

Using errors in halftoning to increase reproduction accuracy

Stephen Herron

EI2006 Conference, January 19, 2006

ABSTRACT

Halftones, or periodic cluster dithers, are point-to-point, contone pixel to halftone dot, conversion processes. This paper describes an approach to achieving accurate halftone reproduction by using the errors caused during the scan conversion process without comparison to the source pixels and maintaining the point-to-point scan conversion paradigm.

Two types of density errors encounter in the halftoning process are considered. 1. An error describing the difference between the input pixel value and the halftone threshold value. 2. An error describing the difference between the density level of the output data and the reflection density of the ink or toner on the page.

Keywords: Threshold array, amplitude modulation, frequency modulation, cluster halftone, transfer function, pulse width modulation, dispersed array, blue noise

INTRODUCTION

Current error handling methods include:

- Diffusion dither or spatial dispersion of an error within a neighborhood of pixels. While each individual dot exhibits a density error, the neighborhood of dots eliminate the errors.
- The relationship between input pixels values to the output dot density values. If the current pixel value results in a different reflection density, then adjust the pixel value proportionally.
- Assign density values to cluster cells. Each cluster in a bit-mask lookup table is assigned a key value representative of the density value of the Pulse Width Modulation engine, (PWM).

In this paper, a device dot is the dot placed on the substrate by the print engine at the resolution of the printer. Many printers have resolutions of 600 dpi and 1200 dpi. The halftone dot is assumed to be a cluster of device dots. Multi-bit or multi-size device dots are not considered in this current research. The point-to-point process is a process where each pixel is associated with one PWM device dot. After the association is made, then the pixel is discarded.

Information about the accuracy of the halftone image is contained within the errors. Halftone errors are metadata derived from a contone characteristic and communicated to the halftone conversion process. This metadata improves reproduction accuracy. Each error is associated with a device dot maintaining the point-to-point relationship.

The contone to halftone conversion is from a pixels's amplitude to a dot's amplitude times frequency. A pixel representing 50% tone may be represented by turning on 128 device dots within a 256 dot cluster. Each dot has an amplitude of 1-bit, and the maximum frequency is 256 dots. Frequency is a method of preserving the pixel's amplitude information. A further use of frequency is to disperse the error among a neighborhood of halftone clusters.

DESCRIPTION OF THE INPUT-PIXEL-TO-OUTPUT-THRESHOLD VALUE ERROR

Many printers encode halftones in 8-bit or 255 shades of a colorant plus white. An 8-bit pixel to a halftone with 255 shades of colorant is scan converted according to this function:

If the pixel, P , is 0 then write 0. If halftone $H \leq P$ and P is > 0 then write H . (1)

This formula looks simple without the possibility of a threshold error. Unfortunately, this simple one-to-one relationship does result in an error. A typical error results from the conversion of the 256 pixel levels, ranging from maximum

lightness to no light, to a matrix of device dots. Since there are 256 device dots in a 16-by-16 halftone dot cluster, and white represents no dot then there are 257 possible levels from maximum reflection of light to no light. The conversion of 256 intensity levels to 257 reflection levels results in rounding errors. The standard threshold halftone solution is to turn on two device dots at the first tonal level. In this case, all tonal levels are one level too dense.

Page description languages can encode many more than 255 shades plus white. A PostScript vector fill or stroke operator can specify shades expressed as a decimal between 0.0 and 1.0 up to four decimal places. The PostScript paradigm uses 256 levels and each level is spaced 0.0039 units apart. Each level is:

$$0.0039 \times N \quad (1 > N \geq 255) \text{ and } 0.0001 \text{ to } 0.0039 = 1. \tag{2}$$

The fence-sitting metaphor illustrates the problem. If an array of numbers is divided into sets is each set delineated by the highest number in the set or the penultimate number? (See figure 1) If 1.0 is divided into 256 levels, or 0.0039 x 256, then three possibilities exist:

1. if $0.0 < P \leq 0.0039$ then first dot
2. if $0.0 \leq P < 0.0039$ then no dot.
3. If $0.0 < P < 0.0039$ then first dot

0.0039 sits on the fence between no dot and one dot. If 0.0 = no dot and 0.0001 to 0.0039 equals the first dot then, 1.0000 equals the 256th dot. If the range 0.0001 to 0.0039 is for no dots, then 1.0000 is the 255 dot. The 50% tonality is also in question. If there are 255 tonal levels not including white then the 127 tone is the 50% tone. Since there are 256 device dots in a cluster then the 128 tone is also the 50% tone.

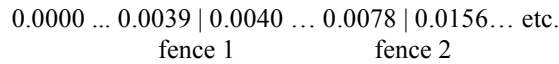


Figure 1

A PDF raster can use 16-bit encoding describing over 65,536 shades of colorant. The fence sitting metaphor applies and quantification errors occur when the output device is an 8-bit printer.

The 16-bit encoding is divided into 256 sets. Each set is separated from the next highest set by the largest number in the set. The fence is the largest number in each set. During the contone to halftone quantification process all numbers within the set are mapped to the fence. (There are various other logical decisions available to quantify many numbers to a single number, but for this example $\lfloor N \dots N' \rfloor = N'$.)

DESCRIPTION OF THE REFLECTION-DENSITY ERROR

The second error is the density gain typically described by the Murray-Davis and Yule-Nielsen functions.

A transfer curve is generated to chart the difference between the numerical value and the actual density value of each cluster size over a range of values. A transfer curve specifies the error between the mathematical density and the printer's physical density. The error in reflection density is the result of an average of min/max of a statistical sampling of three functions – 1. PWM engine density, 2. Cluster dot perimeter gain and 3. halftone pattern density. The typical method of determining density error is with the use of a densitometer. An array of the values is made by associating every pixel's intensity value with the density value that results in the halftone reproduction of that pixel value.

THE ERROR ADJUSTING PROCESS

The error is the sum of the perimeter gain and the threshold errors. The two errors -- the threshold error and the density gain error -- are combined and stored in an alpha plane. The threshold values are then adjusted in an iterative process. An adjustment cycle through the halftone image eliminates the errors in the alpha plane.

$$\sum_1^{256} [a + b] \text{ where } a \text{ is the threshold error and } b \text{ is the perimeter gain or output density error} \quad (3)$$

The algorithm is a two-pass process. During the first pass through the image, each pixel value in the image is converted to the threshold value according to the standard scan conversion method. (See function 1) The difference between the threshold value and the vector or pixel value is accumulated. That value is added to the average density increase. The values are recorded in an alpha plane. The alpha plane maps the halftone errors.

The second pass compares the value in the alpha plane with the threshold value. The highest cluster value is compared to the error value in the alpha plane. An adjustment to the number of device dots is made to the halftone threshold value when the value in the alpha plane indicates a different threshold value.

The algorithm:

1. Scan convert pixels, (P) to halftone dots, (H):
If $P \leq H$ then H
2. Count the number of dots turned on and divide by the number of cells in the cluster
Let X equal the threshold error, C the total number of cells with values > 0 , and T the total number of cells
 $X = C/T$
3. Let Y equal the output density for the current halftone cluster
4. The error, E, is the accumulation of threshold errors, X, and the output density Y
 $E = \sum_0^1 X + Y$
5. Compare the error value to the threshold value and scan convert
6. Write the binary output for the PWM engine
If 0 , then write 0 for white, If > 0 , then write 1 for black

A hypothetical error-handling example using a 16-unit cluster halftone dot follows:

1. The pixel values of a 50% tone when 16 is black:

8	8	8	8
8	8	8	8
8	8	8	8
8	8	8	8

2. The threshold values of the cluster dot:

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

3. The resulting halftone dot after scan conversion:

0	0	0	0
0	2	1	8
0	3	1	7
0	4	5	6

4. The values from the transfer curve indicates a gain of half a tonal level:

8.5	8.5	8.5	8.5
8.5	8.5	8.5	8.5
8.5	8.5	8.5	8.5
8.5	8.5	8.5	8.5

5. Since 9 device dots were turned on, the threshold error equals $-1/16$:

-0.06	-0.06	-0.06	-0.06
-0.06	-0.06	-0.06	-0.06
-0.06	-0.06	-0.06	-0.06
-0.06	-0.06	-0.06	-0.06

6. The sum of the threshold error and the density error:

8.44	8.44	8.44	8.44
8.44	8.44	8.44	8.44
8.44	8.44	8.44	8.44
8.44	8.44	8.44	8.44

7. A device dot cannot change until the error value matches the threshold value. Accumulate the error:

8.38	8.38	8.38	8.38
8.38	8.38	8.38	8.38
8.38	8.38	8.38	8.38
8.38	8.38	8.38	8.38

8. At the eighth set of pixels the error equals 8 and the eight cluster becomes:

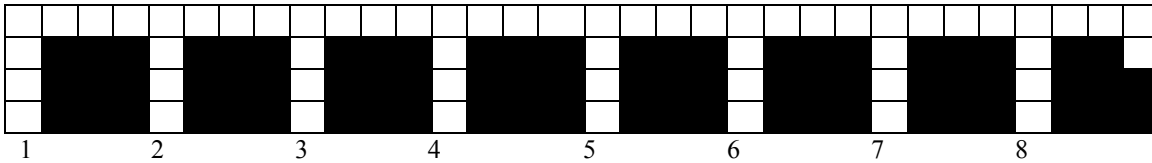
0	0	0	0
0	2	1	0

0	3	1	7
---	---	---	---

0	4	5	6
---	---	---	---

The number of dots and their positions within an area are changed as the comparison to the error map is made.

The resulting halftone cluster dot pattern:



This paper follows the subtractive color system whereby 0 is white and 1, 16, or 256 is black.

FURTHER RESEARCH

The low frequency component (the eighth cluster cell in the example) produce noise. The dispersion of errors within a population of halftone clusters may result in a repetition of a different dot shape from the shape of surrounding dots.

This repetition of a unique dot shape disturbs uniformity of a tone and adds noise to the human visual system's reception of that tone. The pattern of clusters within a neighborhood are arranged to minimize noise. Strategies include angle of inclination of the low frequency dots and placing those dots at higher frequencies than the mean frequency.

However, dot patterns require blue and green noise pattern considerations. The errors described in this paper do not carry information concerning the amount or type of noise contained in the halftone pattern. The modification of dot clusters to eliminate errors results in a noisy pattern.

This method or improving reproduction accuracy can be applied to multi-bit halftones. The errors encountered in the multi-bit conversion increase in inverse proportionally to the number of halftone bit levels – 1-bit halftone has the greatest errors and an 8-bit halftone has the least errors.

CONCLUSION

The result is a amore accurate conversion of pixel intensity values to halftone density values. With point-to-point cluster halftones, the halftone pattern contains clusters of various sizes arranged according to the error map.

REFERENCES:

Robert Ulichney, *A Review of Halftoning Techniques*, Color Imaging, etc.; Proceedings of SPIE, Color Imaging Conference V, Jan, 2000; pgs. 378—391

Qing Yu, et al, *A New Metric for Color Halftone Visibility*, IS&T 1998 PICS Conference, May 1998; pgs. 226--230

Shigeru Kitakubl, et al, *Some Characteristics on Human Visual Sensitivity for Spatial Frequency of Digital Halftone Images*, IS&T NIP 2005 Conference, September 2005 pgs. 118-- 121

Charles Hains, *The 30-year evolution of digital halftoning from the viewpoint of a participant*, Color Imaging IX: Processing, Hardcopy, and Applications. Proceedings of the SPIE, Volume 5293, pp. 434-447 (2003).

Keith Knox, *Evolution of Error Diffusion*, Journal of electronic Imaging, Vol. 8 N. 4, October 1999, pgs. 422-429

Daniel L. Lau, Gonzalo R. Arce, *Modern Digital Halftoning*, Marcel Dekker, Inc, 2001