An Improved Concurrent Multi-Threaded Wavelet Transform Based Image Compression and Transmission Over Internet

Kamrul Hasan Talukder and Koichi Harada

Abstract—**In this paper, a wavelet-based image transmission scheme has been proposed. The proposed scheme has used discrete wavelet transform to transform a digital image from spatial domain into frequency domain. For wavelet transformation phase, Haar wavelet transformation has been used. However, it is computationally rigorous. Use of concurrent computing has significantly reduced computation time overhead as well as the transmission time up to 80% to 82% as compared to the existing sequential discrete wavelet based transmission. The experimental result shows that the proposed scheme maintains the accuracy of reconstructed image.**

Index Terms—**Wavelet transform, concurrent computing, progressive image transformation, haar wavelet transformation.**

I. Introduction

The internet has become an obligatory component of today's transacting world. However, the internet is not always able to quickly transfer the image data of the web page. These images not only take huge time to reach the destination but also often completely slow down or block other traffic on the network. This demands the development and improvement of the compression approaches of the image data to be transferred.

The initial breakthrough in the compression of one-dimensional signals [1] was easily extended to the image domain by concatenating image rows or columns into a single stream. Some techniques such as Shannon Fano coding [2] and Huffman coding [3]-[8] use redundancy-reduction mechanisms which result in shorter codes for more frequently appearing samples. It is necessary to scan the data samples in order to calculate their probabilities of occurrence and create an exact code. Run-length coding [9] is another redundancy-reduction coding method where in a scan-line each run of symbols is coded as a pair that specifies the symbol and the length of the run. While most redundancy-reduction methods are lossless, other arbitrarily lossy coding methods have achieved higher levels of compression. Transform coding [10], subband coding [11] -[14], vector quantization [15]-[17] and predictive coding [18] [19]-[27] are among the ones that have achieved high levels of lossy compression. The major transform coding techniques include cosine/sine [28], Fourier [29], Hadamard [30], Haar [30], [10], slant [10] and principal-component (Karhunen-Loeve) transforms.

The browsing image on the internet has become a common activity as a means of communication with the emergence of World Wide Web. However, it often becomes frustrating due to low speed internet connection. As there are several phases like encoding and decoding of image while transmitting over the internet, it takes much time to transmit the whole of a large sized image. Fast downloading of high quality images is of increasing importance in many applications including photo agencies, geographical information systems, medical databases, distance learning etc. So, requiring long time to complete the whole image transmission process is a big problem. Sometimes, the image presented is not the actual one that the user wants. So, it is necessary to allow the user to have a preview before the whole transmission is completed so that he is allowed to save time by aborting the transmission if required.

Here a better technique has been proposed to send the core portion of the original image and then refining it progressively. The technique divides the original image into several parts. The sender transmits the image to the receiver via different stages, and the receiver has to combine the data from all stages to recover the image from initially blurred to progressively clear. Thus using this method, user can get a glance of the image earlier and can decide whether it is the correct image or abort it. In case of large sized images, the encoding and decoding processes for transmitting image also takes mentionable time, because the traditional way followed for these are computationally rigorous.

The target of our method is to encode the original image into a data stream from which image can be reconstructed efficiently. Moreover, it provides a fast glance of the image to the user. The encoding and decoding processes have been manipulated in such a way that it fulfills the criteria. This paper has proposed a scheme where concurrent computing has been used in both encoding and decoding phase to reduce the computation time. Also concurrent technique is applied during the data stream transmission. Though concurrent computing technique has been applied in both phases the quality of reconstructed image has been ensured. Haar Discrete Wavelet Transformation has been selected and applied in order to transform the digital image into the frequency domain to decompose the spatial pixels into frequency coefficients.

Manuscript received March 25, 2012; revised April 28. This work was supported by Japan Society for the Promotion of Sciences (JSPS).

Authors are with the Department of Information Engineering, Hiroshima University, Higashi-Hiroshima in Japan (e-mail: kht@ hiroshima-u.ac.jp, hrd@ hiroshima-u.ac.jp).

II. Haar Wavelet Technique

To understand how wavelets work, let us start with a simple example. Assume we have a 1D image with a resolution of four pixels, having values [9 7 3 5]. Haar wavelet basis can be used to represent this image by computing a wavelet transform. To do this, first the average the pixels together, pairwise, is calculated to get the new lower resolution image with pixel values [8 4]. Clearly, some information is lost in this averaging process. We need to store some *detail coefficients* to recover the original four pixel values from the two averaged values. In our example, 1 is chosen for the first detail coefficient, since the average computed is 1 less than 9 and 1 more than 7. This single number is used to recover the first two pixels of our original four-pixel image. Similarly, the second detail coefficient is -1, since 4 + (-1) = 3 and 4 - (-1) = 5. Thus, the original image is decomposed into a lower resolution (two-pixel) version and a pair of detail coefficients. Repeating this process recursively on the averages gives the full decomposition shown in Table I:

TABLE I: Decomposition To Lower Resolution

Resolution	Averages	Detail Coefficients
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

Thus, for the one-dimensional Haar basis, the wavelet transform of the original four-pixel image is given by [6 2 1 -1]. We call the way used to compute the wavelet transform by recursively averaging and differencing coefficients, *filter bank*. We can reconstruct the image to any resolution by recursively adding and subtracting the detail coefficients from the lower resolution versions.

Compression of 2D image with Haar Wavelet Technique

It has been shown in previous section how one dimensional image can be treated as sequences of coefficients. Alternatively, we can think of images as piecewise constant functions on the half-open interval [0, 1). To do so, the concept of a *vector space* is used. A one-pixel image is just a function that is constant over the entire interval [0, 1). Let V^0 be the vector space of all these functions. A two pixel image has two constant pieces over the intervals [0, 1/2) and [1/2, 1). We call the space containing all these functions V^1. If we continue in this manner, the space V^j will include all piecewise-constant functions defined on the interval [0, 1) with constant pieces over each of 2^j equal subintervals. We can now think of every one-dimensional image with 2^j pixels as an element, or vector, in V^j. Note that because these vectors are all functions defined on the unit interval, every vector in V^j *is* also contained in V^{j+1}. For example, we can always describe a piecewise constant function with two intervals as a piecewise-constant function with four intervals, with each interval in the first function corresponding to a pair of intervals in the second. Thus, the spaces V^j are nested; that is, $V^0 \subset V^1 \subset V^2 \subset$ …… This nested set of spaces V^j is a necessary ingredient for the mathematical theory of multiresolution analysis. It guarantees that every member of V^0 can be represented exactly as a member of higher resolution space V^1. The converse, however, is not true: not every function $G(x)$ in V^1 can be represented exactly in lower resolution space V^0; in general there is some lost detail.

Now we define a basis for each vector space V^j. The basis functions for the spaces V^j are called *scaling functions*, and are usually denoted by the symbol ϕ. A simple basis for V^j is given by the set of scaled and translated box functions:

$$\phi_i^j(x) := \phi(2^j x - i) \quad i = 0, 1, 2\ldots..2^j - 1 \text{ where}$$

$$\phi(x) := \begin{cases} 1 & \text{for } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

The wavelets corresponding to the box basis are known as the *Haar wavelets*, given by-

$$\Psi_i^j(x) := \Psi(2^j x - i) \quad i = 0, 1, 2\ldots..2^j - 1 \text{ where}$$

$$\Psi(x) := \begin{cases} 1 & \text{for } 0 \le x < 1/2 \\ 1 & \text{for } 1/2 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

Thus, the DWT for an image as a 2D signal will be obtained from 1D DWT. We get the scaling function and wavelet function for 2D by multiplying two 1D functions. The scaling function is obtained by multiplying two 1D scaling functions: $\phi(x,y)=\phi(x)\phi(y)$. The wavelet functions are obtained by multiplying two wavelet functions or wavelet and scaling function for 1D. For the 2D case, there exist three wavelet functions that scan details in horizontal $\Psi^{(1)}(x,y)= \phi(x)\,\Psi(y)$, vertical $\Psi^{(2)}(x,y)= \Psi(x)\phi(y)$ and diagonal directions: $\Psi^{(3)}(x,y)= \Psi(x)\;\Psi(y)$. This may be represented as a four channel perfect reconstruction filter bank as shown in Fig. 1. Now, each filter is 2D with the subscript indicating the type of filter (HPF or LPF) for separable horizontal and vertical components. By using these filters in one stage, an image is decomposed into four bands. There exist three types of detail images for each resolution: horizontal (HL), vertical (LH), and diagonal (HH). The operations can be repeated on the low low (LL) band using the second stage of identical filter bank. Thus, a typical 2D DWT, used in image compression, generates the hierarchical structure shown in Fig. 2.

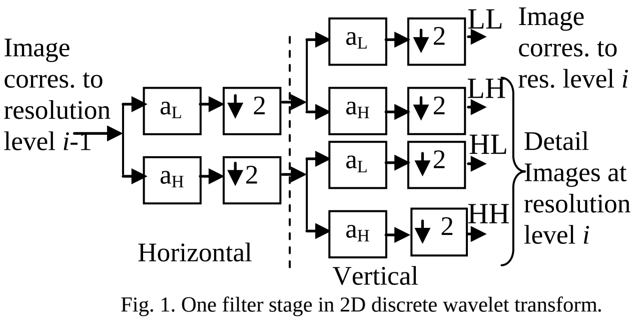

Fig. 1. One filter stage in 2D discrete wavelet transform.

<table>
<tr><td>LL</td><td>HL3</td><td rowspan="2">HL2</td><td rowspan="3">HL1</td></tr>
<tr><td>LH3</td><td>HH3</td></tr>
<tr><td colspan="2">LH2</td><td>HH2</td></tr>
<tr><td colspan="3">LH1</td><td>HH1</td></tr>
</table>

Fig. 2. Structure of wavelet decomposition.

The transformation of the 2D image is a 2D generalization of the 1D wavelet transformed already discussed. It applies the 1D wavelet transform to each row of pixel values. This operation provides us an average value along with detail coefficients for each row. Next, these transformed rows are treated as if they were themselves an image and apply the 1D transform to each column. The resulting values are all detail coefficients except a single overall average co-efficient. In order to complete the transformation, this process is repeated recursively only on the quadrant containing averages.

III. Related Works

Progressive image transmission (PIT) has been studied by many researchers and several schemes have been proposed. Two sub-bands refining method is based on wavelet. This PIT method consists of three steps: decomposition, transmission and reconstruction. In the decomposition step, the scheme performs Haar DWT on the image. Then the wavelet coefficients are represented by sub-bands [31].

In the transmission step, the sender transmits the sub-band along with some selected coefficients in the next sub-band to the receiver for image reconstruction. If the coefficients in the sub-band have been submitted, then the scheme transmits all the coefficients in the next sub-band. A concurrent scheme [32] for wavelet transform of image was proposed earlier. There the image plane is divided into *n* horizontal sections which are horizontally transformed concurrently. After then the image is divided into *n* vertical sections which are then vertically transformed concurrently. This system allows beginning of vertical transformation on some vertical sections before horizontal transformation in all sections is completed. Vertical sections that are already horizontally transformed can be vertically transformed.

----	▨	▨	▨	▨	▨	▨	1
----	▨	▨					2
----	▨	▨					3
----	▨	▨					…
----	▨	▨					…
----	▨	▨					…
----	▨						…
----							N

Fig. 3. Ordering the transformation.

Here it is possible for threads that completed horizontal transformation to go on to vertical transformation without having to wait on other threads to complete horizontal transformation. The gray color in Fig. 3 indicates sections of image data that are horizontally transformed. The white color indicates sections of image data that are not yet horizontally transformed. The gray vertical section with dotted spots can be assigned to a thread for vertical transformation.

IV. Proposed Method

This part demonstrates how the concurrent computing technique works over Discrete Wavelet Transformation (DWT). There are several algorithms for wavelet based compression such as Embedded Zerotree Wavelet (EZW), Set Partitioning in Hierarchical Trees (SPHIT), Wavelet Difference Reduction (WDR), Adaptively Scanned Wavelet Difference Reduction (ASWDR) etc. However, here we have considered the very basic technique called Haar Discrete Wavelet Transformation for color image transformation.

First of all, the image matrix is mapped from the digital image. For horizontal transformation, horizontal transformation threads are initialized. We have used separate process for every pair of rows of the matrix. We have made a little change with the algorithm of traditional Haar wavelet transform and thus the vertical transformation process is also embedded in the row transformation process to speed up the computation process and to avoid complexity. We have used separate threads for transforming red, green and blue (RGB) components. Thus multiple threads are started at a time. The threads normally transform the rows following the discrete Haar wavelet transformation. But as soon as the pair of elements of the same column is transformed the column transformation for those two components is also done. In the concurrent transformation scheme proposed in [32] vertical transformation starts after first element of last row has horizontally transformed. With our new proposal it speeds up the transformation process as no waiting time required for the column transformation after row transformation has been finished.

After completion of encoding process the matrix with detail coefficients and average coefficients is found. Then the required thresholding is applied on it and the final matrix is ready to transmit to the receiver. At the transmission phase we have transmitted the matrix by pair of columns. This is also done by several threads. The decoding process is done as the reverse of encoding. Thus the reconstruction of image requires comparatively lower time.

As illustrated in Fig. 4 (a), threads for horizontal transformation are started for each pair of rows at the same time. The region blue shaded region is horizontally transformed. In figure 4(c) horizontal transformation for the first two elements of the second row is completed. Then the vertical transformation is also done for available elements. The blue shaded region with dotted spots is vertically transformed (Figure 4 (d)).

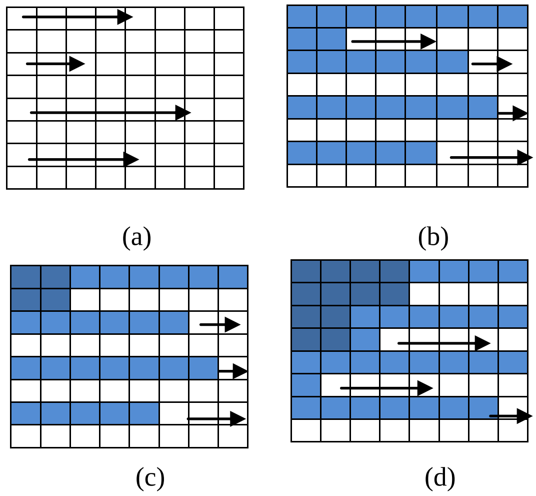

Fig. 4. Concurrent transformation a) Threads for each pair of row, b) Row transformation in progress, c) Column transformation done when pair of column elements are ready, d) Row and column transformation are in progress.

This method forms the output image with fractions as illustrated in Fig. 5. But as the fractions are parts of the original image without modification and it appears very fast the person can identify whether the image is the actual one or not within very short time.

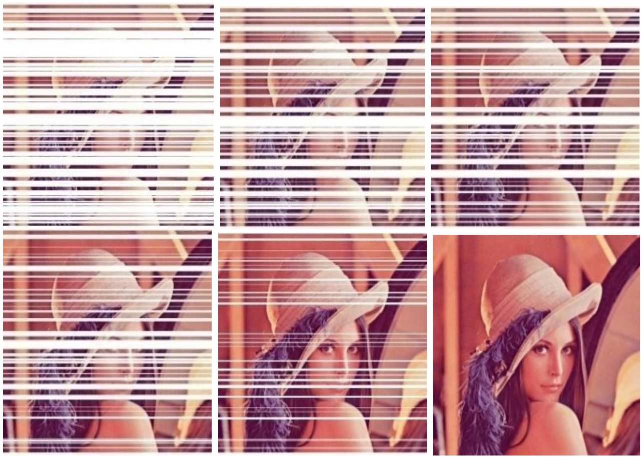

Fig. 5. Reconstructed image after receiving pairs of column concurrently, PSNR values (i) 25.2837dB (ii)26.5822 dB (iii) 27.2201 dB (iv) 28.1388 dB (v) 30.6938dB (vi) Reconstructed image 40.71 dB

V. Experimental Result

We tested some commonly used color images for image processing of various dimensions in the experiment. In Table II, the time required for one stage of encoding and decoding for images for normal discrete wavelet transformation and concurrent computing are shown. Column A's represent the time required for concurrent wavelet transformation and column B's represent the time required for normal discrete wavelet transformation.

The data presented in Table II shows how much the encoding and decoding time has been reduced by applying concurrent computing with comparison to normal sequential computing. Though the transmission time may vary because of transmission lines, the time required will be reduced much as the two main phases requires far less time. From our experimental result we have found that the required time for encoding and decoding is up to 90% or more less than the sequential computing. The performance may vary on different machines but still it gives much better time overhead.

TABLE II: Time Comparison Between Normal Discrete Wavelet Transformation and Concurrent Discrete Wavelet Transformation

	Encoding time (ms)		Decoding time (ms)	
Images	A	B	A	B
lenna(512x512)	0.8	52	0.3	78
baboon(480x480)	0.6	126.7	0.3	215
moon(1986x1986)	11	2173.1	5	2840
airplane(512x512)	2	143	0.7	216
pepper(512x512)	0.9	144	0.3	234

The required time to complete the whole process of image transmission for several images is shown in table II. In the two sub band refining method image data is divided into sub bands and then transmitted. Here all the sub bands have to transmit one by one. Thus the time required for completing the whole transmission process is as same as if the image data is transmitted in a sequential way. Thus the transmission time required without applying concurrency can be considered as the time required for sub band refining method. Table III shows the extent of better performance found after applying concurrent computing rather than sequential computing.

TABLE III: Comparison of Required Time of Transmission

Images	with concurrency(ms)	without concurrency(ms)
lenna	124	646
baboon	128	597
moon	1644	9524
airplane	120	677
pepper	120	650

A computer with Intel Core 2 CPU (2 GHz) is used as client and another one with Intel Pentium 4(2.66 GHz) CPU is used as server to collect data. On average applying concurrent computing it requires 80% to 82% less time than the sequential discrete wavelet transformation. Also the quality of reconstructed image holds acceptable visual quality.

A commonly used measure for quantifying the error between images is the Peak Signal to noise Ratio (PSNR). The PSNR for gray scale image (8 bits/pixel) is defined by-

$$PSNR(dB) = 20 \times \log_{10}\left(\frac{255}{\sqrt{MSE}}\right)$$

where MSE is the Mean Squared Error defined by-

$$MSE = \frac{1}{mn}\sum_{y=1}^{m}\sum_{x=1}^{n}\left(I(x,y) - I^{1}(x,y)\right)^{2}$$

where I is original image, I^1 is approximation of decompressed image and m, n are dimensions of the image.

Generally, when the PSNR is 40dB or larger, then the original image and the reconstructed image after transmission are virtually indistinguishable by human observers. Thus the PSNR of 37 dB or above is acceptable. In this case we've found that with our method, it provides good level of details to the output. Comparing with other wavelet-based PIT schemes [31] the output is acceptable. We have tested several images and found good quality of the reconstructed image compared with other PIT schemes. C.C. Chang has shown the PSNR [31] of 'lenna' found with his proposed approach is 33.03. With our approach this measure has found 40.71. In Table IV we have shown PSNR values for different images.

TABLE IV: PSNR of Reconstructed Image

Images	PSNR of concurrency method	PSNR of Two sub-bands refine method
lenna	40.71	33.03
barbara	36.34	37.16
baboon	38.97	38.46

VI. Conclusion

We have presented a method to enhance the progressive image transmission. The method uses concurrent computing on Discrete Wavelet Transformation to get the enhancement. This scheme is basically to reduce the image browsing time. Our method reconstructs the image with a few distortions and within far less time than the traditional one.

APPENDIX

Algorithm: Encoding

Here pixel matrix is defined by ‘mat’. The height of the image is ‘height’ and width of the image is ‘width’. Component’ defines the color component of the pixel.

```
segment initialize_thread(object reference){
     for offset = 1 to height
   interlock(reference) =true;
   thread_pool(offset,component) //calling thread
   offset:= offset+2;
     end loop
}
segment thread_pool(object  state, int component){
   float tmp[width],
   for i:= state to (state+2)
     int k=0, int mid=width/2,int midh=height/2;
            if i is even then
       for j:=1 to width
                    tmp[k]:=(mat[i,j,component]
                           +mat[i,j+1,component])/2;
         tmp[mid]: =mat[i,j,component]-tmp[k];
        j: =j+2;  k++; mid++;
       for t: =1 to width
          mat[i,t,component] =tmp[t];
              end if
           if  i is odd then
       for j:=1 to width
                    tmp[k]:=(mat[i,j,component]+
                    mat[i,j+1,component])/2;
          tmp[mid]:=mat[i,j,component]-tmp[k];
                               mat_1[(i-1)/2,k,component]=(
                               mat[i-1,k,component]          +
                               tmp[k])/2;
                    mat_1[midh,kk,component]=mat[i-1,k,c
                    omponent]-mat_1[(i-1)/2,
                    k,component)];
                               mat_1[(i-1)/2,mid,component]
                               =(mat[i-1,mid,component]+tm
                               p[mid] )/2;
                               mat_1[midh,mid,component]=
                               mat[i-1,mid,component]-mat_1
                               [(i-1)/2, mid,component];
        j: =j+2;  k++; mid++; midh++;
            end if
           end loop
interlock(reference) =false;
return to thread pool;
}
```

Code for the general Haar wavelet transformation

Algorithm: Transmit

```
Segment transmission(object reference){
int mid=width/2;
for i: = 1 to mid
   interlock(reference) =true;
              initialize a thread from from thread pool to
              transmit ith and (i+mid)th column
          interlock(reference) =false;
   i++;
end loop
}
```

General Haar Wavelet Transform

```
/*row transformation*/
for(i=0;i<row;i++){w=col;
   do{ k=0;
/*averaging*/     for(j=0;j<w/2;j++)
                        a[j]=((mat[i][j+j]+mat[i][j+j+1])/2);
/*differencing*/ for(j=w/2;j<w;j++,k++)
                          a[j]=mat[i][j-w/2+k]-a[k];
            for(j=0;j<row;j++) mat[i][j]=a[j];
            w=w/2;
   }while(w!=1);
}
/*column transformation*/
for(i=0;i<col;i++){ w=row;
   do{k=0;
/*averaging*/    for(j=0;j<w/2;j++)
                        a[j]=((mat[j+j][i]+mat[j+j+1][i])/2);
/*differencing*/for(j=w/2;j<w;j++,k++)
                        a[j]=mat[j-w/2+k][i]-a[k];
           for(j=0;j<w;j++) mat[j][i]=a[j];
           w=w/2;
   }while(w!=1);
}
```

ACKNOWLEDGMENT

The authors would like to thank Japan Society for the Promotion of Sciences (JSPS) for the financial support for performing this research. They also would like to thank the reviewers of this paper for the valuable comments and suggestions to improve the paper.

REFERENCES

[1] R. Williams. *Adaptive Data Compression*. Kluwer Academic Publishers, 1991

[2] C. Shannon. “A mathematical theory of communication.” *Bell Systems Technical Journal*, vol. 27, pp. 623-656, 1948.

[3] N. Faller. An adaptive system for data compression. In *Proceedings of the Asilomar Conference on Circuits, Systems and Computers*, pp. 593-597, 1973.

[4] R. Gallager. “Variations on a theme by Huffman.” *IEEE Transactions on Information Theory*, 24:668-674, 1978.

[5] D. Knuth. “Optimal binary search trees.” *Acta Informatica*, vol. 1, pp.14-25, 1971.

[6] J. Vitter. “Design and analysis of dynamic Huffman codes.” *Journal of the Association for Computing Machinary*, vol. 34, pp. 825-845, 1987.

[7] J. Ziv and A. Lempel. “A universal algorithm for sequential data compression.” *IEEE Transactions on Information Theory*, vol. 23, pp. 337-343, 1977.

[8] J. Ziv and A. Lempel. “Compression of individual sequences via variable-rate coding.” *IEEE Transactions on Information Theory*, vol. 24, pp. 530-536, 1978.

[9] T. Huang. “Run length coding and its extensions. In T. Huang and O. Tretiak, editors,” *Picture Bandwidth Compression*, pp. 231-264. Gordon and Breach, 1972.

[10] A. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[11] J. Lim. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, 1990.

[12] D. O'Shaughnessy. *Speech Communication: Human and Machine*. Addison-Wesley, 1987.

[13] M. Vetterli. “Multi-dimensional sub-band coding: Some theory and algorithms.” *Signal Processing*, 6:97-112, 1984.

[14] J. Woods and S. O'Niel. “Subband coding of images.” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 1278-1288, 1986.

[15] J. Lim. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, 1990.

[16] Y. Linde, A. Buzo, and R. Gray. "An algorithm for vector quantizer design." *IEEE Transactions on Communications*, vol. 28, pp. 84-95, 1980.

[17] J. Makhoul, S. Roucos, and H. Gish. "Vector quantization in speech coding." *Proceedings of the IEEE*, vol. 73, pp. 1551-1558, 1985.

[18] J. Abate. "Linear adaptive delta modulation." *Proceedings of the IEEE*, vol. 55, pp. 298-308, 1967.

[19] S. Alexander and S. Rajala. "Image compression results using LMS adaptive algorithm." *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, pp. 712-717, 1985.

[20] C. Cutler. "Differential quantization of communication signals." U.S. Patent 2 605 361, 1952.

[21] A. Habibi. "Survey of adaptive image coding techniques." *IEEE Transactions on Communications*, vol. 25, pp. 1275-1284, 1977.

[22] N. Jayant. "Adaptive delta modulation with a one-bit memory." *Bell Systems Technical Journal*, vol. 49, pp. 321-343, 1970.

[23] A. Kolmogorov. "Interpolation and extrapolation of stationary random series." *Journal of the Soviet Academy of Science*, pp. 3-14, 1941.

[24] T. Lei, N. Scheinberg, and D. Schilling. "Adaptive delta modulation system for video encoding." *IEEE Transactions on Communications*, vol. 25, pp. 1302-1314, 1977.

[25] J. O'Neal. "Predictive quantization system (differential pulse code modulation) for the transmission of television signals." *Bell Systems Technical Journal*, vol. 45, pp. 689-721, 1966.

[26] T. Lei, N. Scheinberg, and D. Schilling. "Adaptive delta modulation system for video encoding." *IEEE Transactions on Communications*, vol. 25, pp. 1302-1314, 1977.

[27] C. Song, J. Garondick, and D. Schilling. "A variable-step-size robust delta modulator." *IEEE Transactions on Communications*, vol. 19, pp. 1033-1044, 1971.

[28] N. Ahmed, T. Natarajan, and K. Rao. "Discrete cosine transform." *IEEE Transactions on Computers*, vol. 23, pp. 90-93, 1974.

[29] B. Fino. "Relations between Haar and Walsh/Hadamard transforms." *Proceedings of the IEEE*, vol. 60, pp. 647-648, 1972.

[30] H. Andrews. *Computer Techniques in Image Processing*. Academic Press, 1970.

[31] C. C. Chang and T. C. Lu, "A Wavelet-based Progressive Digital Image Transmission Scheme," *Proceedings of the First International Conference on Innovative Computing*, Information and Control – vol. 2, 2006, pp. 681-684.

[32] K. H. Talukder and K. Harada, "A Scheme of Model Verification of the Concurrent Discrete Wavelet Transform (DWT) for Image Compression," *International Journal of Signal Processing*, vol. 5 no. 1.

Kamrul Hasan Talukder received his PhD (Information Engineering) from Hiroshima University, Japan in 2008 before he finished his MSc. in Computer Science from National University of Singapore (NUS) in 2004 and BSc. in Computer Science and Engineering from Khulna University, Bangladesh in 1999. He joined Computer Science and Engineering Discipline of Khulna University, Bangladesh in 2000 where he is an Associate Professor now. He got NUS Research Scholarship for Masters study in Singapore, Japanese Government (Monbukagakusho:MEXT) scholarship for Doctor course study in Hiroshima University under the guidance of Prof. Koichi Harada. He has also received the Japan Society for the Promotion of Sciences (JSPS) postdoctoral fellowship in Hiroshima University. He is a member of Institute of Engineers, Bangladesh (IEB). He has published more than forty research papers in different international conferences and journals on different topics including image processing, formal verification, digital system, Bangla processing, software engineering etc.

Koichi Harada is a professor of the Graduate School of Engineering at Hiroshima University. He received the BE in 1973 from Hiroshima University, and MS and PhD in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research is mainly in the area of computer graphics. Special interests include man-machine interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is a member of ACM, IPS of Japan, and IEICE of Japan.