits that are used to represent the original source image Macro Block. The exponent, mantissa pairs ($e_b$, $\mu_b$) are either applied for all sub-bands or for the LL sub-band only. In the latter case, the exponent/mantissa pairs ($e_b$, $\mu_b$) are determined from the exponent/mantissa pair ($e_0$, $\mu_0$) corresponding to the LL sub-band, according to the following equation:

$$(e_b, \mu_b) = (e_0 + nsd_b - nsd_0, \mu_0)$$

(4)

where $nsd_b$ denotes the number of sub-band decomposition levels from the original image Macro block to the sub-band b. Therefore, $E_b$ for the lower frequency sub-bands tend to be larger and render the quantization steps for these sub-bands to be smaller. Therefore, less distortion results from the quantization error. Each of the wavelet transformed coefficients, $a_b(u,v)$ of the sub-band b is quantized into $q_b(u,v)$ according to the following equation:

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

(5)

The decoding procedures perform only the inverse functions of the Wavelet Transform. The Macro Blocks are merged to reconstruct the image.

V. EXPERIMENTAL RESULTS

The quality of the reconstructed image is computed using a Mean Square Error. This requires the original and the reconstructed image of the same size to evaluate the error between them. The quality measure PSNR is expressed in decibel scale (dB). High value of PSNR indicates a high quality of image [23, 24]. It is defined using Mean Square Error (MSE). Lower value of MSE results in higher value of PSNR. The PSNR is expressed as:

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

(6)

where

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

(7)

$I (i,j)$ : Original Image

$K (i,j)$ : Reconstructed Image and

$m \times n$ : Total number of Pixels in the Original image.

The Block based scheme was implemented in Matlab. As examples, four “standard” test images were chosen for the experiment. All these test images have different dimensions. As per the requirements of JPEG2000 standard, one can select blocks of size 4x4 pixels or more. Smaller the block size, smaller will be the computation time required for processing transforms DWT and IDWT. However, if transformed coefficients are quantized, artefacts are likely to manifest along the edges since there is no correlation between two adjacent blocks transformed. Therefore, a higher sized block would be optimum in terms of computation time as well as minimizing artefacts. For these reasons, macro block of size 16x16 pixels are chosen in the present work.
Table 1 Comparison of Quality of Reconstructed Images Using Different Methods and the Proposed Method

Level of Decomposition: 2D-DWT

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena1.tif</td>
<td>53.0</td>
<td>0.2</td>
<td>53.0</td>
<td>0.2</td>
<td>34.9</td>
<td>20.9</td>
<td>51.9</td>
<td>0.27</td>
</tr>
<tr>
<td>Pisa1.tif</td>
<td>52.3</td>
<td>0.3</td>
<td>52.3</td>
<td>0.3</td>
<td>36.9</td>
<td>13.0</td>
<td>52.1</td>
<td>0.26</td>
</tr>
<tr>
<td>clkgeneva1.tif</td>
<td>51.8</td>
<td>0.4</td>
<td>51.8</td>
<td>0.4</td>
<td>33.5</td>
<td>28.4</td>
<td>51.7</td>
<td>0.28</td>
</tr>
<tr>
<td>titanic51.tif</td>
<td>53.0</td>
<td>0.3</td>
<td>53.0</td>
<td>0.3</td>
<td>41.9</td>
<td>4.1</td>
<td>52.3</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 1 presents the quality measure of the reconstructed images obtained by the proposed method as well as that for other methods. It may be seen that the reconstructed image quality got by the proposed method is comparable with other methods such as the Direct and the Line based methods. The proposed block processing scheme improves the on-chip memory and frame buffer utilization to suit hardware implementation of 2-D DWT/IDWT structure. The reconstructed images by the block based method are presented for various test images in Fig. 7. It may be observed that there are no blocking artefacts.
VI. CONCLUSION

A block based 2D-DWT/IDWT has been implemented in Matlab by applying the transforms on macro blocks in order to minimize artefacts as well as to speed up hardware implementation. This method also helps in the reduction of on-chip memory requirements of the implementation. The original image is divided into non-overlapping macro-blocks, and as a result it offers a convenient approach for compression using entropy coding where an image has to be divided into code blocks. The proposed Block Based method is more favourable in terms of less computation time and less memory utilization when compared to other methods such as the Line Based and Direct methods. The quality of the reconstructed images obtained by the proposed method is comparable with other methods. Currently, RTL Verilog coding is in progress for the block based DWT/IDWT.

REFERENCE


<table>
<thead>
<tr>
<th>Input Image</th>
<th>PSNR for Direct DWT in dB[22]</th>
<th>MSE for Direct DWT</th>
<th>PSNR for Line based DWT[23]</th>
<th>MSE for Line based DWT</th>
<th>PSNR for Block Based DCT in dB</th>
<th>MSE for Block Based DCT</th>
<th>PSNR Proposed Method 16x16 pixels</th>
<th>MSE Proposed Method 16x16 pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena1.tiff</td>
<td>53.0</td>
<td>0.2</td>
<td>53.0</td>
<td>0.2</td>
<td>34.9</td>
<td>20.9</td>
<td>51.9</td>
<td>0.27</td>
</tr>
<tr>
<td>Pisa1.tiff</td>
<td>52.3</td>
<td>0.3</td>
<td>52.3</td>
<td>0.3</td>
<td>36.9</td>
<td>13.0</td>
<td>52.1</td>
<td>0.26</td>
</tr>
<tr>
<td>clkgeneva1.tiff</td>
<td>51.8</td>
<td>0.4</td>
<td>51.8</td>
<td>0.4</td>
<td>33.5</td>
<td>28.4</td>
<td>51.7</td>
<td>0.28</td>
</tr>
<tr>
<td>titanic51.tiff</td>
<td>53.0</td>
<td>0.3</td>
<td>53.0</td>
<td>0.3</td>
<td>41.9</td>
<td>4.1</td>
<td>52.3</td>
<td>0.25</td>
</tr>
</tbody>
</table>