2004

P-HIP: A Multiresolution halftoning algorithm for progressive display

Mithun Mukherjee

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
Title of thesis or dissertation: **P-ship: A MultiResolution Halftoning Algorithm for Progressive Display**

Name of author: **Mithun Mukherjee**
Degree: **Master of Science**
Program: **Electrical Engineering**
College: **KGCOE**

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

**Print Reproduction Permission Granted:**

I, **Mithun Mukherjee**, hereby **grant permission** to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: __________________ Date: **04/05/05**

**Print Reproduction Permission Denied:**

I, ____________________________, hereby **deny permission** to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part.

Signature of Author: __________________ Date: ______________

**Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive**

I, ____________________________, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.

I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: __________________ Date: ______________
Abstract

This thesis describes and implements an algorithmic framework for memory efficient, ‘on-the-fly’ halftoning in a progressive transmission environment. Instead of a conventional approach which repeatedly reconstructs the continuous tone image from memory and subsequently halftones it for display, the proposed method achieves significant memory efficiency by storing only the halftoned image and updating it in response to additional information received through progressive transmission. Thus the method requires only a single frame-buffer of bits for storage of the displayed binary image and no additional storage is required for the continuous data. The additional image data received through progressive transmission is accommodated through in-place updates of the buffer. The method is thus particularly advantageous for high resolution bi-level displays where it can result in significant savings in memory.

The proposed framework is implemented using a suitable multi-resolution, multi-level modification of error diffusion that is motivated by the presence of a single binary frame-buffer. Aggregates of individual display bits constitute the multiple output levels at a given resolution. This creates a natural progression of increasing resolution with decreasing bit-depth. Output images are shown to be comparable in terms of quality to those obtained from the conventional Floyd Steinberg error diffusion algorithm.
Acknowledgments

The work in this thesis has been my first exposure to independent research and it has been a truly educational, satisfying and rewarding experience. It has been made possible thanks to the support from a number of people.

First and foremost, I wish to express my sincere gratitude to Dr. Gaurav Sharma, my thesis advisor, for his invaluable mentorship, expert guidance and infinite patience. From him, I have learned the right approach to tackling a research question.

I would also like to thank my committee members, Dr. Sohail Dianat and Dr. Eli Saber for their valuable comments and suggestions for this work. In the past, I have benefited greatly from their talks and lectures on probability, stochastic systems and pattern recognition.

Finally, I am thankful for the love and blessings of my parents, Dipayan and Aruna, and my sister, Mallika. Their encouragement and support has been a driving force and I am truly lucky to have had this opportunity to a higher education.
Table of Contents

Abstract i
Acknowledgments ii
Table of Contents iii
List of Tables v
List of Figures vi

1 Introduction 1
1.1 Motivation ......................................................... 2
1.2 Thesis Outline .................................................. 3

2 Background 4
2.1 Background ...................................................... 5
2.2 Digital Halftoning Techniques for Binary Displays .......... 6
   2.2.1 Point processes (Screening) ............................ 6
   2.2.2 Neighborhood processes : Error Diffusion (ED) ...... 8
   2.2.3 Iterative processes ..................................... 10
2.3 Multiresolution Decomposition ................................ 10
2.4 Progressive Image Transmission ........................... 13
   2.4.1 Progressive JPEG ....................................... 13
   2.4.2 Progression orders in JPEG2000 ....................... 14
2.5 Current literature ............................................... 16
   2.5.1 Quad-tree based Multiscale Error Diffusion .......... 16
   2.5.2 Halftoning by Multiscale Dot Distribution .......... 18
3 Progressive Halftoning In-Place (P-HIP) 20

3.1 Halftoning a Multi-resolution Progression .......................... 20
  3.1.1 Independent Reconstruction and Halftoning ....................... 21
  3.1.2 Progressive halftoning in-place (P-HIP) ........................... 22

3.2 P-HIP Algorithm .......................................................... 23
  3.2.1 Multilevel error diffusion .......................................... 23
  3.2.2 Halftone display and recovery of contone ......................... 26

3.3 Memory and Computational Complexity ................................. 26
  3.3.1 Computational simplification ....................................... 29

3.4 Implementation ........................................................... 30

4 Results and Conclusion 33

4.1 Conclusion ................................................................. 34

List of References 47
List of Tables

1. Memory requirement, computational complexity and progressive transmission friendliness in Independent Reconstruction and ED Halftoning versus P-HIP .......................... 29
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Progressive image transmission with user interaction</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Bayer fill order and screen relationship</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Block diagram of error diffusion</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>2D separable DWT implementation using a Perfect Reconstruction Filter Bank</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Multiresolution decomposition of an image. ( I_0, I_1, I_2 ) are the image approximations in decreasing resolution. ( HL_0, LH_0, HH_0 ) represent the image detail at level 0. ( HL_1, LH_1, HH_1 ) represent the image detail at level 1.</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Image DWT example showing two levels of decomposition</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>Ordering of DCT coefficients in Progressive JPEG</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Partitions of the grayscale and halftone images for Wong’s multiscale dot-distribution halftoning algorithm</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>Halftoning a multiresolution progression utilizing a storage buffer for the contone image</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>Progressive Halftoning In-Place (P-HIP) Concept</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>Relation between tonal and spatial resolution for a halftone display. (a)Tonal resolution: Binary (2 levels), Spatial resolution: ( 2^k \times 2^k ). (b)Tonal resolution: 5 levels, Spatial resolution: ( 2^{k-1} \times 2^{k-1} ). (c)Tonal resolution: 17 gray levels, Spatial resolution: ( 2^{k-2} \times 2^{k-2} ). (d)Tonal resolution: 65 gray levels, Spatial resolution: ( 2^{k-3} \times 2^{k-3} ).</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>Multilevel error diffusion in P-HIP</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>Screening for display (and storage) in P-HIP</td>
<td>27</td>
</tr>
<tr>
<td>14</td>
<td>2-D Haar wavelet decomposition</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>Program flowchart for the P-HIP algorithm</td>
<td>32</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Everything in the natural world that we observe around us is continuous. Pictorial 2-D rendition of a continuous scene often calls for a discretization of this continuous space due to limitations of the rendering device, be it a printer or an LCD display or any other image output device. In particular, many image output devices are restricted to binary output states, in order to provide improved stability and reproducibility. Digital halftoning is an image processing technique by which these devices binarize a continuous tone (contone) image and exploit the inherent low pass characteristic of the human visual system (HVS) to preserve the appearance of the contone image in the output. Halftoning is commonly used in both monochrome (single-channel) and color reproduction systems. In our presentation, we will primarily be concerned with monochrome halftoning methods. The work still addresses a large class of existing and emerging bi-level output devices that are capable of only monochrome reproduction. In addition, color halftoning methods addressing specific needs are often based on corresponding black and white methods.

For monochrome (grayscale) images, the goal of halftoning is to produce a binary pattern of white and black dots, whose visual perception is close to the
visual perception of a desired contour image. The low pass nature of the HVS helps in achieving this goal by ensuring that high frequency detail in the binary image is not perceived by a human observer. The process of halftoning thus trades off spatial resolution against tonal resolution (number of gray levels).

For more than three decades digital halftoning has been a widely explored area of research, continually improving output quality and adapting to innovations in display technologies. As display technologies and data transmission standards evolve it is imperative that the enabling image processing algorithms also keep up. Particularly, in order to exploit new features and advantages from developments in display and transmission technologies further developments in halftoning algorithms are also required.

![Diagram](image)

**Figure 1:** Progressive image transmission with user interaction

### 1.1 Motivation

In this thesis, we consider the case of a progressive-by-resolution image transmission system feeding a fixed resolution binary display as shown in Figure 1. A direct approach to displaying the incoming image is to process it exactly as would be done in a continuous tone display with halftoning as an additional step just prior to display. Thus at each resolution a continuous tone representation of the image is constructed which is halftoned and displayed. To fully utilize the progressive-by-resolution feature of the transmission system as well
as to optimize the memory requirements of the display, a desirable feature of
the underlying halftoning algorithm is to halftone the incoming image detail
and incorporate it, 'on-the-fly', onto the final image display without the need
for an input image buffer. By utilizing the higher resolution of the display as a
storage element, this thesis presents a novel technique to halftone, 'on-the-fly',
a progressively transmitted image.

1.2 Thesis Outline

Chapter 2 lays out an introductory background in the evolution of halftoning
techniques and the emergence of some progressive transmission and display
methods. Also described here are two related directions of research that have
been reported in the halftoning literature. Chapter 3 describes the proposed
in-place multiresolution halftoning algorithm P-HIP (Progressive Halftoning
In-Place) and the details of its implementation. Sample output results are
presented and the thesis is concluded in Chapter 4.
Chapter 2

Background

Data transmission in a networked environment presents an interesting application for progressive digital halftoning of images. Consider the case of browsing through a high resolution image database within a PACS (Picture Archival and Communication System) system. Apart from thumbnailed images, a desirable feature of the entire user experience would be a progressive display on the display device, giving the user an opportunity to gradually sample a low resolution preview of the original high resolution image during transmission. The user has a choice to abort transmission if the image taking shape is not of interest thus saving time as well as bandwidth. Multimedia cellular phone displays operating within a typically bandwidth limited mobile environment present another possible application of this feature. The digital halftoning algorithm that processes the image prior to display must be able to support this feature. Before delving into the details of the algorithm, a solid foundation to better understanding the problem at hand can be held by looking back briefly into the origin and evolution of halftoning techniques, multiresolution decomposition and progressive image coding standards.
2.1 Background

For many centuries, the limited spatial resolution of the human visual system has been exploited to render images in engraving and weaving techniques. It was only in the middle of the 18th century when photomechanical processes were developed for halftoning images. The 1852 patent, ‘Improvements in the Art of Engraving’ by Henry Fox Talbot, describes the use of a halftone screen or mask manufactured by using ‘two or three folds of black crape or gauze’ [1]. This is by far the earliest known halftone related patent. In 1827, prior to this invention, a similar technique existed which did not use paper or wood as a display medium [2]. Instead black and white silk threads were woven using a loom operated by punch cards to generate silk images that resembled classical engravings. These silk images bore a striking resemblance to Bayer’s dispersed order dither patterns [3]. The lithographic (offset) printing industry benefited greatly from Talbot’s invention. This was followed by decades of improvement in optical halftones generated photographically with contact screens.

With the advent of electronic displays in the 1960s, noise encoding techniques were developed to reduce the contouring effect due to quantization by adding a small amount of random noise before quantization. Pioneered by Roberts [4], this technique of random dithering was developed a step further by adding a deterministic signal, an ‘ordered’ dither, to produce more visually pleasing results [3, 5].

The mid-1970s were witness to a landmark invention in the area of halftoning algorithms by Robert Floyd and Louis Steinberg [6], a simple yet effective adaptive technique called error diffusion. Over the years, there have been many modifications and improvements to the original algorithm which attempt to eliminate characteristic textures, increase efficiency and improve final output.
quality while adapting the technique to continually emerging display technologies. Since the invention of error diffusion, there have been other very successful algorithms based blue noise masks and direct binary search but the concept of error diffusion still enjoys widespread popularity and acceptance because of its simplicity and performance.

2.2 Digital Halftoning Techniques for Binary Displays

The large amount of published works in the area of digital halftoning algorithms can be classified based on their basic algorithmic approach as under [7]:

2.2.1 Point processes (Screening)

Point processes or screening techniques select the binary output value for each pixel in the output image based on a comparison of the value of each pixel in the input image against a threshold value that is determined independently of the image. The threshold may be randomly generated as in random screening or, as is more often the case, it is part of a regular array of predetermined threshold values tiled onto the input image, which is referred to as a halftone screen. This class of halftoning algorithms originates from work in the graphic arts and printing industry and from work on early computer displays that were binary.

The characteristics of the printed output are determined by the type of screens used. Clustered and dispersed dot screening are the two major types of point processes [8]. In printing systems either of these could be advantageous. In display systems, dispersed dot systems are primarily used. These will be the focus of the further discussion.

Bayer's work in 1973 on the development of spectrally based conditions for generation of a dither pattern, also known as the Bayer pattern [3] is still widely used today. Prior to this, Limb [5] and Lippel and Kurland [9] had
developed dither patterns similar to the Bayer dither screen. The following years saw a large amount of work in the development of screen design algorithms. Some bettered the ordering in the dither matrix using the pairwise exchange algorithm to improve on the visual properties of the Bayer screen by reducing the highly visible textures inherent in it [10]. Others provided alternatives to the Bayer screen based on arbitrary screen periodicities [11]. Another landmark event in the area of screen based halftoning was the invention of the Blue Noise Mask [12], a dither array designed so that the halftone image has blue-noise (high-frequency) characteristics in the frequency domain.

**Halftone Screens and Fill Order Sequences**

The process of screening makes a binary on/off decision at each pixel location by comparing that particular pixel value with the threshold at the corresponding location obtained by tiling the input image with the halftone screen. Consider a uniform image $A$ consisting of pixels all possessing the same value, $v$. To halftone image $A$ using a screen $S$, it is first tiled with $S$ in case the dimensions of $S$ are smaller than those of $A$ which is usually the case. Then, the pixel locations for which $v$ is greater than the threshold value at the corresponding screen location are all set to ‘on’ and those for which $v$ is lower than the threshold value are set to ‘off’ to produce a binary output image, $B$. As $v$ is varied, the number of ‘on’ pixels also changes according in a directly proportional manner. For example, if $v$ is increased, pixels at locations where $v$ surpasses the threshold values at those particular locations are turned on, in addition to the previously ‘on’ pixels. Thus the halftone screen may equivalently be defined in terms of a fill order in which pixels are increasingly turned on in a particular sequence defined by the fill order, as $v$ varies from zero to its maximum value, $2^{n-1}$, for an $n$ bit image.
Figure 2: Bayer fill order and screen relationship

Figure 2(a) shows the fill order for an 8 x 8 Bayer screen. The corresponding thresholds for an 8 bit linear dot are shown in Figure 2(b). For a location $i$ in the fill order sequence, the corresponding threshold value $t_i$ in the screen can be calculated by the following relation:

$$t_i = \text{round}(4 * i - \frac{i}{64})$$  \hspace{1cm} (1)

We will use these two terms, 'fill order' and 'screen', interchangeably for the remainder of this thesis.

2.2.2 Neighborhood processes: Error Diffusion (ED)

Much of the basis for this class of halftoning algorithms lies in the seminal work by Floyd and Steinberg in 1976 in which they first proposed the error diffusion algorithm [6]. In this method, the quantization error from binarizing each individual pixel is diffused ahead to neighborhood pixels that have not yet been binarized using a causal filter, hence the name error diffusion. The errors of the neighborhood pixels can cancel out the current pixel's quantization error,
so that the brightness of several pixels close together is close to the average value. This average value is perfected by subtracting the error made at one pixel from its neighbors so that the neighbors become brighter and the total brightness remains the same.

Fig. 3 shows the functional block diagram of the simple error diffusion scheme. Past errors $e(i, j)$ in binarizing an input pixel $y(i, j)$ to $g(i, j)$ are filtered and added (diffused) to the neighboring future pixels $f(i, j)$ using an error filter $h(k, l)$. This is a sequential, single pass algorithm.

While error diffusion provides an output image quality superior to that produced by screening, it has its own characteristic artifacts such as worm like strings of dots at higher gray levels and distinctive patches of texture in the midtones.

Over the years, this simple, adaptive, mean-preserving algorithm has spawned a large body of work, most of which is directed toward improving output halftone image quality by reducing or eliminating artifacts present in the halftones the algorithm generates. The work in this thesis uses the concept of error diffusion built into a progressive-by-resolution display architecture.
2.2.3 Iterative processes

The causality constraint associated with single pass neighborhood processes as described above can be relaxed using iterative processes. These are usually the most computationally intensive class of halftoning algorithms and process the image in multiple passes or iterations. These algorithms can be classified as those that 1) minimize some sort of error metric and 2) those that satisfy a set of constraints. Most metric based approaches try to minimize the frequency weighted mean square error (FWMSE) [13, 14, 15, 16, 17] based on the contrast sensitivity of the human visual system (HVS). There have also been efforts to introduce iteration into the error diffusion process to improve upon final output quality. Notable in this respect is the work by Kumar and Makur [18] which allows the use of a non-causal zero phase error diffusion filter.

2.3 Multiresolution Decomposition

The discrete wavelet transform (DWT) provides a multi-resolution decomposition of an image [19, 20]. This characterization of the wavelet transform allows the decomposition of an image into varying levels of resolution - from the coarsest to the finest.

The 2D separable DWT can be implemented with a two channel filter bank using quadrature mirror filters also known as a Perfect Reconstruction Filter Bank (PRFB) [21]. The algorithm applies a one-dimensional high and low pass filtering step to both the rows and columns to the input image. Each filtering step is followed by subsampling which results in a change in scale.

At each decomposition level there are four output images. An approximation of the input image and three detail images. The information contained in the output sub-bands of the (DWT) are:

LL - An approximation of the input image at the next lower resolution.
Figure 4: 2D separable DWT implementation using a Perfect Reconstruction Filter Bank

**HL** - Vertical image detail at current resolution.

**LH** - Horizontal image detail at current resolution.

**HH** - Diagonal image detail at current resolution.

Once a DWT decomposition is obtained, it may be further decomposed by applying the DWT to the LL sub-band. This gives a multiresolution decomposition. Figure 5 schematically shows the result of decomposition and reconstruction. At the first level of decomposition, the original image is decomposed into four subbands - one low resolution approximation and three detail components. The low resolution approximation is decomposed further into four corresponding subbands at the second level of decomposition. \( I_0, I_1, I_2 \) are the image at resolution levels 0, 1, 2 respectively, in order of decreasing resolution. The subbands \( HL_0, LH_0, HH_0 \) make up the image detail at level 0 which combined with \( I_1 \) will give \( I_0 \). Similarly, the sub-bands \( HL_1, LH_1, HH_1 \) make up the image detail at level 1 which combined with \( I_2 \) will give \( I_1 \). This process may be continued into higher levels of decomposition i.e. lower levels of resolution.
Figure 5: Multiresolution decomposition of an image. $I_0$, $I_1$, $I_2$ are the image approximations in decreasing resolution. $HL_0$, $LH_0$, $HH_0$ represent the image detail at level 0. $HL_1$, $LH_1$, $HH_1$ represent the image detail at level 1.

Thus, additional of detail to the decomposed low resolution approximation image leads to a representation of the image at the next higher resolution. This is the basis of progressive-by-resolution transmission in the context of wavelet based compression methods such as JPEG 2000.

An example two level decomposition is shown in Figure 6, where the blocks of each image are in correspondence with those in Figure 5. Figures 6(a), 6(b) and 6(c) correspond to Figures 5(a), 5(b) and 5(c) respectively.

![Figure 6](image_url)

Figure 6: Image DWT example showing two levels of decomposition
2.4 Progressive Image Transmission

The advantage of a progressive display is that if an image is being viewed on-the-fly as it is transmitted, one can see a preview approximation to the whole image very quickly, with gradual improvement of quality as time progresses. This is a more favorable feature as compared to a slow top-to-bottom raster form display of the image and is especially desirable when transmitting high resolution images. For example, a user may abort transmission midway if the preview makes it apparent that the image is not of interest, thus saving bandwidth and time.

2.4.1 Progressive JPEG

Progressive JPEG is the JPEG equivalent of the interlaced GIF mode. There are two ways to go about implementing this - spectral selection and successive approximation. A fundamental operation within JPEG compression is the block based Discrete Cosine Transform (DCT) which decomposes the input image into 8x8 size blocks consisting of one DC coefficient at the top left corner and 63 AC coefficients representing spatial frequency. The farther the distance between the DC coefficient and a particular AC coefficient, the higher is the spatial frequency represented by it.

The spectral selection approach borrows from the interlaced GIF mode in that it traverses the coefficients in each DCT block in an interlaced manner as shown. The first scan simply picks out the DC components in each blocks. Successive scans add the AC coefficients from one or more spectral bands in an interlaced order.

On the other hand, the successive approximation technique for progressive JPEG improves upon the detail successively by increasing the number of bits that represent the DCT coefficients in each scan. It initially begins with a low
Figure 7: Ordering of DCT coefficients in Progressive JPEG

bit-rate representation and then builds upon the number of bits progressively. For both techniques, all the coefficients need to be stored until the last scan is reached. In terms of storage capacity, this is not always optimum.

2.4.2 Progression orders in JPEG2000

A fundamental component of the JPEG2000 feature set is progressive transmission and this is achievable through four different dimensions of progressivity. Depending upon the application, the JPEG2000 code-stream can be ordered for progression in four dimensions as: Quality, Resolution, Spatial Location and Component. As more data is received during image transmission, the rendition of the displayed imagery is improved in one of these manners. Quality as the first dimension of progressivity is implemented using an increasing the bit rate. As more data are received, the number of bits representing a sample increases
and the image quality is improved. Improving quality is then a simple matter of decoding more bits. Resolution is the second dimension of progressivity. This is especially useful in thumbnail image browsing applications. In this type of progression, the first few bytes are used to represent the thumbnail of an image. As more bytes are received, the resolution of the image is increased by factors of two on each side and eventually the full size image is obtained. The third dimension of progressivity is spatial location. This is particularly useful for applications in which memory is at a premium such as printers. The imagery can be received in a raster fashion, from top to bottom. Spatially progressive code streams can be created on the fly without any need for buffering the entire image or the compressed code stream. The final dimension of progressivity is the component with JPEG2000 supporting more than 16384 'components'. Typically images are 1 component (grayscale), 3 component (e.g. RGB, YUV etc.), or 4 components (CMYK). Overlay components such as additional text or graphics are also possible. Component progression controls the order in which the data corresponding to different components is decoded. The various progression orders of JPEG2000 can be interleaved within a single code-stream.

The method of progressivity can change through the code stream. For example, the initial portion of the code-stream may contain the low quality, grayscale information for an image followed by added quality and color information. The next incoming data might then provide information for higher resolution. Thus, the imagery can be improved in many dimensions as data are received and depending upon the viewers application or discretion, data might be selectively transmitted.

In this thesis we will primarily consider the progressive by resolution component. This is the most natural order of progression for the DWT decomposition used in JPEG2000 is apparent from Figures 5 and 6. Note that in this order
of progression the number of DWT coefficients needed for each successive level of detail increases by a factor of four over the previous level. From Figures 5 and 4, we can see that the final level of detail coefficients comprise $\frac{3}{4}^{th}$ of the total DWT coefficients$^1$. Thus substantial savings in bandwidth may be obtained even if the transmission is terminated only at the final stage of the multi-resolution progression.

2.5 Current literature

Existing literature in the area of halftoning does not address the specific scenario of halftoning for display in a progressive by resolution transmission environment. There are, however, two related directions of research that have been reported. These are described below.

2.5.1 Quad-tree based Multiscale Error Diffusion

The first algorithm, a quad-tree based multiscale error diffusion technique proposed by Katsavounidis and Kuo [22] utilizes a 'maximum intensity guidance' rule to assign dots in the brightest region of the given image and then diffuses the quantization error iteratively. An important component of this technique is the image quadtree which is a pyramidal structure consisting of image arrays, each a representation of the original image at different resolutions. The image array at the finest resolution, the original image itself, lies at the base of the pyramid and arrays visualizing coarser resolutions are built upon it toward the tip of the pyramid. Each pixel element of an array is simply the sum of the intensities of four pixel regions in the corresponding $2 \times 2$ pixel region in the previous array of finer resolution. If we consider square images of dimension $N \times N$ and $r = \log_2 N$, then this relation can mathematically be defined as,

$^1$These do not necessarily correspond to $3/4^{th}$ the of the data in the compressed representation.
\[ X_k(i_k, j_k) = \sum_{i=0}^{1} \sum_{j=0}^{1} X_{k+1}(2i_k + 1, 2j_k + 1) \]

\[ i_k, j_k = 0, \ldots, 2^k - 1; k = r - 1, \ldots, 0. \]

(2)

The tip of the quadtree pyramid is thus a single value representing the sum of the intensities of all the pixels in the original image. Let \( X, B \) and \( E \) be the arrays of dimension \( N \times N \) corresponding to the input image, output binary image and the difference (error) image respectively. The goal of the algorithm is to bring the output image pyramid to resemble the original input image pyramid as closely as possible. At the beginning, the output image pyramid is blank \( (B_k = 0) \) and therefore the error image pyramid \( E_k \) is identical to the input image pyramid \( X_k \). Beginning at the tip, the algorithm traverses down the error image pyramid following a ‘maximum intensity guidance’ rule which states that, in transitioning from a parent pixel region at a particular level of resolution to one of its four child pixel regions or quadrants at the next finer level, the quadrant possessing the maximum intensity shall be chosen. The algorithm thus travels top-to-bottom in a greedy way. Upon reaching the base of the pyramid, the finest resolution, we arrive at a location corresponding to a single pixel in the original input image. A white dot is introduced at this location in the output image \( B \) and the error is diffused to the neighbors of that pixel. The entire image pyramid is then updated upward with the new error values. This process continues iteratively until the tip of the error tree is less than 0.5, which implies that the global error is bounded in absolute value by 0.5.

To construct the image quadtree, the algorithm requires the input contone image in its entirety, thereby rendering it unsuitable for implementation within a progressive transmission scheme. Additionally, although it produces very good
results, its iterative nature as well as the storage requirements for the image quad-tree may pose hurdles in its implementation within such a scheme.

2.5.2 Halftoning by Multiscale Dot Distribution

The second algorithm published by Wong [23] attempts to match the average gray levels of the grayscale image and the halftone image over local neighborhoods, where the neighborhoods are taken with respect to a quad-tree structure.

Consider a grayscale image $X_{m,n}$ of dimension $N \times N$ assuming that $N=2^l$, where $0 \leq X_{m,n} \leq 1$ for all $(m,n)$. Let $B_{m,n} \in (0, 1)$ be the corresponding halftone image. Let the region of support for the image be given by $R_0^{(0)} = (i, j): i=0, 1, ..., N-1; j=0, 1, ..., N-1$. To start off, the algorithm enforces a global mean requirement i.e the average gray level of $X_{m,n}$, the grayscale image and that of $B_{m,n}$, the corresponding halftone, are equal upto rounding errors over the region $R_0^{(0)}$. The total number of white pixels in the halftone represent this average gray level. The following steps then perform a recursive partitioning of $R_0^{(0)}$ using a quadtree structure, where each region is partitioned into four equal quadrants at each step as shown in Figure 8.

![Figure 8: Partitions of the grayscale and halftone images for Wong’s multiscale dot-distribution halftoning algorithm](image)

At each stage the number of black and white pixels are redistributed over the four quadrants to satisfy the mean requirement over that region. The algorithm
continues until the last stage where each quadrant represents one pixel. The accuracy of the halftone is up to the level of rounding errors.

Again, this algorithm begins with the image at its final resolution and is not intended for a progressive by resolution transmission environment.
Chapter 3

Progressive Halftoning In-Place (P-HIP)

Progressive Halftoning In-Place (P-HIP) is a novel technique for halftoning a multiresolution progression of image data. It combines multilevel error diffusion and screening to progressively halftone and render a grayscale image with gradually increasing resolution. The strength of this algorithm lies in its memory efficiency. In this chapter, we first describe a straightforward approach to halftoning a multiresolution progression. This is followed by a description of the P-HIP algorithm, its complexity, and its implementation. Finally, we present sample results and compare the performance of P-HIP with the straightforward method.

3.1 Halftoning a Multi-resolution Progression

Consider a multiresolution image being communicated in a progressive-by-resolution system to a bi-level display. In order to avail the benefits of progressive transmission, the bi-level halftone display must constantly be updated to display the available image information. Since power and memory are both
precious commodities in any portable display, for it to be capable of rendering progressive imagery, a halftoning algorithm that utilizes minimum power and memory is desirable.

3.1.1 Independent Reconstruction and Halftoning

A straightforward approach to halftoning a multiresolution progression is to handle the multiresolution progression and the halftone display independently of each other. At each stage in the transmission, the available image data is utilized to recreate the best possible contone representation which is halftoned and displayed. This contone representation is preserved for further refinement as additional data at finer resolution is received. This is shown in Figure 9.

![Figure 9: Halftoning a multiresolution progression utilizing a storage buffer for the contone image](image)

The reconstructed image is stored at its current resolution in a buffer in memory. It is then halftoned and displayed (with potential scaling to suitable size). When the next level of detail is received, it is added to the current image thereby increasing its resolution. The higher resolution contone is stored in the buffer and the same steps are performed recursively until the final resolution of the image is attained and displayed.
3.1.2 Progressive halftoning in-place (P-HIP)

In a situation where memory is at a premium, the additional memory required to store the entire contone image can be a significant system overhead. This is particularly true for high resolution/large scale devices where the large image dimensions can require a significant amount of storage memory. Since the frame buffer (or other memory) used for storing the displayed image already contains an approximate representation of the contone image, it can be used instead of a separate memory buffer, to recover the original contone from the displayed halftone for the next stage of progression. It is also possible to save memory by storing only a compressed representation of the image but this entails an additional computational burden as the image reconstruction at each progression of resolution must be recomputed from scratch. This scheme is illustrated in Figure 10 below.

![Diagram of Progressive Halftoning In-Place (P-HIP) Concept]

A halftone display trades off spatial resolution against the number of gray levels. If one interprets the gray level as a spatial average then there is a natural progression of increasing tonal resolution with decreasing spatial resolution in a halftone display. Consider a binary display of size $2^k \times 2^k$ pixels. If one considers $2 \times 2$ blocks of pixels within this image, each block as a whole may represent 0-5 gray levels i.e. the number of ‘on’ pixels may vary from 0 to 4. Thus, by reducing
the spatial resolution by half it is possible to increase the tonal resolution from binary (2 levels) to 5 gray levels. The image dimensions have fallen from the original size of $2^k \times 2^k$ to an equivalent size of $2^{k-1} \times 2^{k-1}$. As a generalization, decreasing the spatial resolution by a factor $r$ by picking blocks of size $2^r \times 2^r$ shall lead to an increase in tonal resolution from 2 gray levels to $2^{2r} + 1$ gray levels. The spatial resolution thus reduces from an original image size of $2^k \times 2^k$ to a final image size of $2^{k-r} \times 2^{k-r}$. The inherent averaging or low pass involved in this process may be considered an approximation to that within the human visual system. The image decomposition in spatial terms assumes the structure of a quadtree. Figure 11 below shows a graphical example of this inversely proportional relationship between tonal resolution and spatial resolution.

3.2 P-HIP Algorithm

P-HIP operates in two steps - a progression based multilevel quantization step followed by a fill order based halftoning operation on this quantized output for updating the display. The highest level of decomposition corresponds to the lowest resolution or the first stage of progression and vice versa. This convention shall be used in the following description of the algorithm.

The progression based multilevel error diffusion step is shown in Figure 12. The multilevel image at the current stage of progression, or decomposition level $(i+1)$, is combined with the incoming image detail for the level $i$. Multilevel error diffusion is performed on the image produced to obtain the higher resolution but lower bit-depth image at the the next, i.e. $i^{th}$, level of progression.

3.2.1 Multilevel error diffusion

The coarseness of quantization within the multilevel error diffusion block is gradually increased with increasing resolution. i.e the degree of quantization is finer at lower resolutions and coarser at higher resolutions. In other words,
Figure 11: Relation between tonal and spatial resolution for a halftone display.
(a) Tonal resolution: Binary (2 levels), Spatial resolution: $2^k \times 2^k$. (b) Tonal resolution: 5 levels, Spatial resolution: $2^{k-1} \times 2^{k-1}$. (c) Tonal resolution: 17 gray levels, Spatial resolution: $2^{k-2} \times 2^{k-2}$. (d) Tonal resolution: 65 gray levels, Spatial resolution: $2^{k-3} \times 2^{k-3}$. 
Figure 12: Multilevel error diffusion in P-HIP
the number of gray levels or tones are higher at lower resolutions and vice versa. With each progression of incoming detail, the quantization step size is quadrupled to mimic a multiresolution quadtree. At the final stage, when the last remaining image detail has been received, the quantization is at its coarsest, thus representing the equivalent of a simple bi-level error diffusion halftoning operation. This process is well matched to the inherent trade-off between the tonal and spatial resolutions as illustrated in Figure 11.

### 3.2.2 Halftone display and recovery of contone

Following the multilevel error diffusion, a bi-level representation of the lower bit-depth multilevel image is displayed using a halftone screen as shown in Figure 13. An $n^{th}$ order Bayer fill pattern is used to display the individual pixels in a suitable bi-level format where $2^n + 1$ is the number of gray levels in the image. Aggregates of individual display bits constitute the multiple output levels at a given resolution. The multilevel contone image can be recovered from this bi-level representation by simply collecting these aggregates. This recovered contone image is then combined with the next level of image detail. This process continues till level 0, at which point all available image detail has been received and the image has been rendered at its final resolution.

In this sense, the halftoning for the purpose of display may be considered simply as the storage of the multilevel error diffusion output in the bits afforded by the corresponding area in the binary display.

### 3.3 Memory and Computational Complexity

A major advantage of the P-HIP algorithm is the saving in memory achieved by utilizing a single frame buffer as compared to the method of independent reconstruction and halftoning which requires separate memory for storage of the contone image at its current resolution. With P-HIP, at any given point in
Figure 13: Screening for display (and storage) in P-HIP
time, only a single display size buffer is required to store the halftone as well as to display it. This can be a useful feature in a portable device display where memory is at a premium. Since a single buffer is used for display as well as storage, for a display size of $M \times N$ pixels, this accounts for a memory saving of $M \times N$ bytes (assuming 8 bit images).

Recovery of the contone image from the fill order screened output halftone can impose an additional computational burden in return for the savings in memory. This computation involves counting the number of ‘on’ pixels in each block of halftone output corresponding to a single pixel in the contone image. A significant reduction in this computation can be achieved by incorporating a binary search operation within this process. For a screen size of $N \times N$, straightforward counting requires a maximum of $N^2$ operations. Instead, a binary search within the screen using the known fill order for the screen will reduce this figure to $\log_2(N^2)$ operations. We begin the search at the midpoint of the tonal range. For example, for an 8 bit image (256 gray levels), we begin at the midpoint of the tonal range i.e. 128. If the pixel in the halftone block corresponding to the location marked 128 in the fill order is ‘on’, it is obvious that the pixel value in the contone image being recovered is greater than 128. Therefore, we restrict the search to the range of values between 128 and 255. If not, we restrict the search to values between 0 and 127. This process is repeated recursively until the final value is obtained. The use of suitable hardware that automatically interprets the bits within the chosen block of binary values as a single multilevel value may allow further improvements in efficiency.

Table 1 above compares P-HIP against the method of independent reconstruction and halftoning using error diffusion described in Section 3.1.1 in terms of memory requirement, computational complexity and suitability in a progressive display environment. As emphasized already, P-HIP provides improved
Table 1: Memory requirement, computational complexity and progressive transmission friendliness in Independent Reconstruction and ED Halftoning versus P-HIP

<table>
<thead>
<tr>
<th></th>
<th>Independent Reconstruction and ED Halftoning</th>
<th>P-HIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory requirement in bits</td>
<td>$8 \times N \times N$</td>
<td>$N \times N$</td>
</tr>
<tr>
<td>(for $N \times N$ 8 bit images)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computational complexity</td>
<td>$K \times L^2$</td>
<td>$4/3[1 - (1/4)^K] \times L^2$</td>
</tr>
<tr>
<td>$(L \times L$ image display)</td>
<td>$K$ levels of progression</td>
<td></td>
</tr>
<tr>
<td>Progressive transmission</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>friendly</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

memory efficiency by utilizing the display buffer as a storage medium from which to recover the original contone image instead of a separate storage memory. Thus, for an 8 bit image of size $N \times N$, P-HIP requires only a memory buffer of $N \times N$ bits as compared to $8 \times N \times N$ bits required if the independent reconstruction and ED halftoning approach is adopted. Table 1 also lists computational complexities of both approaches in terms of the number of pixels processed. Finally, it is worth mentioning that P-HIP is more suitable within a progressive transmission environment than independent reconstruction and halftoning.

3.3.1 Computational simplification

Section 3.1.2 described the multiresolution progression of a halftone display based on the relationship between increasing tonal resolution (increasing block sizes) and decreasing spatial resolution. Beyond a certain block size, the number
of pixels in each block may equal the maximum number of gray levels in the original image. Since this block size represents all the gray levels in the original image, larger block sizes beyond this stage may be considered redundant. Thus, the block size may be fixed beyond this point to still lower the computation required. For example, for an 8 bit image (256 gray levels), the block size may be fixed at $16 \times 16$ beyond the fourth level of decomposition.

### 3.4 Implementation

A simple software application to demonstrate the P-HIP algorithm has been implemented using Visual C++ and Microsoft Foundation Classes (MFC).

For illustration we consider a simple multiresolution decomposition based on the Haar wavelet transform. To simulate a progressive-by-resolution image transmission environment the input image is decomposed using an elementary Haar wavelet transform. The computation of the Haar wavelet coefficients is illustrated in Figure 14.

![Figure 14: 2-D Haar wavelet decomposition](image)

The addition of detail at each stage (level) of reconstruction represents the progression of resolution. Beginning at the first stage of reconstruction the low

---

1With the DWT multi-resolution decomposition, we are not assured that lower-resolution approximations obtained from the DWT of an 8-bit image also satisfy the 256 gray level assumption but the quantization error is usually quite small in practice.
resolution image undergoes multilevel error diffusion to yield an output image with fewer gray levels. The quantized image is then displayed on screen using a Bayer fill pattern whose order depends on the bit-depth of the multilevel image. For example, a 64-level image is represented using an $8 \times 8$ Bayer fill pattern. The image detail for the next stage of reconstruction is added and the process is repeated until the last stage, where the low resolution image is simply error diffused to give the final halftoned output image. Figure 15 shows the algorithm flowchart.
Figure 15: Program flowchart for the P-HIP algorithm
The proposed algorithm provides a memory efficient solution for halftone display in a progressive image transmission environment. Instead of halftoning a received image once data reception is complete, it presents the halftoned image gradually 'on the fly' as image detail is being received, thus exploiting the very nature of progressive image transmission. A straightforward approach to achieving this is to independently halftone and display the incoming contone image at its current resolution and store it in memory. Incoming image detail is combined with this stored contone image and subsequently halftoned and displayed again and the process repeats. P-HIP offers a saving in memory requirement by utilizing the display buffer as a storage medium from which to recover the current resolution contone image, thus doing away with the additional storage memory required.

We utilize a set of 8 bit grayscale test images shown in Figures 16, 17 and 18 to illustrate the output of the P-HIP algorithm. Figures 16 and 17 are of dimensions 256 × 256 pixels and are used to show more clearly the textures obtained from the halftoning process. A larger image, Figure 18, of dimensions 1024 × 1024 pixels is more suitable for demonstrating the typical application
scenario of progressively transmitting high resolution imagery.

The following set of images in Figures 19, 20, 21, 22 illustrate an eight stage halftone progression (corresponding to an 8 bit input image) using the P-HIP algorithm beginning with the lowest resolution at decomposition level 8 and ending with the final halftoned image at its final resolution at decomposition level 1. (Note: Due to scaling, printed reproductions of the results may not completely and accurately represent actual screen output.)

The results are also illustrated by comparing the output of the final stage of the P-HIP algorithm to that of the conventional Floyd Steinberg error diffusion algorithm. Figures 23(a), 23(c) and 23(b), 23(d) show halftones generated using the P-HIP algorithm and the Floyd Steinberg error diffusion algorithm respectively. From the displayed images we can see that the final images produced by the two methods are visually comparable in terms of quality.

It is worth noting that the examples presented in Figures 19, 20, 21, 22 are meant for illustrating the algorithm. The major benefit of progressive transmission is realized for larger sized images where the image content is still recognizable after several levels of decomposition. We illustrate this by means of an example using the 'Dog' image of larger dimensions (1024 × 1024 pixels) in Figures 24, 25, 26 and 27. In this case, the initial displayed halftones at lower stages of progression (higher levels of decomposition) are more meaningful and convey greater information about the final image.  

4.1 Conclusion

This thesis has proposed a new system for halftone display intended for use in a progressive transmission environment. The algorithm uses a combination of multilevel error diffusion and screening to progressively render incoming image

1The author acknowledges the tonal bias introduced just prior to the final output as a subject of further investigation.
data in a resolution progression on a bi-level display, using only single binary frame-buffer to store the displayed halftone image and to recover the progressive resolution images from the incoming data. The algorithm thus offers significant memory efficiency since separate storage is not required for a contone image corresponding to the progressive transmission. The final halftones produced by the algorithm are comparable in quality to the those from the generally accepted Floyd-Steinberg error diffusion algorithm.
Figure 16: The original ‘Lena’ test image (256x256)

Figure 17: The original ‘Cameraman’ test image (256x256)
Figure 18: The original 'Dog' test image (1024x1024)
Figure 19: P-HIP halftones for the first four progression stages of the 'Lena' image
Figure 20: P-HIP halftones for the last four progression stages of the 'Lena' image
Figure 21: P-HIP halftones for the first four progression stages of the 'Camera-man' image
Figure 22: P-HIP halftones for the last four progression stages of the 'Camera-man' image
Figure 23: Comparison of final images from P-HIP against Floyd-Steinberg error diffusion
Figure 24: Fifth level of progression of the 'Dog' image
Figure 25: Sixth level of progression of the 'Dog' image
Figure 26: Seventh level of progression of the 'Dog' image
Figure 27: Eighth level of progression of the ‘Dog’ image (Final halftoned image)
List of References


