Journal of Electronic Imaging Vol. 8 No. 2, p.133-144, 1999.

Adaptive Cluster Dot Dithering

Hagit Z. Hel-Or[‡], Xuemei M.Zhang, and Brian A. Wandell

Dept. Of Psychology
Stanford University
Stanford CA 94306

E-mail: {gigi,xmei,brian}@white.stanford.edu

Abstract

Cluster Dot Dithering is one of the most common halftoning techniques. It is fast, low in complexity and allows for variability and inconsistencies in point spreads in printer outputs. Determination of the basic dither cell size is critical for the quality of the halftoning. There is a basic tradeoff between large and small cell sizes: spatial resolution vs gray tone resolution. Large dither cell sizes produce good tone resolution but poorly reproduce spatial details in the image. Small dither cells, on the other hand, produce fine spatial resolution but lack the tone resolution which produces smooth gray tone gradients in halftone images. Typically, Cluster Dot Dithering assumes a pre-defined dither cell size that compromises between fine detail reproduction and good gray tone reproduction. It is clearly advantageous to allow variability in the dither cell size using small cell sizes in image regions of fine details and using large cell sizes in image regions where gray tones are to be accurately reproduced. In this paper, we introduce and discuss several *Adaptive Dithering* techniques based on Cluster Dot dithering.

Key words: Dithering, Halftoning, Printing, Adaptive Dithering, Cluster Dot Dither.

1 Introduction

Even with the advent of advanced printing and display technologies, many devices are still bi-level and reproduce gray tone images using only black and white pixels (dots). The specific arrangement and density of the dots give rise to perception of gray tones. The process of converting a gray tone image to such a binary representation is called *Halftoning*.

There are various approaches to Halftoning. One of the basic approaches is the class of *Dithering* methods. The Dithering methods are based on thresholding of the image pixel values according to a predefined thresholding pattern. Classically, the dithering pattern is determined by the collection of threshold values in a square or rectangular block of pixels. This block of threshold values is called the Dither Cell. This cell pattern is then duplicated creating a grid of cells which tessellates the original image and is used as the thresholding pattern of the image (see Figure 1). A number of dither patterns have been introduced having various characteristics in terms of the halftoning patterns they produce. Several well known methods for producing dither cell patterns are the Void and Cluster, Bayer, and Cluster Dot (see [15, 19] for a review).

[‡]Current address: Dept of Computer Science, Haifa University, Haifa 31905, Israel.

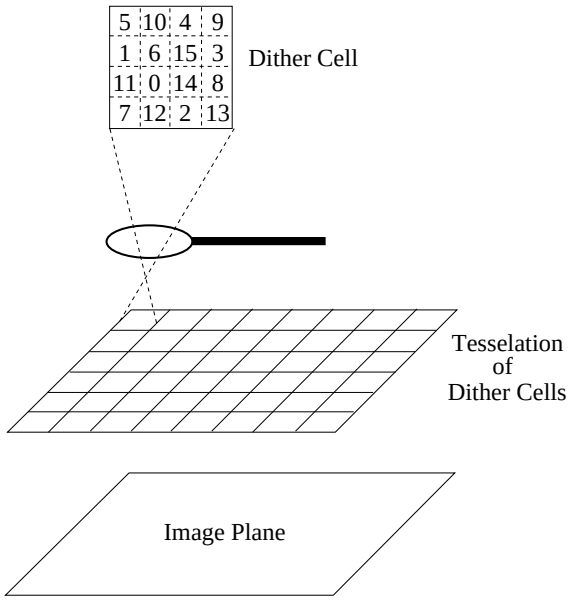


Figure 1: A tessellation of the image is created by tiling the basic dither cells of threshold values in a grid fashion. This tessellation is used to threshold a gray tone image.

Other approaches to halftoning include the *Error Diffusion* methods in which a continuous scan of the image pixels, produces the halftoned image pixel-by-pixel by propagating accumulated error and minimizing an error metric [5]. Another approach uses the Binary Search method (BNS) which iteratively updates the value of the halftone image pixels to minimize a global error measure [1].

There are practical considerations in choosing the halftone method in printing technology. Complexity and run time of the halftoning algorithm is critical for obtaining reasonable printing speeds. More important, however, are the constraints imposed by the mechanics and technology of the printing device, namely, the inability to accurately reproduce single dots. It has been shown that point spreads in printer outputs widely vary from printer to printer. Usually the point spread exceeds the abstract boundaries of a pixel in the reproduced digital image, requiring special considerations in calibrating the output device to produce correct gray tones. It can be shown that variability is reduced when printing clusters of dots rather than single dots. Given these constraints and limitations, the preferred halftoning methods for printers are those that reproduce gray tones using patterns that contain clusters of dots rather than distributions of single dots. Thus quite often the Cluster Dot Dithering (Chapter 5 in [19]) is used which creates gray tones using patterns of clusters of dots rather than using the single dot patterns that arise when using Bayer [2] or Void and Cluster [18] dithering patterns. Cluster Dot Dithering, as well as other traditional halftone screening methods, have the additional advantage over the Error Diffusion methods and the BNS methods in terms of simplicity and time complexity. In this paper we consider the issues discussed above and deal only with Cluster Dot Dithering as the halftoning scheme.

When designing a cluster dot dither pattern, a preliminary decision is usually made that defines the size of the basic dither cell. Given the size of the dither cell, the cluster dot pattern of thresholds is created for such a cell. This patterned cell is used repeatedly over the original image during the dithering process. In other words, a tessellation of the image is created by tiling the dither cells in a grid fashion (see Figure 1). Two examples of threshold patterns for a single cell of size 6x6

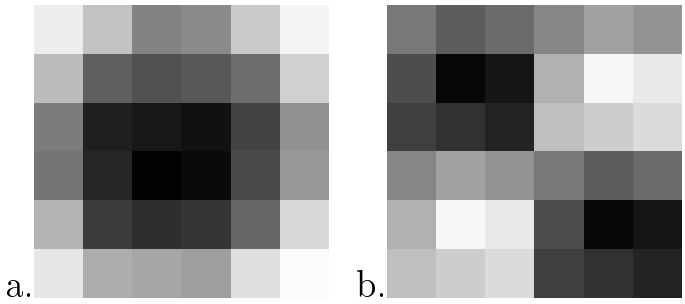


Figure 2: Examples of threshold patterns in Cluster Dot dither cells
a) Creates a horizontal/vertical pattern of clusters. b) Creates a 45° pattern of clusters.

are shown in Figure 2. The cell pattern of Figure 2a creates a horizontal and vertical pattern of clusters and that of Figure 2b creates a 45° pattern of clusters. A Cluster dot dither cell of size $n \times n$ pixels is denoted Cn.

Determining the basic dither cell size is critical for the quality of the halftoning outcome. There is a basic tradeoff between large and small cell sizes, namely spatial resolution vs gray tone resolution (see Chapter 10 in [4]):

- **Small** cell size - Using a small cell size, the maximal clusters are small and allow for a high resolution reproduction that captures the fine details of the original image. However, since the number of threshold levels is determined by the cell size, dithering with small cell sizes allows only a small number of threshold levels and accordingly only a small number of possible gray tones.
- **Large** cell size - Large cell sizes, typically produce large clusters that tend to consume fine image details. However, large cells contain a large number of threshold values and produce a larger number of gray levels. It has been shown that the required number of grey levels is inversely proportional to spatial frequency ([11]). Thus in regions of smoothly varying gray tones (low spatial frequency), dithering with large cells will appropriately reproduce the original image by producing smoothly varying gray tones, whereas, dithering with a small cell size will produce false contouring due to the inadequate number of gray tones available.

Most Dithering algorithms choose a dither cell size that compromises between fine detail reproduction and good gray tone reproduction. It is easily seen, however, that a better halftone quality can be obtained if we allow variability in the dither cell size; using small cell sizes in regions of an image in which fine details should be preserved and using large cell sizes in regions of an image in which gray tones are to be accurately reproduced (specifically those regions with smoothly varying gray tones). *Adaptive Dithering* or *Variable Dithering* denote dithering techniques that vary the cell size locally for every image region.

In this paper we discuss a number of approaches to adaptive cluster dot dithering, allowing local variations in dither cell size.

2 Variable Dithering

As described in Section 1, there is a tradeoff between using large and small cell sizes for the Cluster Dot dithering. Areas in the image having fine details are better reproduced using small cells which preserve the image details. Slowly changing areas in the image are better reproduced using large cells which reproduce the gray tones more accurately and smoothly.

Areas of fine details and areas of slowly varying gray tones can be determined by the local frequency in the image. Local high frequencies indicate fine details and require small dither cell sizes for dithering. Local low frequencies represent slowly changing regions in which gray tone reproduction is important and large cell sizes are appropriate for dithering.

In this paper we use a measure of *Busyness* which is dependent on local image frequencies, to determine the level of detail in every pixel, and accordingly determine the appropriate cell size for a given region in the image.

We measure the busyness of an image by convolving the image with a Laplacian kernel (given in Table 1) and averaging the absolute values obtained over a region (16x16 pixels) to produce a single busyness value for every pixel in the image. Values are large in image regions with high frequency content and low in image regions of low frequency content.

0	-2	-4	-2	0
-2	-4	8	-4	-2
-4	8	16	8	-4
-2	-4	8	-4	-2
0	-2	-4	-2	0

Table 1: Laplace kernel used in computing the busyness of an image.

Any number of other busyness measures may be chosen and substituted for the measure described here. We found that the busyness measure we have chosen provides a wide enough range of values that vary slowly over the image (due to the large averaging window). This measure is adequate for use in determining local dither cell size in the Variable Dithering algorithms described in this paper.

In the rest of this paper we refer to the busyness value as the values obtained using the procedure described in this section.

3 Previous Work

Although the tradeoff between spatial resolution and gray tone reproduction relating to halftone dithering (as discussed in Section 1) has long been known and accepted, relatively few studies have actually approached this problem.

Adaptive control of threshold values in error diffusion has been proposed in [7]. Threshold values

in dithering and error diffusion have been adaptively controlled at edges and other spatial features to enhance spatial resolution at image boundaries [6, 14]. However these do not change cluster size and although they produce enhanced details at edges, they do not necessarily improve gray tone resolution at low frequency regions of the image.

In [13, 16] dither cell threshold patterns were adaptively controlled to produce oriented dither patterns along image edges and gradients. The oriented dither cells were chosen from a preselected set or dynamically computed. In both cases the dither cell size remained constant thus there is no adaptability in terms of gray tone resolution.

Adaptively varying dither cell size was introduced in [3] and recently in [20]. In [20] halftoning is performed by scanning the original image using a space filling curve and adaptively defining dither cells as segments of varying length along the scanned path. Thus dither cells are of varying size and shape determined according to local gradients in the image. For every such dither cell, the threshold values in the cell are determined dynamically by considering the average and the peak gray scale value in the image region corresponding to the dither cell. The dither cells produced in this method may be elongated and not necessarily convex (see discussion in Section 5.2). Additionally, the authors restrict the cell size to a maximum of 11 pixels which does not allow large variability in cluster size and in gray tone resolution. Recently another similar approach was suggested [22].

In [3] a tessellation of the image plane was produced using a quadtree structure which recursively divides the image into quadrants. The number of levels in the recursive division is determined adaptively for different regions of the image. The quadtree produces a tessellation of the image into (square) cells of varying size. Each cell is then considered a dither cell and threshold values are mapped into it. The threshold values are taken from a set of dither cells which have been previously defined. One such set suggested by the authors is a cluster dot dither pattern. Due to the nature of the quadtree tessellation, the dither cell sizes are restricted to a limited set (namely to cell sizes of $2^m X 2^m$, for positive integer m). The transition between cell sizes is large and may give rise to textural boundaries in the halftoned image (see discussion in Section 4.1).

Iterative approaches to halftoning, reflect adaptivity, by evolving local parameters that are dependent on the image. In [8] an iterative approach to halftoning is suggested which is based on models of human vision which display variable sensitivities to image content. In [9] both a human vision model and a printer model are considered. In [21], an iterative approach which evaluates local gray value and local gradients to produce frequency preserving halftones is presented. Another approach [17], uses Artificial Neural Nets to iteratively calculate the halftoning pattern based on local gradients computed using the Wavelet Transform.

4 Variable Dithering: Pixel-by-Pixel

A straight forward approach to obtaining variable dithering is to actually combine several halftoned images obtained using different cluster dot cell sizes. The simplest approach combines the dithered images on a pixel by pixel basis.

The design of this approach is shown in Figure 3. The original monochrome image is halftoned using a number of different sized cluster dot cells producing a number of dithered images.

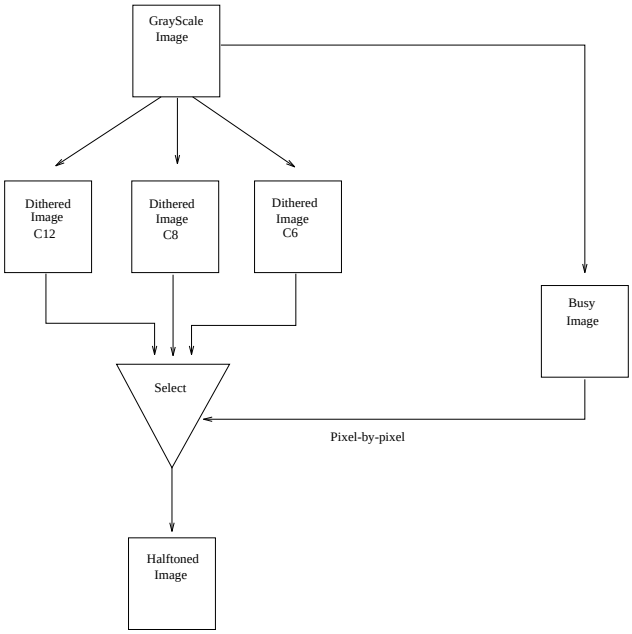


Figure 3: Outline of the pixel-by-pixel variable dithering method.

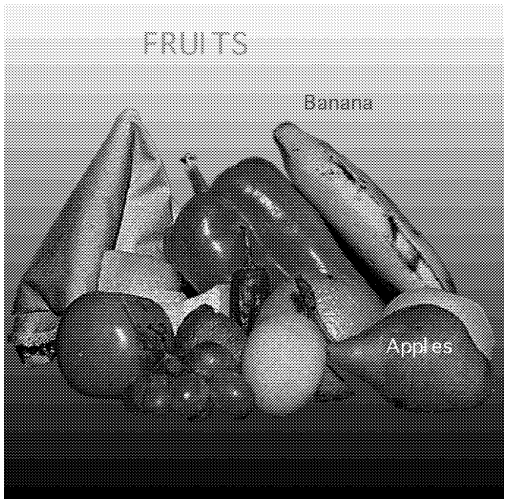
The final Variable Dithered halftoned image, is created by combining pixels of the various dithered images into a single output image. For every pixel in the final halftoned image, the corresponding pixel in the busyness image is evaluated to determine which of the dithered images should be used as a source for the current halftone pixel. If the busyness value is high then the current pixel is taken from the dithered image which was produced from the small cluster dot cell. If the busyness value is low, then the current pixel is taken from the dithered image which was produced from the large cluster dot cell.

4.1 Results

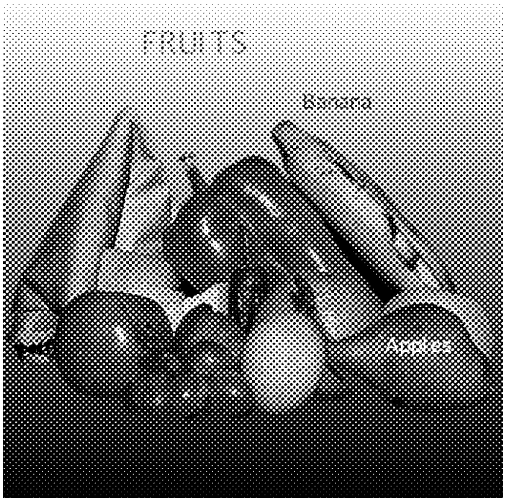
Figure 4 displays an example of the variable dithering method described in this section. The original grayscale image is 1024x1024 pixels and includes regions of high busyness values (text regions, kiwi slice, etc.) and regions of low busyness (melon and orange surfaces). The dithered images used to create the variable dithered image, were created using cluster dots C6, C8, C10 and C12 (dither cells of size 6x6, 8x8, 10x10 and 12x12 respectively). Figure 4a and 4b display two of the four cluster dot dithered images used (created from C6 and C12 cluster dot patterns respectively). The variable dithered image is shown in Figure 4c. As can be seen, the regions of high busyness were created by extracting the corresponding pixels in the C6 dithered image, thus these regions reproduce the details in the original image. The regions of very low busyness were created by extracting the corresponding pixels in the C12 dithered image, thus obtaining good gray scale resolution in regions having no high frequency details. The appropriate dithered image was chosen by thresholding the busyness values into four levels corresponding to the four choices of dithered images.

However, the results in Figure 4c reflect the problem which arises using the pixel-by-pixel technique of variable dithering. False texture boundaries appear in several places in the halftoned image.

a.



b.



c.

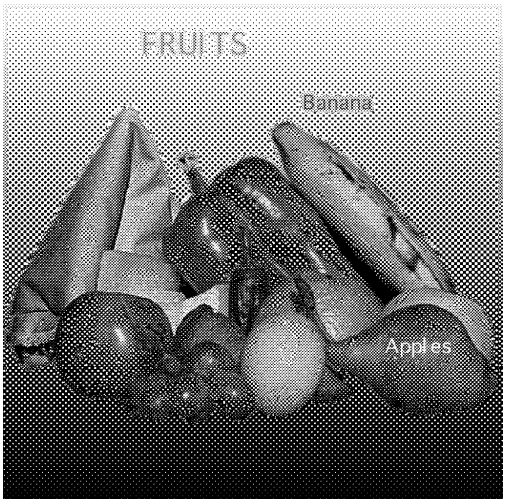


Figure 4: Variable Dithering using the pixel-by-pixel method.
a) Cluster Dot dithering using a C6 cell. b) Cluster Dot dithering using a C12 cell.
c) Variable Dithering using the pixel-by-pixel method.

These texture boundaries arise from the transition between busyness values below and above a threshold. Thus the two sides of the texture boundary comprise of pixels extracted from different dithered images. The textures in each dithered image are determined by the size of the cluster dot dither cell used. The greater the difference between dither cell sizes in the different dithered images, the more pronounced is the texture boundaries produced in the halftoned image. Figure 5 shows an enlarged example of such a texture boundary. The texture boundary emerges due to the change in pattern (small clusters to large clusters) and also due to the fact that the clusters along the border are truncated.

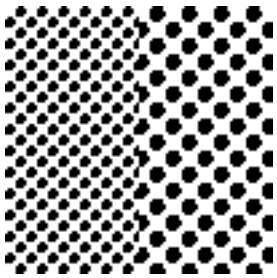


Figure 5: Texture Boundaries
Texture boundaries arise in the halftoned image due to the difference in textures between the dithered images from which pixels are extracted. In this example, the left pixels were extracted from the dithered image created with small Cluster Dot cells and the right pixels were extracted from the dithered image created with large Cluster Dot cells.

5 Variable Dithering: Using Flexible Boundaries - Springs

As seen in the results in Section 4.1, the Pixel-by-Pixel dithering method creates texture boundaries. This problem arises when the dither cells are not complete (i.e. not all threshold levels in a given cell are used) or when there is a significant difference between dither patterns due to large differences in cell sizes.

To overcome these problems, the array of thresholding levels must be created as a tessellation of complete dither cells of continuously varying sizes. This can be obtained by refraining from committing to any previously determined collection of cell sizes, rather allowing the tessellation to include a continuous range of sizes. In this section we develop a variable dithering method which uses such a tessellation. In order to obtain the desired tessellation we relax the implicit constraint used in most dithering techniques and allow the dither cells to lose their classical rectangular shape.

The variable dithering technique described in this section uses Flexible Boundaries to warp a rectangular grid into a non-isotropic grid of quadrilaterals of various sizes. The quadrilaterals are small in areas of high busyness and large in areas of low busyness values. These quadrilaterals are then used as cluster dot dither cells, by bilinearly warping a standard cluster dot cell of thresholds into each of the quadrilaterals. The design of this approach is shown in Figure 6. Details are given in the following subsection.

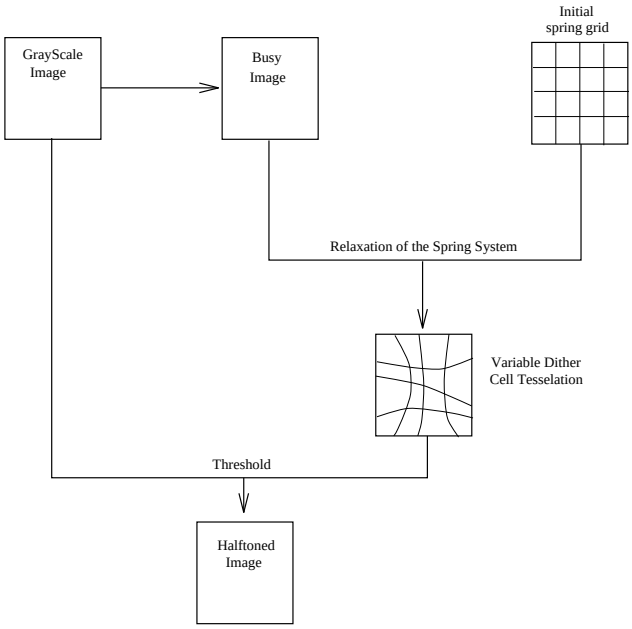


Figure 6: Outline of the Flexible Boundary (Spring) variable dithering method.

5.1 Implementation: "Super Springs"

To create such a tessellation we use a grid of springs whose spring constants are determined locally by the busyness values of the underlying original image. Relaxation of this system of springs causes contraction in areas of high busyness creating small grid cells at the expense of expansion in regions of low busyness values where large grid cells form. Due to the characteristics of the springs, cell sizes vary gradually.

The process is initialized by dividing the image plane into a grid of rectangular cells (Figure 7a). Each cell boundary is then replaced with a spring. The springs connect at vertices initially positioned at the corners of the rectangular grid (Figure 7b). Every spring is assigned a spring constant proportional to the busyness values in the underlying image. The systems of springs are then relaxed allowing the vertices to move freely about the image plane (the vertices positioned on the image boundary are constrained to move only along the image boundary). Due to the spring constants assigned to the springs, the spring system relaxes so that regions with high busyness values (having stronger contracting springs) will shrink, while regions of low busyness will expand (Figure 7c).

The relaxation procedure is computed by iteratively calculating new positions for each of the vertices as follows: The original location of a vertex (i, j) is denoted $X_{i,j}$. There are four springs converging at each vertex (except for the boundary vertices), each spring has an assigned spring constant denoted $U_{i,j}, D_{i,j}, L_{i,j}, R_{i,j}$ - up, down, left and right respectively (see Figure 8). When computing the new location of vertex (i, j) we assume the neighboring vertices are locked in their current position. The new location of vertex (i, j) is computed to minimize the energy of the system. It can be shown that the new location of vertex (i, j) is the average location of it's four neighbors,

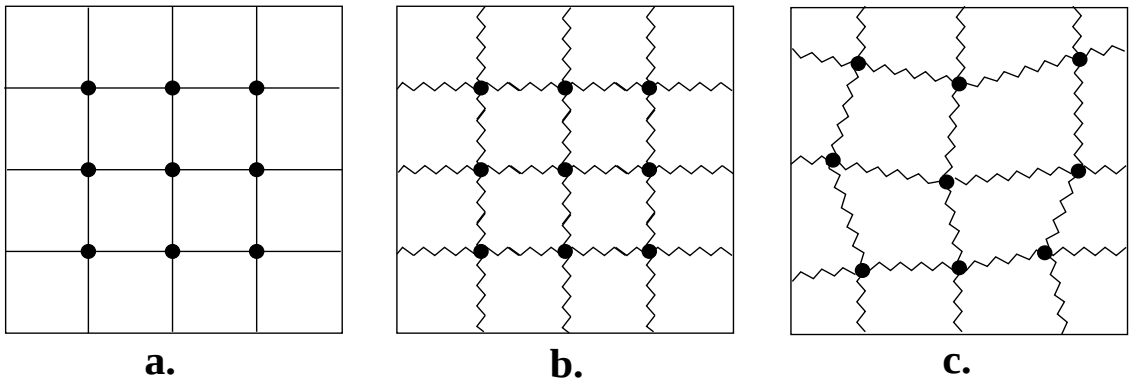


Figure 7: Creating a warped grid using springs.
a) Image plane is tessellated by a uniform grid of rectangular cells. b) Every cell boundary is replaced by a spring. c) After assigning appropriate spring constants, the spring system is relaxed to equilibrium.

weighted by the spring constants:

$$X_{i,j} = \frac{U_{i,j}X_{i,j-1} + D_{i,j}X_{i,j+1} + L_{i,j}X_{i-1,j} + R_{i,j}X_{i+1,j}}{U_{i,j} + D_{i,j} + L_{i,j} + R_{i,j}}$$

At each iteration we compute a new location for every vertex. The iterations continue until some terminating criteria of minimum motion is satisfied. After convergence of the iterative method, every vertex is positioned at a new location in the image plane.

It can be easily shown that the process always converges using this procedure. However, the convergence of the above iteration method is very slow. To accelerate convergence we use a multi-resolution (multi-grid) approach. We first use only a single vertex positioned centrally in the image plane as shown in Figure 9a. It's upper vertical spring will be the average of all the vertical springs in the upper half of the grid, it's right spring will be the average of all the horizontal springs of the right half of the grid, etc. Following the rapid convergence, additional vertices are inserted into the system (to double it's linear resolution). Locations of the new vertices are interpolated between existing vertices. The new system is brought to equilibrium, new vertices added and the process is repeated until all required vertices have been inserted (Figure 9b-c). This approach greatly accelerates the convergence of the entire process.

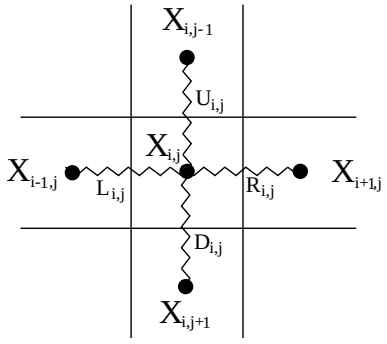


Figure 8: A system of springs for vertex (i,j).

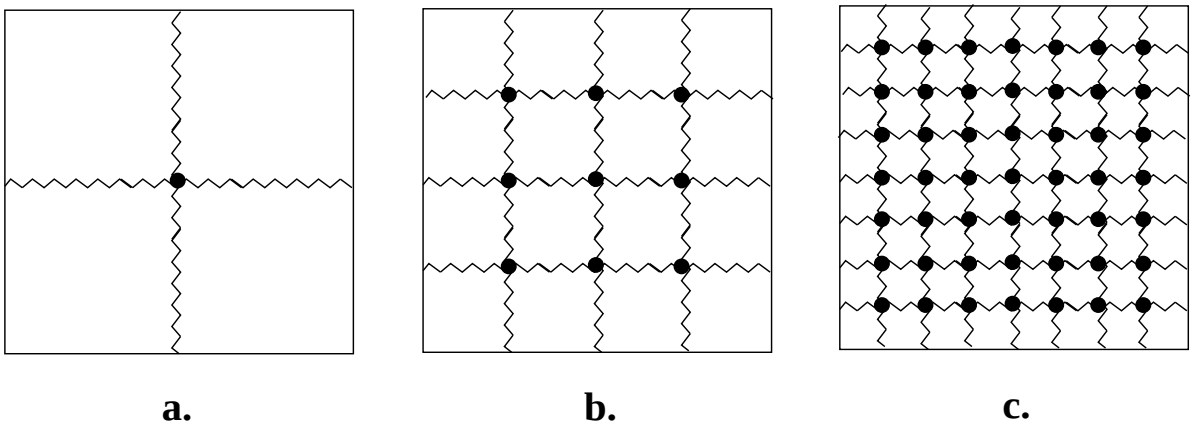


Figure 9: Several stages in the Multi Grid approach.

At the end of the convergence process of the spring system, we obtain a warped grid of quadrilaterals cells, each of which represents a cluster dot dither cell. The cells are small in areas of high busyness values in the original image and are large in areas of small busyness values. To obtain the actual threshold values of each of the cells we map, using a bilinear interpolation, a standard cluster dot cell into each of the quadrilateral cells (Figure 10).

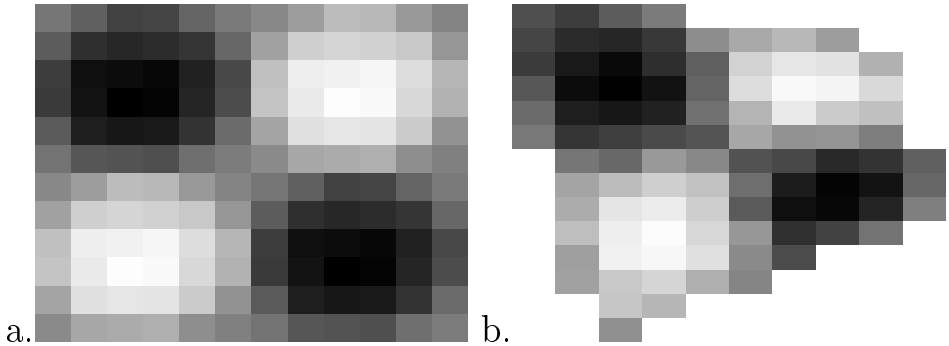


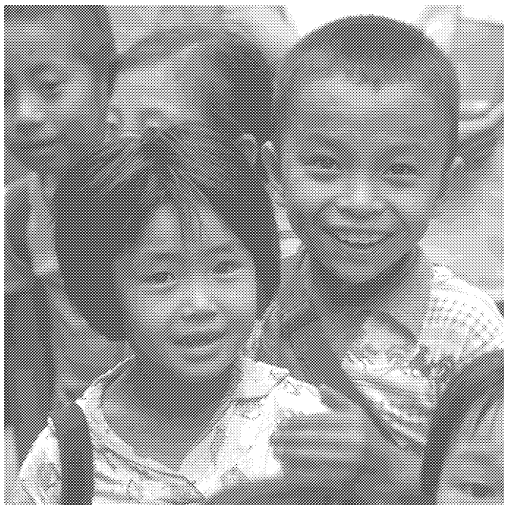
Figure 10: Using bilinear interpolation, the threshold values of a standard cluster dot cell (a) are mapped into a warped 4-sided cell (b).

5.2 Results

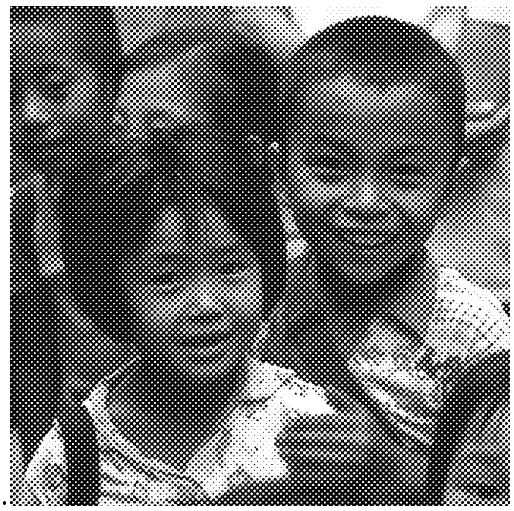
Figure 11 shows the result of applying the variable dithering using flexible boundaries as described in this section. The original image is 1024x1024 and contains regions of high busyness (patterned sweater) and regions of low busyness with slowly varying gray tones (background, little boy's forehead).

The spring system was initialized with a grid of rectangular cells of 8x8 pixels each. As can be seen from the halftone image that results (Figure 11c) the dither cells are small in high busyness regions and large in low busyness regions. For a comparison, we display halftoned images obtained using a grid of standard Cluster Dot dither cells. Figure 11a displays the halftoned image obtained using C6 cells. Figure 11b displays the halftoned image obtained using C12 cells.

a.



b.



c.

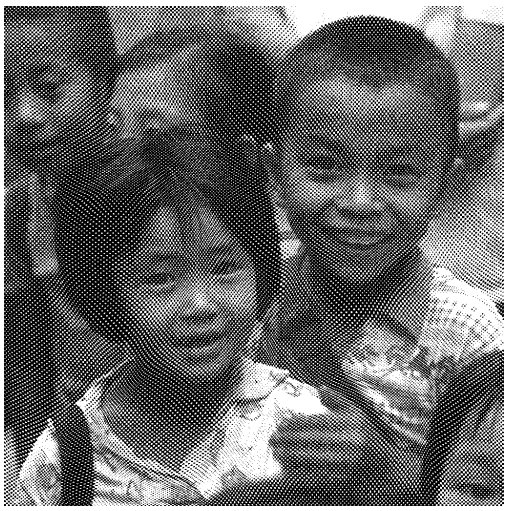


Figure 11: Variable Dithering using the flexible boundaries method.
a) Cluster Dot dithering using a C6 cell. b) Cluster Dot dithering using a C12 cell.
c) Variable Dithering using the deformable boundaries method.

We see that fine details are better preserved in the Variable dithered image compared to the C12 dithered image (note the patterns in the sweater for example). The false contouring appearing in the C6 dithered image (on forehead for example) which are due to the inability of the C6 cell to capture the slowly varying gray tones, are eliminated in the variable dithering technique.

A drawback of this method, however, is that aliasing patterns tend to appear in the variable dithered images. A more pronounced example is shown in Figure 15d. The aliasing patterns in the form of perceptual lines and bands arise from the spring process and the basic arrangement of grid cells in columns and rows. The constrained connectivity between neighboring cells and the characteristics of a spring system create correlated deformations between neighboring cells, thus a row of cells may be oriented and stretched similarly creating a visually perceived band (as stretch marks might arise when pulling at silk stockings).

Several improvements can be incorporated into the flexible boundary variable dithering: The aliasing patterns are prominent when the cells are elongated with a high width to length ratio. Reducing the aliasing can be obtained by avoiding elongated cells. This can be done by post-processing the cell tessellation, detecting elongated cells and dividing each into two or more non-elongated cells. Another possibility is to incorporate constraints into the spring system during relaxation, either by restricting vertex motion or by adding additional diagonal springs to each cell which will induce some rigidity in shape deformation of each cell.

6 Variable Dithering: Using Voronoi Tessellation

To overcome the aliasing problem arising when using the spring method described in the previous section, a different tessellation should be used, which uses cells having aspect ratios close to one and which does not align the cells in continuous strips (as occur in the form of rows and columns in the tessellation obtained from the spring system as described in the previous section).

A classic tessellation that fulfills these requirements is the *Voronoi Tessellation* [10]. A Voronoi Tessellation is a division of the image plane into sets. The division is based on a given set of *seeds* which are designated by their coordinates. Given a set of n seeds, the Voronoi tessellation produces a division of the image plane into n sets such that each set contains all points closest to one of the seeds. An example is shown in Figure 12. The seeds are marked as dots, the line segments denote the division of the image plane into sets.

The cells in the Voronoi tessellation usually do not align in rows and columns and the cells tend not to be elongated. Thus the Voronoi tessellation is an appropriate candidate for creating a tessellation of dither cells. In applying the Voronoi tessellation to variable dithering, we require the tessellation to have varying sized cells such that the cell sizes are small in image regions with high busyness value (fine details) and cells are large in regions of low busyness value (slowly varying gray tones). Since the Voronoi tessellation is determined by the seed distribution, it is the local density of the seeds which must vary with the busyness of the image. Seed density should be high in high busyness regions and density should be low in low busyness regions. In every region the seeds should be as evenly distributed as possible according to the local density.

To obtain variable density of voronoi seeds which are dependent on the busyness, an iterative

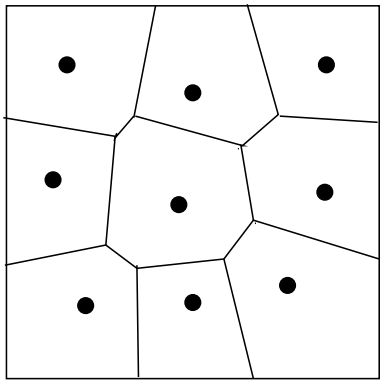


Figure 12: Voronoi Tessellation
Given a set of n seeds, the Voronoi tessellation produces a division of the image plane into n sets such that each set contains all points closest to one of the seeds.

random distribution or a weighted random distribution can be used. However the statistics of such distributions can not guarantee that seeds will be as evenly distributed as possible.

To obtain a density varying even distribution we propose the novel approach of using the *Void and Cluster* (VC) dither cell pattern [18]. This approach takes advantage of the properties inherent in the design of the *Void and Cluster* dither cell, namely, the threshold values in the cell, greater than any given gray tone value are always distributed as evenly as possible. When using the VC cell for dithering, this characteristic creates halftone patterns in which the dots are highly unclustered (which, as discussed in Section 1, is one of the disadvantages in using the VC for dithering).

Given, that thresholding with the VC pattern provides evenly distributed dots at any gray level, we can obtain evenly distributed dots at varying densities dependent on the busyness values, by thresholding the busyness image with a VC pattern. The dots obtained by this thresholding process are dense in regions of high busyness values and sparse in regions of low business values. Thus the output dots of such a thresholding can be used as Voronoi seeds for the Voronoi tessellation process which will produce a tessellation with variable sized cells.

Creating a Voronoi tessellation of a digitized image plane from a set of seeds can be performed in linear time using a two pass algorithm (see [12] for details). A by product of the two pass algorithm for computing the Voronoi tessellation, is a distance map which gives the distance of every pixel to the seed of it's assigned tessellation cell.

Given such a Voronoi tessellation, every tessellation cell is considered a warped dither cell. Threshold values must now be assigned to the pixels of each warped cell. The actual threshold values of a warped cell are determined by sorting the distance map entries for all pixels in that cell. The pattern of threshold values obtained for the warped cells are spirally increasing, similar to the cluster dot threshold pattern shown in Figure 2a.

The threshold values in the warped dither cells are used to threshold the original image. The design of the Variable Dithering technique using the Voronoi tessellation is given in Figure 13.

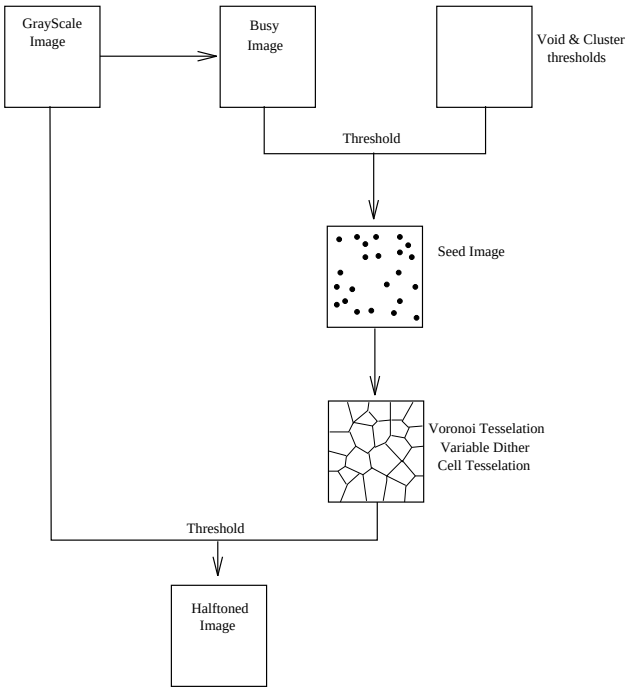


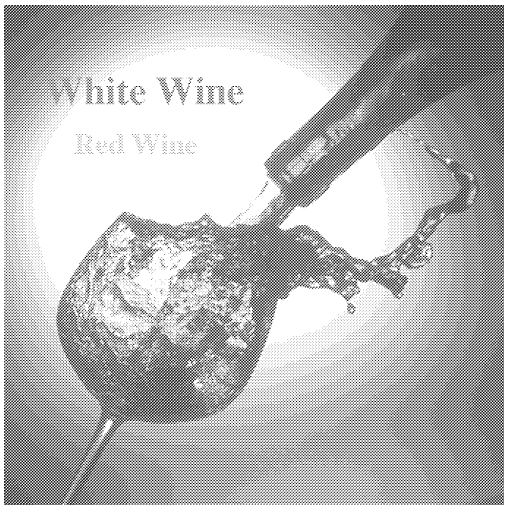
Figure 13: Outline of the Voronoi variable dithering method.

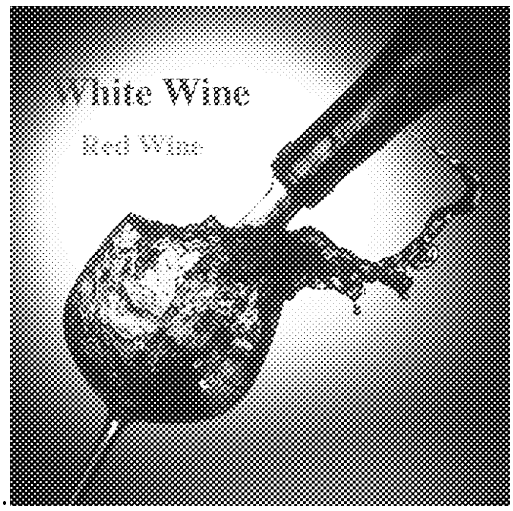
6.1 Results

Figure 14 shows the result of applying the variable dithering using the Voronoi tessellation as described in this section. The original image is 1024x1024 and contains regions of high busyness (text, ice in glass) and regions of low busyness with slowly varying gray tones (background). As can be seen from the halftone image that results (Figure 14c), details are reproduced in image regions of high busyness and gray tones are more accurately reproduced without the false contours in regions of low busyness values. The number of seeds used for the Voronoi tessellation was such that the average size of a dither cell was comparable to a C10 dither cell. Compare with the halftoned images obtained using a grid of standard Cluster Dot dither cells: Figure 14a displays the halftoned image obtained using C6 cells. Figure 14b displays the halftoned image obtained using C12 cells.

Additionally, the aliasing patterns appearing in the halftoned images created using the Flexible Boundaries technique described in Section 5, do not appear when using the Voronoi tessellation.

The halftoned image created with the Variable Dithering using Voronoi tessellation, consists of irregular patterns similar to those created using error diffusion (compare with the homogeneous pattern of clusters created using standard cluster dot dithering). It is questionable whether this pattern is less pleasing, most likely the answer is dependent on resolution. Future investigation is directed to try to produce a more pleasing pattern using the Voronoi method. Since the VC pattern is the determinant of the cell pattern and consequently the halftoned image pattern, producing a variant of the VC cell pattern which incorporates ordering at every threshold level, will be investigated.

a. 

b. 

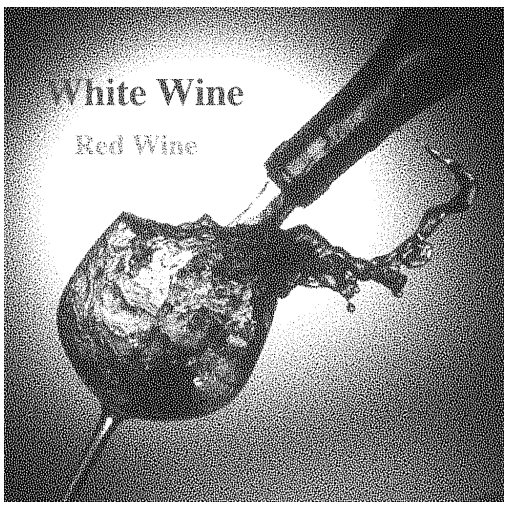
c. 

Figure 14: Variable Dithering using the Voronoi method.
a) Cluster Dot dithering using a C6 cell. b) Cluster Dot dithering using a C12 cell.
c) Variable Dithering using the Voronoi method.

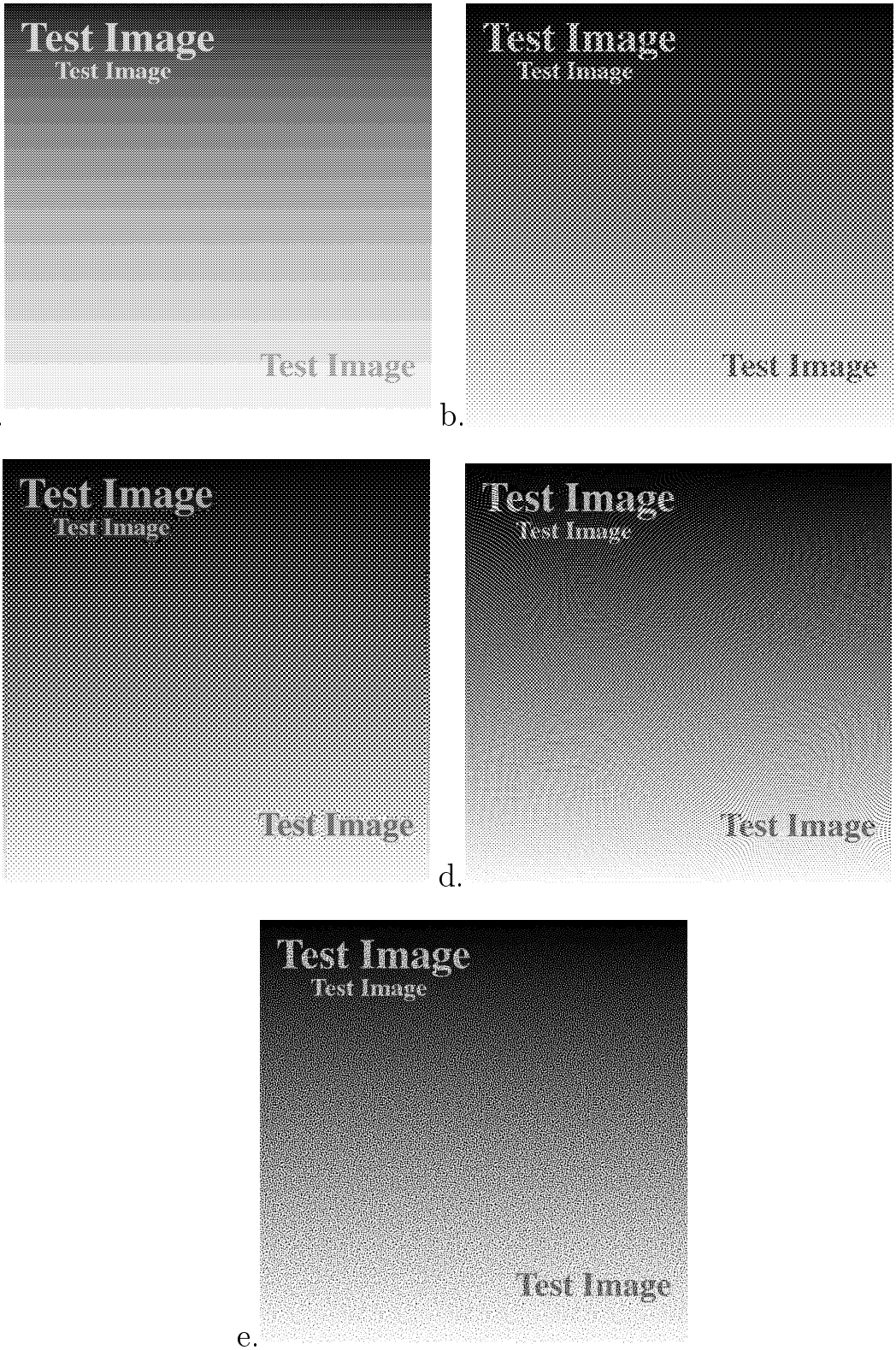


Figure 15: Comparison between a number of Variable Dithering methods. a) Cluster Dot dithering using C6 cell. b) Cluster Dot dithering using C12 cell. c) Variable dithering using the pixel-by-pixel method. d) Variable dithering using the deformable boundaries method. e) Variable dithering using the Voronoi method.

Figure 15 compares the halftoning results obtained using the different variable dithering methods presented in this paper. Note that in addition to the pattern differences, there is a difference in contrast between the halftoning results. This is due to the difference in callibration required for the various techniques. Calibration for correct tone production is dependent on cluster cell size. Thus the callibration must be incorporated into the variable dithering process by correcting the threshold values according to the cell size.

7 Discussion - Complexity and Memory Issues

In choosing the halftoning method to use, two practical issues must be considered:

- **Complexity** - Computationally expensive and time consuming methods are inadvisable. Although desktop output devices do not operate at offset rate, a reasonable rate of output printing is still required. Thus halftoning methods which are computationally efficient and are not time consuming are preferable.
- **Memory** - Standard output devices are limited in memory capacity, allowing only small images or portions of large images to be stored at any given time in the output device. This requires halftoning methods (performed at the output device) to compute the output image using local image operation based on small portions of the image data. The halftoning methods should avoid the necessity of storing the full original image in memory.

There are, however two caveats to be noted concerning complexity and memory:

- Most desktop printers offer the user a choice in the tradeoff between speed and quality. This allows slow and complex halftoning to be used when high quality output is required.
- Memory restrictions of the printer driver can be overcome by computing the halftone image in software at the computer and sending the raw halftoned image to the printer. In general, computations performed in software are more time consuming than when performed at the printer server. Additionally, this approach increases the dependence of the printer on the computer and it's available software.

In this section we discuss these practical issues in relation to the adaptive halftoning techniques presented in this paper.

A common factor in all the adaptive methods introduced in this paper, is the efficiency of computing the busyness value at every pixel of the original image. As described in Section 2, the busyness is computed by convolving the image with a laplacian kernel followed by an averaging kernel. This approach allows the busyness value at every pixel to be computed locally, using only neighboring pixels (a 20x20 neighborhood of pixels in our specific case). This is very efficient in terms of memory storage requirements since for every busyness value computed for a pixel only the immediate neighboring pixels need to be stored in memory. In terms of computational efficiency, the busyness value can be computed efficiently by incrementally shifting the local neighborhood window and updating the previous pixel's busyness value to obtain the next pixel's busyness value.

7.1 Pixel-By-Pixel Adaptive Halftoning Method

The pixel-by-pixel adaptive halftoning method described in Section 4 is efficient in both memory and computational complexity. The method is computationally inexpensive: given the busyness value at a pixel (which as described above can be computed using only local pixel neighborhoods), this method requires a decision step for selecting the appropriate cluster dot size and requires a single thresholding operation for each pixel.

This method is efficient in terms of the memory requirements: Since the thresholding step requires the storage of a single pixel, the memory requirements is limited only by the pixel storage required for calculating the busyness value.

7.2 The Flexible Boundary Adaptive Halftoning Method

The Flexible Boundary halftoning method, as described in Section 5, is both computationally and memory expensive. At every iteration a local average must be performed for each vertex in the spring grid. Using the multi-resolution approach convergence is greatly improved and accordingly the computational complexity. In our simulations we found that 2-3 iterations were required for convergence in the first few stages and 5-6 iterations at the subsequent stages. Run time and complexity are still extreme for reasonable printing speed. However, the algorithms used in the Flexible Boundary method are easily and inherently, parallelizable. Given a parallel architecture complexity and run-time become very reasonable for desktop printing.

In terms of memory requirements, notice that at every stage of the multi-resolution procedure, only a subset of the data is required, namely, the average busyness values of those regions represented by the springs of the grid. At later stages of the multi-resolution procedure when the grid is dense, almost all the busyness data is required for the computation and convergence of the spring system. However, at these final stages, the convergence is such that the changes in the spring system are mostly local, and the final converged state of the system can be closely approximated by computing convergence on local portions of the spring system. This implies that, again only a subset of the busyness data needs to be stored in memory at any given time. The problem is that the partial information that is required at every step is not systematically incrementing (as happens when pixels are scanned using a scan line procedure, or using moving windows of pixels). Continuous communication and data transfer is needed between the system containing the original image, and that which is performing the Flexible Boundary algorithm.

To conclude, the Flexible Boundary halftoning algorithm is not applicable in its current state, as an efficient halftoning scheme for output devices. Further work on efficiency of the algorithm is required.

7.3 Voronoi Adaptive Halftoning Method

The Voronoi method for adaptive halftoning, as described in Section 6 is moderately efficient in terms of computation and memory:

Computing the seeds for the Voronoi process is performed by thresholding the busyness image with a VC pattern. This requires a single scan of the busyness image and a single threshold operation per pixel. Calculating the Voronoi tessellation is performed (as described in Section 6) using a two pass algorithm. However a single pass is sufficient when using one of the variants of the sweep algorithm for calculating Voronoi tessellation (see [10]).

The bottle neck in terms of complexity for the Voronoi halftoning method, is the mapping of the standard cluster dot cell to the warped cell obtained by the Voroni tessellation. This requires sorting the distance values for every warped cell. An improvement may be achieved by using some form of lookup table to associate pixels of a warped cell with the appropriate threshold value. This is left for future study.

Thus, the Voronoi halftoning method can be performed with a single window-scan over the original image. At every scan step, the busyness of the pixels in the window must be calculated and passed through the VC threshold pattern. The Voronoi tessellation algorithm is updated to include the pixels in the window and create new tessellation cells. The newly created tessellation cells are assigned threshold values and used to threshold the original image. The halftoned window can be outputed and all associated data can be discarded from memory.

Further studies should further improve the efficiency of the Voronoi halftoning algorithm.

8 Conclusion

In this paper, we introduced three *Adaptive Dithering* techniques based on Cluster Dot dithering. These adaptive dithering methods allow variability in the dither cell size within a single image. Thus, small cell sizes are used in image regions having fine details and large cell sizes are used in image regions where gray tones are to be accurately reproduced.

The Pixel-by-Pixel method introduced in Section 4 is computationally efficient however produces visual artifacts such as artificial texture boundaries. The Flexible Boundary method introduced in Section 5 is computationally complex and requires relatively large memory capacity. This method eliminates the texture boundaries however it produces other visual artifacts, namely aliasing. The Voronoi method overcomes both the texture boundary problem and the aliasing problem. It produces a halftoned image containing a texture pattern more similar to that obtained using error diffusion techniques. It is also reasonably computationally and memory efficient.

9 Acknowledgment

The authors thank Prof. Jan P. Allebach for fruitful discussion and comments.

References

[1] M. Analoui and J.P. Allebach. Model based halftoning using direct binary search. In *SPIE/IS&T Symposium on Electronic Imaging Science and Technology*, volume 1666, pages 96–108, San Jose, CA, 1992.

[2] B.E. Bayer. An optimum method for two-level rendition of continuous-tone pictures. In *IEEE International conference on Communication*, volume 1, pages 26–11–26–15, 1973.

[3] M.F. Carlson and P.W. Besslich. Adaptive selection of matrix size for pseudogrey rendition of images. *Optical Engineering*, 24(4):655–662, 1985.

[4] E.R. Dougherty (Ed.). *Digital Image Processing Methods*. M. Dekker, New York, 1994.

[5] R. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. In *Proceedings of the Society for Information Display - SID*, volume 17, pages 75–77, 1976.

[6] K.T. Knox and R. Eschbach. Threshold modulation in error diffusion. *Journal of Electronic Imaging*, 2(3):185–192, 1993.

[7] R. Levien. Output dependent feedback in error diffusion halftoning. In *Proceedings of IS&T 46th Annual Conference*, pages 115–118, Cambridge, MA, 1993.

[8] J.B. Mulligan and A.J. Ahumada. Principled halftoning based on human visual models. In *Proceedings of the SPIE: Human Vision, Visual Processing, and Digital Display III*, volume 1666, pages 109–120, San Jose, CA, 1992.

[9] T.N. Pappas and D.L. Neuhoff. Least-squares model-based halftoning. In *Proceedings of the SPIE: Human Vision, Visual Processing and Digital Display III*, volume 1666, pages 165–176, San Jose, CA, 1992.

[10] F.P. Preparata and M.I. Shamos. *Computational geometry*. Springer-Verlag, New York, 1985.

[11] P.G. Roetling. Visual performance and image coding. *Proc. SID*, 17(2):111–114, 1976.

[12] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *ACM Journal*, 13:471–494, 1966.

[13] K.R. Sloan and D. Campbell. Texture as information. In *Proceedings of the SPIE*, volume 1913, pages 344–354, San Jose, 1993.

[14] S.Thurnhofer and S.K. Mitra. Nonlinear detail enhancement of error-diffused images. In *Proceedings of the SPIE*, volume 2179, pages 170–181, San Jose, 1994.

[15] J.C. Stoffel and J.F. Moreland. A survey of electronic techniques for pictorial reproduction. *IEEE Transactions on Communication*, COM-29(12):1898–1925, 1981.

[16] H. Szu, Y. Zhang, M.Sun, and C. Li. Neural network adaptive digital image screen halftoning (DISH) based on wavelet transform preprocessing. In *Proceedings of the SPIE*, volume 2242, pages 963–966, 1994.

[17] H. Szu, Y. Zhang, M. Sun, and L. Ching-Chung. Neural network adaptive digital image screen halftoning (DISH) based on wavelet transform preprocessing. In *Proceedings of the SPIE: Wavelet Applications*, volume 2242, pages 963–966, Orlando, FL, 1994.

[18] R. Ulichney. The void-and-cluster method for dither array generation. In *Proc. SPIE: Human Vision, Visual Processing and Digital Display IV*, volume 1913, pages 332–343, San Jose, CA, 1993.

[19] R.A. Ulichney. *Digital Halftoning*. Mit press, Cambridge, Mass, 1987.

[20] L. Velho and J. Gomes. Stochastic screening dithering with adaptive clustering. In *SIGGRAPH-Computer Graphics Proceedings*, pages 273–276, Los Angeles, CA, 1995.

[21] T. Zeggel and O. Bryngdahl. Adaptive halftoning based on iterative convolutions. In *Proceedings of the SPIE: Human Vision and Electronic Imaging*, volume 2657, pages 456–463, San Jose, CA, 1996.

[22] Y. Zhang. Adaptive ordered dither. *Graphical Models and Image Processing*, 59(1):49–53, 1997.