Progressive Image Transmission By Linear Quadtree Coding and Wavelet Transformation

by

Hon Wai Cheung
B.A.Sc., State University of New York at Binghamton, 1992

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Applied Science in The Faculty of Graduate Studies (Department of Electrical Engineering)

We accept this thesis as conforming to the required standard

The University of British Columbia
March 1997

© Hon Wai Cheung, 1997
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, Canada

Date 29th April, 1997
Abstract

Progressive image transmission allows an approximate image to be built quickly and the details to be transmitted progressively through several stages over the channel. This technique appears very useful for picture communication over slow channels. This thesis proposes to use linear quadtree encoding combined with wavelet based technique as well as other methods to achieve a hybrid coding progressive transmission system.
# Table of Contents

Abstract .................................................. ii

List of Figures ........................................ V

List of Tables ........................................... Viii

Acknowledgments ......................................... ix

Chapter One Introduction ................................ 1

Chapter Two Pseudo Progressive Transmission System ........................................... 3
  2.1 Progressive Transmission .................................. 3
  2.2 Display Algorithms ....................................... 6
    2.2.1 Interlacing Display Format ............................ 6
    2.2.2 Interlaced GIF Display Format ...................... 7

Chapter Three Quadrant interlacing Progressive Transmission System .................. 11
  3.1 Pyramid Data Structure .................................. 11
  3.2 Display Algorithm ....................................... 12
    3.2.1 Region Quadtree Decomposition .................... 12
    3.2.2 Linear Quadtree Encoding ......................... 14
    3.2.3 Linear Quadtree Decoding ......................... 15
    3.2.3 Quadrant Interlacing Display Format ............ 16
  3.3 Compression Method ..................................... 21
    3.3.1 Quadtree Condensation .............................. 21

Chapter Four Subband Coding Progressive Transmission System ......................... 22
  4.1 System Architecture .................................... 22
  4.2 Coding Method ......................................... 23
    4.2.1 Subband Coding Techniques ....................... 23
  4.3 Compression Methods .................................... 25
    4.3.1 Quantization ....................................... 25
      4.3.1.1 Scalar Quantization ........................... 25
      4.3.1.2 Vector Quantization ........................... 26
    4.3.2 Entropy Coding ................................... 26
  4.4 Progressive Transmission Scheme .................... 27

Chapter Five Hybrid Coding Progressive Transmission System .......................... 29
  5.1 System Architecture .................................... 29
  5.2 Transform Method ....................................... 30
    5.2.1 Wavelet Transforms ................................ 30
    5.2.2 Daubechies Wavelet Transform With ............. 30
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadrature Mirror Filter Bank</td>
<td>32</td>
</tr>
<tr>
<td>5.3 Progressive Transmission Schemes</td>
<td>38</td>
</tr>
<tr>
<td>5.3.1 Linear Quadtree Embedding</td>
<td>40</td>
</tr>
<tr>
<td>5.3.2 Zerotree Embedding</td>
<td>42</td>
</tr>
<tr>
<td>Chapter Six Results and Evaluation</td>
<td>46</td>
</tr>
<tr>
<td>Chapter Seven Conclusion</td>
<td>72</td>
</tr>
<tr>
<td>References</td>
<td>75</td>
</tr>
</tbody>
</table>
List of Figures

Figure 2.1 Richard (64x64) on raster display format ........................................ 4
Figure 2.2 Interlacing display format ................................................................. 6
Figure 2.3 Richard (64x64) on interlaced GIF display format ......................... 9
Figure 2.4 25% of Richard (64x64) on 3 different display formats ............... 9
Figure 2.5 Lenna (256x256) on interlaced GIF display format .................... 10
Figure 2.6 25% of Lenna (256x256) on 3 different display formats .......... 10
Figure 3.1 (a) Bitmap of a 8x8 binary image .................................................... 13
(b) Region decomposition of the image ......................................................... 13
(c) Quadtree representation of the same image ........................................... 13
Figure 3.2 (a) Linear quadtree representation of the binary image ............. 16
(b) Euclidean coordinates of the same image ............................................ 16
Figure 3.3 The flow chart of quadrant interlacing implementation ............ 18
Figure 3.4 Richard (64x64) on quadrant interlacing display format ........ 19
Figure 3.5 Lenna (256x256) on quadrant interlacing display format ........ 20
Figure 4.1 Subband coding progressive transmission system .................. 22
Figure 4.2 (a) System diagram of two-band spitting .................................. 24
(b) System diagram of two-band recombination .................................... 24
Figure 5.1 Hybrid coding progressive transmission system .................... 29
Figure 5.2 Transform matrix for DAUB4...................................................... 35
Figure 5.3 Pyramid algorithm ................................................................. 35
Figure 5.4 One stage in a multiscale image decomposition .................... 37
Figure 5.5 Image decomposition ......................................................... 37

Figure 5.6 One stage in a multiscale image reconstruction ....................... 38

Figure 6.1 (a) Richard (64x64) simple computer graphic ......................... 50

(b) Lenna (128x128) common processing image ................................. 50

(c) Brain (256x256) image of human brain obtained by MRI .............. 50

(d) Finger (512x512) FBI finger print sample ................................... 50

Figure 6.2 (a) The frequency distribution of the image "Richard" (64x64) and
the corresponding wavelet coefficients after several iterations .. 51

(b) The frequency distribution of the image "Lenna" (128x128) and
the corresponding wavelet coefficients after several iterations .. 52

(c) The frequency distribution of the image "Brain" (256x256) and
the corresponding wavelet coefficients after several iterations .. 53

(d) The frequency distribution of the image "Finger" (512x512) and
the corresponding wavelet coefficients after several iterations .. 54

Figure 6.3 (a) The corresponding wavelet coefficients map of figure 6.2(a) ..55

(b) The corresponding wavelet coefficients map of figure 6.2(b) ..56

(c) The corresponding wavelet coefficients map of figure 6.2(c) ..57

(d) The corresponding wavelet coefficients map of figure 6.2(d) ..58

Figure 6.4 (a) Hybrid coding progressive transmission display of "Richard"..59

(b) Hybrid coding progressive transmission display of "Lenna" ....60

(c) Hybrid coding progressive transmission display of "Brain" ....61

(d) Hybrid coding progressive transmission display of "Finger" ...62
Figure 6.5  
(a) Quadrant interlacing display of "Richard" ..........................63 
(b) Quadrant interlacing display of "Lenna" ..............................64 
(c) Quadrant interlacing display of "Brain" ..............................65 
(d) Quadrant interlacing display of "Finger" ..............................66 

Figure 6.6  
(a) The compressed images by Interlaced GIF .......................... 67 
(b) The compressed images by Quadrant interlacing .................... 68 
(c) The compressed images by Linear quadtree embedding ......... 69 
(d) The compressed images by Zerotree embedding .................... 70 
(e) The compressed images by SPIHT ..................................... 71
List of Tables

Table 6.1 Performance comparisons for the 512x512 Lenna image showing the peak-signal-to-noise rate (PSNR) in dB .................................................................48

Table 6.2 Subjective tests on 512x512 image "Lenna" ..................................................49
Acknowledgment

I would like to thank Dr. Gunther Schrack for his supervision and advice. His enthusiasm and policy made work with him a pleasure. I would like to thank Dr. Panos Nasiopoulos for his contributions at the beginning of this project. I also thank my colleagues Clifford Loo, Lawrence Tang, Philip Liu, and William Woo for their helpful support during the year. The calm friendship of Sandra Tam is much appreciated. Her support ensured that my time as a student was an enjoyable one. Most importantly I wish to thank my family, whose unquestioning support, both emotional and material, has been with me through all the endeavors of my life.
Chapter One

Introduction

It is often a problem how to store and transmit images efficiently. Since the high demand on Internet browsing, limited bandwidth and cost on mobile communication, it is necessary to compress images and transmit them progressively. Because of compression, less storage space is required on both transmitter and receiver sites. Then it will take less time to let the receiver display an image because of progressive transmission. There are numerous research on progressive image transmission has been conducted, such as using JPEG [Zhang, 1994]; Laplacian pyramid coding [Mongatti, 1992]; vector quantization [Riskin, 1994]; stack-run image coding [Tsai, 1996]; space-frequency quantization [Xiong, 1993]; embedded zerotree wavelet [Shapiro, 1993] and set partitioning in hierachical trees [Said, 1996]. In this research, we propose a quadrant interlacing system and a hybrid coding system to achieve such a goal.

Two different progressive image transmission techniques are provided by the proposed systems. The first is called the quadrant interlacing technique. It employs a pyramid data structure to decompose the image. It will produce a progressive display while the image is reconstructed. The second technique applies linear quadtree coding and wavelet transforms. By using linear quadtree embedding scheme or zerotree embedding scheme, the wavelet coefficients are
embedded into a bit stream sorted by numerical importance. A receiver has been designed so that it will display the image as soon as the first packet of coefficients arrives and will refine the image by adding on the incoming decoded coefficients. This technique will produce a good progressive image display. This thesis proposes the idea of using linear quadtree encoding combined with wavelet theory as well as other methods to achieve a hybrid coding progressive transmission system. In chapter 2, current interlaced display formats are discussed. In chapter 3, an interlaced format by using the pyramid data structure is proposed. In chapter 4, current subband progressive transmission system is described. In chapter 5, a hybrid coding progressive transmission system is proposed. In chapter 6, the performances of the proposed systems are evaluated. In chapter 7, the results of this thesis are summarized and discussed.
Chapter Two

Pseudo Progressive Transmission System

2.1 Progressive Transmission

Conventionally, an image is displayed in raster order – from left to right, top to bottom. Figure 2.1 is an example of a simple image displayed in raster order. However, if the transmission rate is very low, there are advantages to reconstruct the image differently. For example, 10 percent of the way through a raster transmission, only the top tenth of the picture can be seen, and usually this is not very informative. Generally it will be better to be able to see the whole picture at a tenth of the final resolution. This is why progressive image transmission is employed.

Progressive image transmission is a class of image transmission techniques where the information concerning the image is transmitted in two or more stages. At the first stage, the transmitter sends the information of a coarse image; at later stages, further details of the image can be transmitted on the request of the viewer at the receiver. A typical application of progressive image transmission is on-line-browsing [Rashwan, 1993][Malak, 1991], where the recipient wishes to see a recognizable image as quickly as possible. If it is the desired image, the recipient will request more information, perhaps at successive stages, to improve the quality of the image; if not, the recipient can terminate the transmission immediately.
Figure 2.1 Richard (64x64) on raster display format
In this thesis, we present two methods for progressive transmission. One is called "quadrant interlacing". It is a fractal coding technique to provide a progressive transmission effect. The other one is to apply a discrete wavelet transform followed by linear quadtree encoding to arrange the image spatial frequency information hierarchically.
2.2 Display Algorithms

2.2.1 Interlacing Display Format

Generally, image pixels are displayed in progressive order sequentially and progressively line by line. It is called raster (more specifically, non-interlaced raster) display format [Witten, 1994]. Figure 2.1 is an example of a raster display format. It is instrumentally the simplest form and is appropriate for scanning still pictures. However, for low transmission rates, other display formats have been considered to let the browser more easily recognize the image. Most of these involve scanning a fraction (a field) of the total number of pixels of a single scan and returning to scan the remaining pixels in subsequent scans. This type of display format is called interlaced raster scanning [Brown, 1967], in that each field, in a well chosen pattern, is spatially interlaced with the previous one. An interlacing format is illustrated in Figure 2.2

![Interlacing display format](image-url)
2.2.2 Interlaced GIF Display Format

The *Interlaced GIF* [Wilson, 1991] format implements interlacing. It is used to give the browser an idea of what an image looks like before all the bitmap data has been received. A *non-interlaced* image displays the image in a linear fashion from the top to the bottom. Over a slow link, it may take several minutes for an image to be painted. By the time when 50% of the bitmap data is received, only the top half of the image is displayed. The content of the bottom half is still a mystery to the browser.

In a non-interlaced GIF, the pixel lines are displayed in consecutive order. For example, the lines of a 16-line non-interlaced GIF file are stored as:

\[1, 2, 3, 4, 5 \ldots \ 16.\]

*Interlaced GIF* paints quickly over the entire display first as a very low resolution image. Only about 12.5% of the bitmap is displayed in the first pass. The GIF image is then repainted in three more passes, with each pass supplying more resolution until all of the data is received (12.5%, 25%, and 50% respectively). A browser can usually get a good idea of the entire image when only 30 ~ 50% of the bitmap data has been received. This is useful for knowing whether or not to abort an image being viewed via a slow link.
The lines of the same 16-line bitmap in an *interlaced GIF* are stored as:

1, 9, 5, 13, 3, 7, 11, 15, 2, 4, 6, 8, 10, 12, 14, 16.

Figure 2.3 and Figure 2.5 are examples of interlaced GIF display. Richard is a simple (64 x 64) computer image and Lenna is a complex (256 x 256) image from a photograph. A survey was conducted among ten people by asking their subjective advise. It shows that the images could both be recognized at the 50% stage.

In the next chapter, another interlaced format is introduced, called the "quadrant interlacing" display format. Figure 2.4 and Figure 2.6 show 25% of the images displayed by three different display formats. It is clear that "quadrant interlacing" provides more image information for the browser.
Figure 2.3  Richard (64x64) on Interlaced GIF display format

Figure 2.4  25% of Richard (64x64) on 3 different display formats
Figure 2.5 Lenna (256x256) on interlaced GIF display format.

Figure 2.6 25% of Lenna (256x256) on 3 different display formats
Chapter Three

Quadrant Interlacing Progressive Transmission System

3.1 Pyramid Data Structure

The progressive transmission technique of the quadrant interlacing progressive transmission system is based on the pyramid data structure [Alexandrov, 1993], where the images at the bottom of the pyramid correspond to a set of reduced-resolution approximations. Progressive image transmission is achieved by transmitting the levels of the pyramid ranging from the lowest resolution level (top level) to the original resolution level (bottom level). Several pyramid data structures have been described in the literature [Burt, 1983] [Tanimoto, 1979]. A pyramid data structure which is useful in representing digital images is the region quadtree [Samet, 1984]. Region quadtrees are attractive for a number of reasons:

. Relative simplicity compared to other methods (e.g., discrete-cosine-transform based coding), which makes it an attractive method for applications such as High definition television (HDTV) [Strobach, 1988].

. Allows focusing on specific local areas since the data representation is based on a decomposition [Chitale, 1991].

. The decomposition actually results in an image segmentation. This segmentation can be used for a variety of different image processing applications, e.g., progressive transmission [Caron, 1994].
3.2 Display Algorithm

3.2.1 Region Quadtree Decomposition

A natural gray-level image usually can be divided into regions of different sizes with varying amounts of detail and information. Such segmentation of the image is useful for an efficient coding of the image data. Region quadtree decomposition [Shusterman, 1994] is a technique which divides the image into two dimensional homogeneous regions.

For a given n x n image, its corresponding region quadtree is defined as follows: the root node represents the entire image. If all the pixels of the image are of the same color, the root node itself becomes a leaf node holding the color of the image. Otherwise, the root node has exactly four children (or descendents), the quadrants, which represent the colors of the northwest (NW), northeast (NE), southwest (SW) and southeast (SE) blocks of the image. This decomposition continues for all nodes until the block represented by the corresponding node consists of only one color. Figure 3.1 is an example of a region quadtree representation of an 8 x 8 binary image. Quadrants of each node are arranged in the order of NW, NE, SW and SE from the left. In Figure 3.1(a), 0 and 1 represent WHITE and BLACK pixels, respectively. It is assumed that the background color of the image is WHITE. Black blocks and their quadtree nodes are marked with A to P in Figure 3.1(b) and Figure 3.1(c).
Figure 3.1 (a) Bitmap of a 8x8 binary image; (b) Region decomposition of the Image; (c) Quadtree representation of the same image
3.2.2 Linear Quadtree Encoding

One way to store a quadtree is to use the tree structure. Since non-leaf nodes need space for the pointers to their quadrants, the space requirement for such a representation is high especially when the quadtree consists of a large number of nodes. Due to this fact, representations without using pointers are of interest. One of the pointerless quadtree representations is called the linear quadtree proposed by [Gargantini, 1982]. She pointed out that it is possible to attach a unique location code to a node, to interpret a location code as an interleaved coordinate and to replace a pointer-based quadtree by a list of the location codes of the black nodes. Each leaf node of a quadtree may be represented by a pair of integers \( \{n, l\} \), where \( n \) is a spatial address named the location code and \( l \) is the level. The root of the tree structure is declared as level 0, the first subdivision as level 1, etc. The level of a node is thus its distance in the tree from the root. A leaf node at level \( r \) is called a pixel. A location code is a quaternary integer; its digits comprise the path which is traversed in the quadtree to reach a leaf node from the root. Quadrants are labeled according to a labeling scheme called the Z-order or Morton path, hence, NW, NE, SW and SE quadrants are labeled 0, 1, 2, and 3 respectively. The most significant digit of a location code stores the quadrant of level 1, the following digit is the quadrant of level 2, etc., the least significant digit is the quadrant of the pixel-level. A location code has always exactly \( r \) quaternary digits. A node at level \( l < r \) is represented by the location code of its NW corner pixel, and ends in \( r-l \) 0 bits. A
linear quadtree is a list of node/level pairs of all black nodes. If a pixel has gray scale instead of being black, a value that indicates the intensity is assigned with the node/level pair as well. Figure 3.2(a) shows the linear quadtree of an (8 x 8) binary image.

3.2.3 Linear Quadtree Decoding

The location code of a pixel not only represents the traversal path in a quadtree. It is possible to calculate the Euclidean coordinates of the pixel from the location code [Schrack, 1992]. It is easy to show that the binary representation of the location code \( n \) of a pixel is an interleaved coordinate that has the following structure:

\[
n = y_{r-1}x_{r-1} \ldots y_{1}x_{1}y_{0}x_{0},
\]

where \( x = x_{r-1} \ldots x_{0} \), \( y = y_{r-1} \ldots y_{0} \), and \( x_{i} \), \( y_{i} \) are the binary digits of the internal representation of the coordinates. Figure 3.2 (b) shows the corresponding Euclidean coordinates of Figure 3.2(a).
3.2.4 Quadrant Interlacing Display Format

A quadrant interlacing display format is a pyramid data structure of the image. By using the idea of region quadtree decomposition and interlacing, we can provide a better interlaced format. Consider Figure 3.1(a) as an example.

Figure 3.1(c) is the region quadtree representation of the blocks. At the resolution 1, all four blocks are leaf nodes. Therefore the receiver prints black those four quadrants at [(000, 1), (100, 1), (200, 1), (300, 1)]. At resolution 2, only ten blocks are leaf nodes. So the quadrants [(010, 2), (030, 2), (100, 2), (120, 2), (200, 2), (210, 2), (230, 2), (300, 2), (310, 2), (320, 2)] are printed black. As the resolution increases, the detail of the quadrants is improved.
The implementation is shown in Figure 3.3. It involves linear quadtree coding. The upper left corner pixel of each quadrant is defined as the “P-Point”. Then the image is divided into four quadrants. If the quadrant has a black node, the “P-Point” and the level of that quadrant will be transmitted. Therefore, at the first time, it will have a maximum of four black quadrants. It is the image at resolution 1. At the next attempt, each quadrant is divided by 4. More “P-Points” are calculated for transmission. When the receiver places them back to the position and fills that specific square, the resolution is improved. If this procedure is operated recursively, the user will be able to recognize the image effectively. Figure 3.4 and Figure 3.5 are examples of quadrant interlacing displays of Richard 64x64 and Lenna 256x256 respectively. A subjective observation experiment has been conducted. It shows that Richard can be recognized at 25% (resolution is 5) of data transmitted while Lenna can be recognized already at only 6.25% (resolution is 6) of data transmitted.
Input image bitmap

Linear quadtree encoding

Level = 1

Calculate the "P-Point" and the size of each quadrant

Quadrant = 1

Check: any pixel lies in that quadrant?

Yes

Write the (P-Point, level, gray scale) to the file and skip the rest of the pixels which lay in that quadrant

No

Increment quadrant by 1

No

End of file?

Yes

Increment level by 1

No

Level = resolution?

Yes

End

Figure 3.3 The flow chart of quadrant interlacing implementation
Resolution = 1
4 coefficients transmitted

Resolution = 4
256 coefficients transmitted

Resolution = 2
16 coefficients transmitted

Resolution = 5
1024 coefficients transmitted

Resolution = 3
64 coefficients transmitted

Resolution = 6
4096 coefficients transmitted

Figure 3.4 Richard (64X64) on quadrant interlacing display format
Figure 3.5 Lenna (256x256) on quadrant interlacing display format
3.3 Compression Method

3.3.1 Quadtree Condensation

As demonstrated by Figure 3.2, four identical pixels are grouped into a single node and assigned a lower level value to the linear quadtree code, reducing the size of the image file. For example, the segments B, D, J, P can be stored as four linear quadtree codes at level two instead of sixteen linear quadtree codes at level three. Obviously, an image which has more spatial redundancy will perform better. Also, it works well on an image which has many zero nodes because the linear quadtree coding excludes the zero nodes. The degree of compression is image-dependent. Some images, such as Richard, yield a better compression ratio than others.

The above technique is a lossless compression. If a user can tolerate some errors, a lossy compression approach can be considered. Take Lenna as an example. There is not a large visual difference between the images of resolution 7 and of resolution 8. Therefore, the image can be stored at resolution 7 and yield a 75% compression.
Chapter Four

Subband Coding Progressive Transmission System

4.1 System Architecture

The design of a subband coding progressive transmission system can be divided into three parts: subband coding techniques, quantization strategies, and error-free encoding techniques. In each part, one has the freedom to choose from a pool of candidates and this choice will ultimately affect the system performance.

In this chapter, a general system design is introduced. Most of the systems are using the following architecture.

![Subband coding progressive transmission system diagram](image)

Figure 4.1 Subband coding progressive transmission system
4.2 Coding Method

4.2.1 Subband Coding Techniques

The basic idea of subband coding [Woods, 1986] is to decompose the image into subimages, according to the frequency zones. A bit rate is assigned to each subimage individually according to the energy content of that subimage. The low frequency bands contain maximum information, while high frequency bands contain less information. Hence more bits are allocated to low frequency bands. This results in a high quality coding at low bit rates.

In a two-band subband decomposition, the encoder consists of two filters, a lowpass filter $h_0(z)$ and a highpass filter $h_1(z)$, followed by downsampling by two.

It gives rise to two subimages which are full band at a lower sampling rate $\frac{1}{2}f_s$.

The whole process is known as *decimation*. Similarly, on the synthesis side, there is an upsampling by two followed by an interpolation by filters $\tilde{h}_0(z)$ and $\tilde{h}_1(z)$. Figure 4.2 shows a two-band splitting system.
Figure 4.2 (a) System diagram of two-band splitting
(b) System diagram of two-band recombination
4.3 Compression Methods

4.3.1 Quantization

Quantization is the process to convert continuous quantities into digital quantities. The data may be in scalar form or vector form. The conventional techniques such as pulse code modulation perform the quantization on the scalars, e.g. individual real valued samples of the waveform or pixels of an image.

4.3.1.1 Scalar Quantization

A scalar quantizer discretizes one sample at a time. It is a special case of vector quantization. A scalar quantizer $Q$ with $L$ levels is entirely defined by two sets of values: $L+1$ values $x_0, x_1, \ldots, x_L$ called decision levels, which divide the space of real numbers, and $L$ reproduction values (output values) $y_1, y_2, \ldots, y_L$ such that:

$$Q(x) = y_i \quad \text{if} \quad x_{i-1} \leq x < x_i \quad \text{for} \quad i = 1, 2, \ldots, L$$

Suppose the resolution of an image is $N$. This will yield $3N+1$ subbands. Since the variance of each subband is generally different, a quantizer is designed for each subband. Two types of errors occur during quantization.

1. Quantization noise: this is the difference between the input value and the reproduction value in the domain $[x_0, x_L]$.
2. Overload noise: this is the truncation effect when the signal exceeds the boundary decision thresholds $x_0$ and $x_L$ and the values are quantized as $y_f$ and $y_L$.

### 4.3.1.2 Vector Quantization

Vector quantization is a generalization of scalar quantization in which vectors, or blocks, of data are quantized instead of the data themselves. To apply vector quantization to subband coding, the common approach is still to consider each subband individually. A sub-code-book is generated for each subband, and a multiresolution codebook is obtained by assembling all sub-code-books. Senoo and Girod [Senoo, 1992] compared several vector quantization algorithms for subband image coding.

### 4.3.2 Entropy Coding

After the quantization, the subbands usually are highly correlated. Entropy encoders, such as the Huffman coder or the run-length coder, are used to further compress the data. Commonly, run-length encoding an abundance of zeros, when combined with Huffman encoding of the nonzero values, produces good results.
4.4 Progressive Transmission Scheme

A progressive image transmission technique for subband coding was proposed by Westerink [Westerink, 1989]. The basic idea of subband coding is to split up the frequency band of a signal and then to encode each subband separately. For the progressive transmission of a subband encoded image, the subbands are sent one after the other. The transmission order of subbands is determined by the total mean squared error, $D_{\text{tot}}$, which can be expressed as the sum of the mean square error values $d_i$ per subband:

$$D_{\text{tot}} = \sum_{i=1}^{N} d_i$$

where $N$ is the number of subbands. To determine the transmission order, the transmitter evaluates the maximum of decrease in total mean square error which is equal to:

$$\max_{i \in S} \frac{\sigma_i^2 - d_i}{b_i}$$

where $b_i =$ number of bits for subband $i$; $\sigma_i^2 =$ variance of subband $i$;

$S =$ set of subbands that has not yet been transmitted.

The actual transmission order will be made known to the receiver by encoding the subband numbers and sending these along with the corresponding subband data. A specific scanning pattern is used to replace the coefficients within the subband. The zigzag scanning pattern and raster scanning pattern are commonly used.
There are several disadvantages of this scheme. Firstly, traditional transform coders, such as those using the discrete cosine transform, decompose images into a representation in which the edge information tends to disperse so that many non-zero coefficients are required to represent edges with good fidelity. Secondly, no code embedding occurs within the subband because it has to follow a specific scanning pattern.

In the next chapter, we propose a wavelet transform system with linear quadtree code embedding to solve the above problems.
Chapter Five

Hybrid Coding Progressive Transmission System

5.1 System Architecture

The Hybrid coding scheme is characterized by the following concepts:

1. Daubechies wavelet transform
2. Linear quadtree encoding

The Daubechies wavelet transform decorrelates most images fairly well. The significant coefficients can be encoded as a quadtree. After the transformation, linear quadtree encoding assigns location codes to the coefficients for embedding. Finally, the coefficients will effectively transmitted by a embedding scheme. Figure 5.1 shows the architecture.

Figure 5.1 Hybrid coding progressive transmission system
5.2 Coding Method

5.2.1 Wavelet Transforms

Wavelets are functions generated from one single function $\Psi$ by dilation and translations,

$$\Psi^{a,b}(t) = |a|^{-\frac{1}{2}} \Psi\left(\frac{t-b}{a}\right)$$

where $t$ is a one-dimensional variable.

The mother wavelet $\Psi$ has to satisfy the condition:

$$\int_{-\infty}^{\infty} \Psi(x)dx = 0$$

The definition of wavelets as dilates of one function means that high frequency wavelets correspond to $a<1$ or narrow width, while low frequency wavelets have $a>1$ or wide width.

The basic idea of the wavelet transform is to represent an arbitrary function $f$ as a superposition of wavelets. Any such superposition decomposes $f$ into different scale levels, where each level is then further decomposed with a resolution adapted to the level. One way to achieve such a decomposition writes $f$ as an integral over $a$ and $b$ of $\Psi^{a,b}$ with appropriate weighting coefficients. In practice, one prefers to write $f$ as a discrete superposition (as a sum rather than an integral). It introduces a discretization, $a = a_0^m, b = nb_0a_0^m$, with $m, n \in \mathbb{Z}$, and $a_0 > 1, b_0 > 0$ fixed. The wavelet decomposition is then
\[ f = \sum c_{m,n}(f) \Psi_{m,n} \]

with

\[ \Psi_{m,n}(t) = \Psi^{a_0^{-m}, b_0^{-n}}(t) = a_0^{-m} \Psi(a_0^{-m}t - nb_0) \]

Decomposition of this type were studied in [Daubechies, 1986]. For \( a_0 > 2, b_0 > 1 \) there exist very special choices of \( y \) such that the \( \Psi_{m,n} \) constitute an orthonormal basis, so that

\[ C_{m,n}(f) = \langle \Psi_{m,n}, f \rangle = \int \Psi_{m,n}(x)f(x)dx \]

Different bases of this nature were constructed by Daubechies [Daubechies, 1988] and others. All these examples correspond to a multiresolution analysis, a mathematical tool invented by S. Mallat [Mallat, 1989] which is particularly well adapted to the use of wavelet bases in image analysis, and which gives rise to a fast computation algorithm.
5.2.2 Daubechies Wavelet Transforms with Quadrature

Mirror Filter Bank

The discrete wavelet transform, as developed by Daubechies, has many similarities to the Fast Fourier transform. Both take an input vector whose length is normally a power of two and output a different vector of the same length. The entire process is also reversible which means that transform data can be used to reconstruct the original input at any point in the procedure. The Fast Fourier transform yields a unique decomposition in terms of continuous sines and cosines as the transform output. The wavelet transform yields a decomposition which is neither continuous nor unique. The primary advantage of the discrete wavelet transform is that it does not have the limited time-frequency resolution of the Fast Fourier transform and thus provides a more accurate representation of the input [Rioul, 1991].

There are many different types of discrete wavelet transforms which have been explored since the original work in the 1980's and each have their own advantages and disadvantages when performing data analysis. As an example, some provide smoothing the representation of a signal, while others represent the signal more compactly for data compression. A particular class of wavelets may have several different high order solutions which yield a variety of characteristics from highly compacted to highly smooth within the same family [Lu, 1996].
A specific discrete wavelet transform is identified by a particular set of numbers called wavelet filter coefficients. One of the simplest and most localized wavelet transforms is referred to as DAUB4 which is based on four wavelet coefficients first solved by Daubechies [Daubechies, 1986]. This particular wavelet transform is easy to implement, and is computationally efficient.

Daubechies discovered that the wavelet transform could be implemented with a specifically designed pair of finite impulse response filters called a quadrature mirror filter pair. A finite impulse response filter performs a dot product between wavelet filter coefficients and the input data vector. The output of this filter is a decimated (every other sample removed) version of the input data. Repeating this process recursively results in a group of signals that divide the spectrum of the original signal into octave bands with successively coarser measurements in time as frequency decreases [Riou, 1992].

The same effect can be accomplished mathematically by repeatedly applying a transform matrix comprising wavelet coefficients to a column vector of discrete input data. After each application of the transform matrix, the detail data and the smoothed or decimated data must be separated. The transform matrix is then applied to the smoothed data as the process is repeated.

Figure 5.2 and 5.3 illustrates the wavelet transform process. The transform matrix (in this case DAUB4) is applied to the input \((y_1 \ldots y_{16})\) yielding smooth
data (s1 . . . s8) interleaved with detail data (d1 . . . d8). The results are permutated to separate the smooth and detail data (Figure 5.3). The detail data is simply stored while the transform matrix is applied to the smoothed data. Each repetition of the process divides the smooth data in half. The process can be terminated at any point, but usually proceeds until there are only two data points left. The final output is exactly the same number of data points as the input and can be used to reconstruct the original data.

Note the pattern of coefficients used for the diagonal of the transform matrix in Figure 5.2. The rows consist of identical pairs of coefficients. The exception to this are the last two rows which simply wrap the values back into the first column. The odd rows are the smoothing filters while the even rows yield the detail data. This accounts for the interleaved data output in Figure 5.2. The values wrap in the last row because it is assumed that data represents a repeating waveform. In a signal processing context, this pattern is called a quadrature mirror filter [Vaidyanathan, 1990].
\[
\begin{bmatrix}
  c_0 & c_1 & c_2 & c_3 \\
  -c_3 & c_2 & -c_1 & c_0 \\
  c_0 & c_1 & c_2 & c_3 \\
  -c_3 & c_2 & -c_1 & c_0 \\
  c_0 & c_1 & c_2 & c_3 \\
  -c_3 & c_2 & -c_1 & c_0 \\
  c_0 & c_1 & c_2 & c_3 \\
  -c_3 & c_2 & -c_1 & c_0 \\
  c_2 & c_3 \\
  -c_1 & c_0
\end{bmatrix}
\]

\[
\begin{bmatrix}
  y_1 \\
  y_2 \\
  y_3 \\
  y_4 \\
  y_5 \\
  y_6 \\
  y_7 \\
  y_8 \\
  y_9 \\
  y_{10} \\
  y_{11} \\
  y_{12} \\
  y_{13} \\
  y_{14} \\
  y_{15} \\
  y_{16}
\end{bmatrix}
\]

Figure 5.2 Transform matrix for DAUB4

Figure 5.3 Pyramid algorithm
The DAUB4 transform matrix is orthogonal, thus the inverse is just a transposed version of the original. The inverse matrix can be used to reverse the wavelet transform and reconstruct the original data. DAUB4 provides a very compact representation of the input and most of the information for reconstruction is contained in the highest amplitude points of the wavelet data.

For images, a two-dimensional wavelet transform is needed. It is accomplished by two separate one-dimensional transforms. Figure 5.4 represents one stage in a multiscale pyramidal decomposition of an image: wavelet coefficients of the image are computed, as in the one-dimensional case, using a subband coding algorithm. The h and g are quadrature mirror filters. The coefficients of DAUB4 are given by [Daubechies, 1986]:

\[ h_0 = \frac{1 + \sqrt{3}}{2}, \quad h_1 = \frac{3 + \sqrt{3}}{2}, \quad h_2 = \frac{3 - \sqrt{3}}{2}, \quad h_3 = \frac{1 - \sqrt{3}}{2} \]

and

\[ g_i = (-1)^i h_{i-1}, \quad \text{where} \quad i = 0, 1, 2, 3 \]

This decomposition provides subimages corresponding to different resolution levels and orientations (see Figure 5.5). The reconstruction scheme of the image is presented Figure 5.6.
Figure 5.4 One stage in a multiscale image decomposition

<table>
<thead>
<tr>
<th>Low resolution sub-image</th>
<th>horizontal orientation sub-image (LH1)</th>
<th>horizontal orientation sub-image (LH2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical orientation sub-image (HL1)</td>
<td>diagonal orientation sub-image (HH1)</td>
<td>vertical orientation sub-image (HL2)</td>
</tr>
</tbody>
</table>

Figure 5.5 Image decomposition
5.3 Progressive Transmission Schemes

The wavelet transform concentrates the original values of a two-dimensional data set or image into a relatively small number of coefficients. The information is carried by the large magnitude coefficients. The majority of the coefficients are relatively small in value and do not contribute much information to the reconstruction of the original data. However, the position of all the coefficients must still be retained for reconstruction of the wavelet coefficient map.
The major objective in a progressive transmission scheme is to select the most important information to be transmitted first. The mean square error (\(D_{\text{mse}}\)),

\[
D_{\text{mse}} = \frac{1}{N} \sum_i \sum_j \left( F(i,j) - f(i,j) \right)^2 = \frac{1}{N} \sum_i \sum_j \left( W(i,j) - w(i,j) \right)^2
\]

where \( N = \text{the number of image pixels} \)

\( F = \text{the original image} ; f = \text{reconstructed image} \)

\( W = \text{transform coefficient of the original image} \)

\( w = \text{transform coefficient of the reconstructed image} \)

and \((i, j) = \text{the pixel coordinates}\)

which measures the largest distortion reduction, is used to determine the transmission order of the coefficients. Other proposals for progressive image transmission suggest the same strategy, for example, [DeVore, 1992] stated that after a unitary hierarchical subband transformation, the coefficients with larger magnitude should be transmitted first because they have a larger content of information.

Two progressive transmission schemes are presented below. Each has its advantages and disadvantages. The choice of the scheme depends on the application.
5.3.1 Linear Quadtree Embedding

An image may be represented as a multiscale process by utilizing an orthogonal
wavelet transform of the original data. In the dyadic orthogonal case, the
discrete wavelet transform provides information about the edges in an image in
the form of subbands along three different orientations: horizontal (LH), vertical
(HL) and diagonal (HH) (see Figure 5.5). If the nodes on a tree are viewed as
the wavelet coefficients, then a causal relationship may be defined between
edges of the image at a coarse scale and those in the same spatial position at a
finer scale along one orientation.

For example: consider an 8 x 8 image, which can be transformed into three
different resolutions. For each orientation, it will have three wavelet coefficient
subbands. For each coefficient in the coarser band at each orientation, there are
four coefficients at the next resolution and sixteen at the finest resolution. The
21 coefficients represent the edge information along one orientation for an 8 x 8
image. The relationships of these coefficients may be represented by a
quadtree which means that linear quadtree coding can be applied in order to
locate the position of these wavelet coefficients.

First, the image is processed by a two-dimensional wavelet transform. Then,
each coefficient is assigned a location code by the linear quadtree coding
method which described in section 3.2. Each coefficient now becomes a
coded-coefficient-pair which includes the location code and the value of the coefficient. After sorting the wavelet coefficients by magnitude, the information is ordered in importance. The transmitter first sends a resolution value (R) which is used to calculate the size of the image:

\[
\text{the size of the image} = 4^R
\]

followed by a series of embedded coded-coefficient-pairs.

At the receiver, the decoder simply decodes the location code and then places the coefficient into the quadtree for each coded-coefficient-pair. The image can be reconstructed by an inverse wavelet transform of the received coefficients.

There are several advantages of this scheme. First, the encoder and decoder design can be very simple and can easily be hardware implemented. Second, the linear quadtree coding and discrete wavelet transform are both very fast operations. Third, the system can be very robust. Because the coded-coefficient-pairs are independent of each other. For example, the channel noise caused by a the burst error only affects a small area on the reconstructed image. Also, the decoding of each coefficient location does not depend on previous information so that the system is stable even over a noisy channel.

The disadvantage of this scheme is the poor compression performance. It is due to the attachment of a location code to each coefficient. The characteristic of the
following scheme is almost opposite. It results in a good compression ratio by trading off a complicated coding method.

### 5.3.2 Zerotree Embedding

This progressive image transmission scheme is similar to the embedded image coding method proposed by [Shapiro, 1993]. Linear quadtree coding and the neighbor finding method are applied to speed up the search of zero roots.

After the discrete wavelet transformation, every coefficient at a given scale can be related to the set of coefficients of the next finer scale of similar orientation (see Figure 5.5). The coefficient at the coarse scale is called the parent, and all coefficients corresponding to the same spatial location at the next finer scale of similar orientation are called the children. For a given parent, the sets of all coefficients at all finer scales of similar orientation corresponding to the same location are called the descendants. For a given child, the sets of all coefficients at all coarser scales of similar orientation corresponding to the same location are called the ancestors.

After the linear quadtree coding, the parent-child relationship can be determined by their location code. For example, let \( P \) be the location code of a parent. The children are the next four nodes starting at location \( P \times 4^1 \) and the next sixteen nodes beginning at location \( P \times 4^2 \) ... etc. For a given location, the location all of
the descendants can be calculated by the above method. Also, the scanning of the coefficient map is performed by incrementing the location codes so that no child node is scanned before its parent.

Given a threshold level $T$, a coefficient is called significant if it is larger than $T$. The entire wavelet coefficient map can be coded as a string of symbols from a 4-symbol alphabet which can be represented by 2 bits each. This is called Zerotree coding. The four symbols used are:

(a) Positive significant (P) if the coefficient is positive and significant.

(b) Negative significant (N) if the coefficient is negative and significant.

(c) Isolated zero (I) if the coefficient is insignificant but has some descendants that are significant.

(d) Zero root (Z) if the coefficient and all of its descendants are insignificant.

To perform the coding, a sequence of thresholds ($T_j$):

$$T_j = \frac{T_{j-1}}{2}$$

are chosen. Two separate files are maintained during encoding and decoding. Initially, the dominant file contains all coded-coefficient-pair, and the subordinate file is empty. During processing, the dominant file contains the coded-coefficient-pair of those coefficients that have been found to be insignificant. The subordinate file contains the magnitude of those coefficients that have been found to be significant. For each threshold, each file is scanned once.
During a dominant pass, coefficients in the dominant file are compared to the threshold $T_j$ to determine their significance. The result is coded by the four zerotree coding symbols and appended to the output file for transmission. Each time a coded-coefficient-pair is encoded as significant, its magnitude is appended to the subordinate file, and it is erased from the dominant file.

A dominant pass is followed by a subordinate pass. During a subordinate pass, the width of the effective quantizer step size, which defines an uncertainty interval for the true magnitude of the coefficient, is cut in half. For each magnitude on the subordinate file, this refinement can be encoded using a binary alphabet with a "1" symbol indicating that the true value falls in the upper half of the old uncertainty interval and a "0" symbol indicating the lower half. The symbols are then appended to the output file for transmission. The process continues to alternate between dominant passes and subordinate passes where the threshold is halved before each dominant pass.

In the decoding operation, each decoded symbol, both during a dominant and a subordinate pass, refines and reduces the width of the uncertainty interval in which the true value of the coefficient may occur. The center of the uncertainty intervals are used as the reconstruction values. When the quadtree of the wavelet coefficients is reconstructed, the coefficients are inverse wavelet transformed in descending order according to their magnitude.
The advantage of this scheme is the good compression performance. It takes about one byte to represent the location and the value of the coefficient. Also, it gives a good reconstructed image quality even at high compression ratios.

There are several disadvantages of this scheme. First, the encoder and decoder design is relatively complex. Second, the coding scheme is slow due to swapped and scanned information in several files. Third, the system is not robust. Because the reconstruction of the wavelet coefficient quadtree depends on the previous information so that the system may not be stable over a noisy channel.
Chapter Six

Results and Evaluation

Results from computer simulations using the proposed progressive image transmission schemes on four test images are presented. The first image, shown in figure 6.1(a), is a 64x64 computer graphic "Richard". It features a simple image structure with only a few gray levels. The second, shown in figure 6.1(b), is a 128x128 image "Lenna". It is characterized by fine detail. The third, shown in figure 6.1(c), is a 256x256 image "brain" obtained by magnetic resonance imaging. It represents a possible application in medical image transmission. The fourth, shown in figure 6.1(d), is a 512x512 finger print sample "finger" provided by the FBI. FBI is using one of the wavelet transform algorithms for compression of their finger print files. This image is characterized by many sharp edges.

A Matlab toolbox called WaveLab [Buck, 1996] was obtained from Stanford University to perform the two-dimensional Daubechies wavelet transformation. A set of programs were written to implement the different algorithms which are described in the previous chapter. The results are listed below.

The first step feeds an image into the Daubechies 4 wavelet transformer. The wavelet transformer concentrates the original values of an image into a relatively small number of coefficients. Figure 6.2(a)-(d) shows the frequency distribution
of the original images and the corresponding wavelet coefficients after several iterations. The majority of the coefficients are relatively small in value and do not contribute much information to the reconstruction of the original data. Figure 6.3(a)-(d) are the wavelet coefficients maps of those images. The figures show the images decomposed into subbands by wavelet transformation and could be embedded by linear quadtree coding. The hybrid coding with the linear quadtree embedding scheme and the quadrant interlacing progressive transmission display of each image on each level are shown in figures 6.4(a)-(d) and figures 6.5(a)-(d).

For an evaluation of system performance, objective and subjective fidelity criteria are applied on the 512x512 image "Lenna". On objective fidelity measure, the root mean square error (MSE) and the peak-signal-to-noise ratio (PSNR) are calculated. The MSE provides a pixel by pixel measure of the image change from one pass to another. The average square error between the two images, where the size of the image is given as M x N, is given by the expression:

\[ MSE = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} [g(m,n) - f(m,n)]^2 \]

where M and N are the number of rows and columns of the images and \( f(m,n) \) and \( g(m,n) \) are the \( (m,n) \) pixels of the original and the reconstructed image, respectively. The peak signal to noise ratio (PSNR) is given by the expression:
\[ PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \text{ dB} \]

The coding results for the image "Lenna" at different rates are listed in table 6.1. For comparison purposes, we also include in table 6.1 the results for Interlaced GIF, EZW [Shapiro, 1993] and SPIHT [Said, 1996].

<table>
<thead>
<tr>
<th>Rate (bpp)</th>
<th>Interlaced GIF</th>
<th>Quadrant Interlacing</th>
<th>Linear Quadtree Embedding</th>
<th>Zerotree Embedding</th>
<th>EZW</th>
<th>SPIHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>11.64</td>
<td>24.18</td>
<td>26.55</td>
<td>29.32</td>
<td>30.23</td>
<td>31.11</td>
</tr>
<tr>
<td>0.25</td>
<td>11.71</td>
<td>25.23</td>
<td>27.47</td>
<td>32.62</td>
<td>33.17</td>
<td>34.14</td>
</tr>
<tr>
<td>0.5</td>
<td>11.86</td>
<td>26.37</td>
<td>30.11</td>
<td>35.65</td>
<td>36.28</td>
<td>37.25</td>
</tr>
<tr>
<td>1.0</td>
<td>12.19</td>
<td>28.67</td>
<td>33.21</td>
<td>38.77</td>
<td>39.55</td>
<td>40.46</td>
</tr>
</tbody>
</table>

Table 6.1: Performance comparisons for the 512x512 Lenna image showing the peak-signal-to-noise ratio (PSNR) in dB

Subjective tests are used to find the answer how good an image looks to a human observer. The subject quality of an image can be evaluated by showing the image to a number of observers and averaging their evaluations. A six-grade quality scale is applied:

1. Excellent - The image is of extremely high quality.
2. Fine - The image is of high quality, providing enjoyable viewing.
3. Passable - The image is of acceptable quality.
4. Marginal - The image is of poor quality and it should be improved.
5. Inferior - The image quality is very poor.
6. Unusable - The image is so bad that it cannot be recognized.
Figure 6.6(a)-(e) shows the compressed images by the schemes that are listed in table 6.2. Subjective testing results for the progressive transmission algorithms are shown in Table 6.2. The results have been derived from test sessions, where two observer groups of 3 and 5 people were used. One was an expert group and the other group consisted of non-experts. All groups had quite consistent opinions, although some people said that it was very hard to find any differences between some algorithms.

<table>
<thead>
<tr>
<th>Rate(bpp)</th>
<th>Interlaced GIF</th>
<th>Quadrant Interlacing</th>
<th>Linear Quadtree Embedding</th>
<th>Zerotree Embedding</th>
<th>SPIHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>6.0</td>
<td>4.8</td>
<td>3.2</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.25</td>
<td>6.0</td>
<td>3.2</td>
<td>2.3</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.5</td>
<td>6.0</td>
<td>2.4</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>6.0</td>
<td>1.5</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 6.2: Subjective tests on 512x512 image “Lenna”

The results show that the wavelet-based embedding coders consistently outperform the interlacing methods. However, the quadrant interlacing method still produced acceptable results and it is a very fast and simple algorithm. The wavelet-based schemes all produced very high quality images, even the linear quadtree embedding scheme with a high compression ratio.
Figure 6.1 (a) Richard (64x64) simple computer graphic  
(b) Lenna (128x128) common processing image  
(c) Brain (256x256) image of human brain obtained by MRI  
(d) Finger (512x512) FBI finger print sample
Figure 6.2(a) The frequency distribution of the image "Richard" (64x64) and the corresponding wavelet coefficients after several iterations. On each graph: Y axis: Frequency of occurrence. X axis: Coefficients value.
Figure 6.2(b) The frequency distribution of the original image "Lenna" (128x128) and the corresponding wavelet coefficients after several iterations.
On each graph: Y axis: Frequency of occurrence
X axis: Coefficients value
Figure 6.2(c) The frequency distribution of the original image "Brain" (256x256) and the corresponding wavelet coefficients after several iterations. On each graph: Y axis: Frequency of occurrence, X axis: Coefficients value.
Figure 6.2(d) The frequency distribution of the original image "Finger" (512x512) and the corresponding wavelet coefficients after several iterations. On each graph: Y axis: Frequency of occurrence, X axis: Coefficients value.
Figure 6.3(a) The corresponding wavelet coefficients map of figure 6.2(a)
Figure 6.3(b) The corresponding wavelet coefficients map of figure 6.2(b)
Figure 6.3(c) The corresponding wavelet coefficients map of figure 6.2(c)
Figure 6.3(d) The corresponding wavelet coefficients map of figure 6.2(d)
Figure 6.4(a) Hybrid coding progressive transmission display of "Richard"
Figure 6.4(b) Hybrid coding progressive transmission display of "Lenna"
Figure 6.4(c) Hybrid coding progressive transmission display of "Brain"
Figure 6.4(d) Hybrid coding progressive transmission display of "Finger"
Figure 6.5(a) Quadrant interfacing display of "Richard"
Figure 6.5(b) Quadrant interlacing display of "Lenna"
Figure 6.5(c) Quadrant interlacing display of "Brain"
Figure 6.5(d) Quadrant interlacing display of "Finger"
Figure 6.6(a) The compressed images by Interlaced GIF
Figure 6.6(b) The compressed images by Quadrant interlacing

R=0.125; PSNR=24.18

R=0.25; PSNR=25.23

R=0.5; PSNR=26.37

R=1; PSNR=28.67

Original
Figure 6.6(c) The compressed images by Linear quadtree embedding
Figure 6.6(d) The compressed images by Zerotree embedding
Figure 6.6(e) The compressed images by SPIHT
Chapter Seven

Conclusions

Two types of progressive transmission systems have been investigated in this thesis. The performance and computational complexity vary drastically among systems, and there is always a trade-off between the performance and complexity.

The quadrant interlacing progressive transmission system employs a region quadtree to decompose the image. By assigning linear quadtree codes on the pixels, an interlacing display format algorithm is formulated which produce a progressive display while the image is reconstructed. According to a subjective test, it performs better than the interlaced GIF.

The hybrid coding progressive transmission system employs wavelet transforms with linear quadtree coding. Two schemes to embed the information have been investigated. The first scheme is called linear quadtree embedding. The encoding and decoding processes of this scheme are fast and simple. The algorithm provides an acceptable quality result even for small amounts of information received. The disadvantage is that the compression performance is not as good as for the second scheme. The second scheme is called zerotree embedding. According to an objective test, the performance is very close to the
EZW scheme. It provides high quality results even at a bit rate of 0.125 bits per pixel.

In conclusion, the hybrid coding progressive transmission system has the highest computational load, but it offers the best performance. On the other hand, the quadrant interlacing system needs the least amount of computation, but its performance is the poorest.

Both systems are using the pyramid structure to embed the image data. This approach appears very attractive since the highest level images can be reconstructed directly and with minimum processing. Therefore, it offers a natural advantage of picture browsing capability. How to select a proper progressive transmission technique depends much on the specific application. For example, in the situation of personal computer to personal computer communication using modems, the simple quadrant interlacing technique may be the ideal candidate. On the other hand, the hybrid coding approach can be employed for its superior performance at the expense of high system complexity.

The above proposed system designs are novel, and the author welcomes further research. Future research may investigate the following topics:

1. Investigate the effect of different mother wavelets on images other than DAUB4.

2. Compare the systems' performance to other progressive image transmission methods such as progressive JPEG.
3. Simulate the systems for cellular radio channels.

4. Fine-tune the systems to optimize their performance.

5. Design program modules for an Internet browser, e.g., Netscape or Internet Explorer.
Reference


