Practical Issues in Color Inkjet Halftoning

Raph Levien

artofcode LLC, 940 Tyler St. Studio 6, Benicia CA 94510

ABSTRACT

Even Toned Screening has evolved from a research project into a practical module for halftoning on color inkjet printers, with many commercial and free software users worldwide. Feedback from these users has motivated tuning and other modifications to make the algorithm more practical. This paper discusses both the core algorithms and the practical issues involved in driving real printers for real users. The specific issues include: Nonsquare aspect ratios, Interaction between dither microstructure and weaving patterns, multilevel dot generation, processing speed, and interactions between microstructures in overlapping planes.

Keywords: Error diffusion, output-dependent feedback, inkjet, halftoning

1. INTRODUCTION

The core algorithm is very similar to that presented in [1]. A standard Floyd-Steinberg error diffusion kernel produces microstructures of very high quality on modern printers, more than a quarter century after the original publication. However, Floyd-Steinberg is well known for its generation of highly anisotropic "wormy" patterns of dots in highlights and shadows. An extremely effective method for making these highlight dots more homogenous is output dependent feedback. For each pixel, the distance to the nearest previously-generated dot is computed, and used as a threshold modulation in the error diffusion kernel. The result is a highly uniform, isotropic distribution of highlight dots.

Computing these distances efficiently remains an area of interest. The original paper referred vaguely to a "forward difference" method, but gave no specifics. Direct search can be quite slow, and quite dependent on the image contents. Subsequent literature has presented alternate, more efficient techniques. We present the original forward difference technique. Time complexity is O(1) per pixel, and the space utilization is three integers per pixel in a scanline buffer. We also show the adaptation of the distance calculation to nonsquare aspect ratios.

Modern inkjet printers are capable of multiple dot levels, resulting in considerably smoother reproduction of tones than bilevel devices. Ordinarily, error diffusion algorithms can simply be run with a multilevel quantizer rather than a step function. However, output dependent feedback requires care in the combination of the error term and the feedback term, otherwise darker dots may appear in regions intended for the lightest dot only. We present a simple combination technique, and also discuss its impact on tones near a constant dot size.

Many inkjet printers use multiple passes to achieve full resolution. To mitigate banding, these passes are finely interwoven. The resulting "weaving" patterns are of fairly high spatial frequency, and are thus not very visible. However, when the halftone contains repeating patterns of similar spatial frequency, objectionable patterns arise. We have found that addition of a small amount of randomness improves overall rendering considerably. We characterize typical weaving patterns and discuss their interactions with dither microstructure.

Dither patterns between the planes can interact, which has motivated a significant amount of research on color halftoning. We have found these interactions to be particularly significant in highlight regions, largely because of the highly uniform placement of dots. We have implemented a relatively simple technique to couple the dither patterns of multiple planes, in effect using lowpass filtered results from the darker planes as threshold modulation for the lighter planes. Quality improves as a result of three mechanisms. First, dots no longer overlap in highlight regions, resulting in tones as uniform as in the single-plane case. Second, in midtones, overlapping dots are minimized (and, because of conservation of ink, regions of no dot), reducing the overall luminance variation of the dither patterns. Finally, eliminating overlapping dots increases color saturation somewhat.

Further author information:
E-mail: raph@artofcode.com, Telephone: 1 707 746 6398, Web: http://www.artofcode.com/

Many of the techniques presented are fairly straightforward. However, error diffusion halftones are known for fragility—even a tiny change can have a profound impact on the quality of the final result. The techniques presented here have been selected to work well in concert to produce practical, high-quality output. Full source is available under the terms of the GNU General Public License, encouraging further experimentation and use.

2. DISTANCE CALCULATION

The core of the algorithm is an output-dependent feedback term based on the distance to the nearest previously generated dot. A larger distance biases the error diffusion quantization towards generating a dot, while a smaller distance biases away. This bias makes the distribution of inter-dot distances much tighter than traditional error diffusion algorithms.

The original paper[1] mentioned a "forward difference" approach, but did not specify it in detail. Marcu[2] described a search technique designed to be more efficient than brute-force, but still with a time complexity greater than $O(1)$ in the inter-dot distance. Eschbach[3] proposed a slightly different technique based on "imprints," also designed for faster execution than brute force search.

Here, we present a refined version of the original forward-difference approach. In general, it runs in two passes: a forward pass, typically running in parallel with the actual dot generation; and a backward pass, to ensure that distances propagate down and left as well as down and right. This propagation prevents artifacts analogous to the well known "knights move" artifact in Floyd-Steinberg halftoning. The algorithm of the original paper operated on a serpentine (boustrophedonic) scan, but we find that standard raster scan produces better microstructures in midtones.

Let us call the output image $t_{x,y}$, each value either 0 or 1. For a given (x, y), we wish to determine the distance to the nearest 1-valued $t_{x',y'}$ preceding (x, y) in the raster scan. Formally:

$$d_{x,y} = \min_{x',y'}(x - x')^2 + (y - y')^2, \quad \text{where: } t_{x',y'} = 1$$
$$\text{and } y' < y \vee (y' = y \wedge x' < x) \tag{1}$$

Note that, in this formulation, d corresponds to the square of the Euclidean distance to the nearest dot. To compute this efficiently, we make use of the recurrence relation $(x + 1)^2 = x^2 + (2x + 1)$. We store two additional line buffers $a_{x,y}$ and $b_{x,y}$ corresponding to $2x + 1$ and $2y + 1$, respectively.

In the forward pass, we compute:

$$
\left.
\begin{aligned}
r_{x,y} &= r'_{x-1,y} + a_{x-1,y} \\
a_{x,y} &= a'_{x-1,y} + 2 \\
b_{x,y} &= b'_{x-1,y}
\end{aligned}
\right\} \quad \text{if } r_{x-1,y} + a_{x-1,y} < r''_{x,y-1}
$$
$$
\left.
\begin{aligned}
r_{x,y} &= r''_{x,y-1} \\
a_{x,y} &= a''_{x,y-1} \\
b_{x,y} &= b''_{x,y-1}
\end{aligned}
\right\} \quad \text{otherwise}
\tag{2}
$$

The first case represents left-to-right propagation, and the second downward propagation. The resulting $r_{x,y}$ value is used in the threshold modulation term of the error diffusion quantizer, resulting in a pixel $t_{x,y}$. This pixel affects the propagation of distances to the right:

$$
\left.
\begin{aligned}
r'_{x,y} &= 0 \\
a'_{x,y} &= 1 \\
b'_{x,y} &= 1
\end{aligned}
\right\} \quad \text{if } t_{x,y} = 1
$$
$$
\left.
\begin{aligned}
r'_{x,y} &= r_{x,y} \\
a'_{x,y} &= a_{x,y} \\
b'_{x,y} &= b_{x,y}
\end{aligned}
\right\} \quad \text{otherwise}
\tag{3}
$$

After a left-to-right scanline as above, we compute the backward pass:

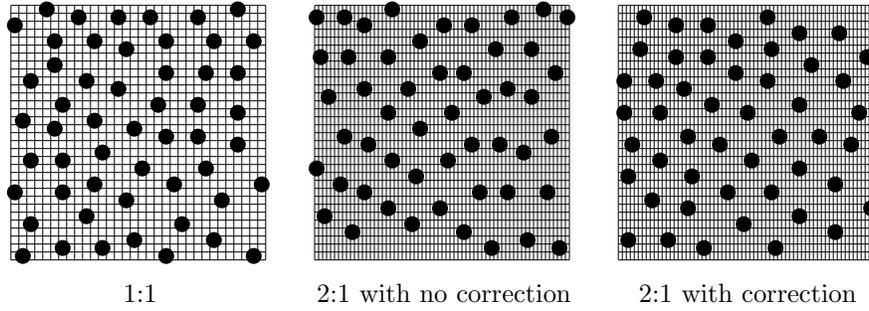|		
1:1	2:1 with no correction	2:1 with correction

Figure 1. Effect of aspect ratio correction.

$$\left.\begin{aligned}
z_{x,y} &= z_{x+1,y} + a''_{x+1,y} \\
a''_{x,y} &= a''_{x+1,y} + 2 \\
b''_{x,y} &= b''_{x+1,y}
\end{aligned}\right\} \quad \text{if } z_{x+1,y} + a''_{x+1,y} + b''_{x+1,y} - 2 < r''_{x,y-1} + b''_{x,y-1}$$

$$\left.\begin{aligned}
z_{x,y} &= r'_{x,y} \\
a''_{x,y} &= a'_{x,y} \\
b''_{x,y} &= b'_{x,y} + 2
\end{aligned}\right\} \quad \text{otherwise} \tag{4}$$

$$r''_{x,y} = z_{x,y} + b''_{x,y} - 2$$

Here, $z_{x,y}$ is a temporary. The first case corresponds to down-and-left propagation, while the second corresponds to downward propagation. For values outside the scan area, default values of 0 for $r_{x,y}$ and 1 for $a_{x,y}$ and $b_{x,y}$ may be used.

Thus, we effectively compute the nearest dot distance in constant time and modest memory.

2.1. Applying the distance term

As mentioned above, we use the distance term as a threshold modulation for the error diffusion quantization. This section describes the details.

It is important for threshold modulation terms not to contain DC bias dependent on the input gray shade, or transients will be blurred or sharpened, with a proportionately greater effect in extreme highlights. Thus, we estimate the average inter-dot distance, and use the difference between the actual and estimated distance as the threshold modulation term.

Empirically, we have determined that the expected value for d_{avg} for a grayshade g is:

$$d_{avg} = \frac{0.95}{g^2} \tag{5}$$

In the case where $g = 0$, the output pixel is always white, and the threshold modulation, including equation (5) is skipped.

3. ASPECT RATIO

Many modern inkjet devices have modes with non-square pixel aspect ratios. Halftones with uniform inter-dot distance measured on a square pixel grid display considerable anisotropy when rendered on non-square pixels.

Our approach extends straightforwardly: compute the distance $x^2 + cy^2$, where c is not necessarily 1. For example, with a 2:1 aspect ratio (1440×720 dpi), $c = 4$. We also update the scale coefficient for d in equation (5). We empirically determined a value of 1.8 for 2:1 ratios, and 3.6 for 4:1.

The effect of this correction for aspect ratio is shown in Figure 3.

We have experimented with other error diffusion kernels for non-square pixel aspect ratios, but have not found any that produce consistently better results than Floyd and Steinberg's original 5, 7, 3, 1 kernel.

4. RANDOMNESS

Adding random noise as a threshold modulation is a common trick to mask defects in the underlying error diffusion algorithm, and especially the interaction of the resulting microstructures with the device characteristics. Even Toned Screening is no exception.

Many modern inkjet printers produce their output in multiple passes, in order to accommodate a wide range of resolutions and quality/speed tradeoffs. Typically, the inkjet nozzles are spaced somewhat widely, for example at 1/180 inch, and the dot firing rate is also smaller than the maximum resolution of the divice, typically 1/360 inch. To produce output at, say, 720 dpi, eight passes are used. As a result, the assignment of a pixel (x, y) to a nozzle is typically a pattern with a repeat rectangle 2 pixels wide by 4 pixels tall.

In an ideal device, the assignment of pixel to nozzle would be irrelevant, because all nozzles would behave the same. In practise, it is not so. Real individual nozzles tend to have systematic biases in ink volume, as well as positional error. In addition, in higher speed modes it is common to run alternate passes in left-to-right and right-to-left directions, usually introducing further systematic positional errors.

As such, real devices are sensitive to periodic or quasiperiodic patterns appearing in the resulting halftone. A typical example is the "checkerboard" pattern common in error diffusion halftones for a dot density of 1/2. This pattern is usually worst-case for real devices, as it commonly results in some passes with 100% coverage and others with 0%. For tones near 50%, the result contains medium-scale domains of alternating phase, typically producing a strong mottled appearance.

Similar quasiperiodic patterns are common for other dot densities near small-denominator rational fractions, including 1/4, 1/3, etc.

Our approach to these patterns is to add a random threshold modulation, the amplitude of which depends on the input gray level. The amplitude is very small for most gray values, but larger in the area of rational fractions. Please refer to the source code for the exact values.

There are other approaches to these periodic patterns, some of which might be preferable. Fawcett and Schrack[5] used output-dependent feedback to dissipate them. A topic of future research is to add additional correction factors dependent on the actual weaving pattern of the target device.

5. MULTIPLANE OPTIMIZATION

Most inkjet printing today uses multiple planes of colorants. It's possible to process each plane separately with a single-plane halftone algorithm, but optimizing multiple planes at once can improve quality. Our approach is straightforward: we compute all planes in parallel, and use a simple "coupling" signal as a threshold modulation.

Our goal in multiplane optimization is simple: we wish to minimize the area of overlapping dots between planes. By conservation of ink, this goal also minimizes the white area not covered by dots. The result is a much more uniform appearance.

We present a very simple scalar coupling algorithm first, then extend it to a local-area approach.

We order the colorants in order from darkest to lightest, and, for each location in the raster scan order, process a pixel for each plane in succession. With subscripts representing planes, the coupling threshold modulation term is:

$$k_i = \sum_{j < i} c_j \cdot (g_j - t_j) \tag{6}$$

Here, $g_j - t_j$ represents the raw, or unfiltered, error resulting from error diffusion quantization. The vector c_j represents the relative strength of the coupling. For a typical 6-color printer, we use a vector 0.5, 0.2, 0.2, 0.1, 0.1, 0.05 for the colorants black, cyan, magenta, light cyan, light magenta, and yellow, respectively.

The scalar form reduces the direct overprinting, but has little or no effect on overlap caused by dot expansion. Thus, we use a simple lowpass filter to blur the k_i signal above.

Multiplane optimization is only effective when the registration between planes is accurate. While this is generally not the case for offset and laser printing, fortunately it is usually the case on inkjets.

6. MULTILEVEL QUANTIZATION

Modern inkjets are capable of a number of a number of levels, rather than the on-or-off of traditional halftoning. Four levels (including none) is common today, and the number will probably increase.

Error diffusion extends readily to multilevel quantization. The traditional step function is replaced by a rounding function:

$$t_{x,y} = \lfloor (n-1)(g_{x,y} + e_{x,y} + tm_{x,y}) + 0.5 \rfloor \tag{7}$$

where n is the number of levels, $e_{x,y}$ is the filtered error term, and $tm_{x,y}$ is the sum of threshold modulation terms. However, when the threshold modulation is large, this form of quantization can produce undesirable effects, in particular the quantization to larger dots in transients near highlights. We address this by clamping the error and threshold modulation to ± 0.55 of one quantization level. This ensures that only the smallest dots are generated in highlights, while preserving the distribution of dot sizes at shades near integral dot sizes.

7. CONCLUSION

Even Toned Screening combines numerous techniques to produce high quality color dithering at practical speeds. The full source code is available under the GNU GPL free software license at the artofcode web site[4].

REFERENCES

[0] Steinberg, R. W., and Floyd, L., An adaptive algorithm for spatial greyscale. In Proceedings of the Society for Information Display, 17(2), pp. 75–77, 1976.

[1] Levien, R. Output dependent feedback in error diffusion. In IS&T 46th Annual Conference (Future of Imaging), Cambridge, Massachusetts, May 1993. Reprinted in Recent Progress in Digital Halftoning, Reiner Eschbach, ed., IS&T, 1995.

[2] Marcu, G. G. Error diffusion algorithm with output position constraints for homogenous highlight and shadow dot distribution. In Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts III, pp. 341–352, 1998.

[3] Eschbach, R. Error diffusion algorithm with homogenous response in highlight and shadow areas. In Journal of Electronic Imaging 06(03), pp. 348–356, July 1997.

[4] http://www.artofcode.com/eventone/

[5] Fawcett, G. S., and Schrack, G. F., Halftone techniques using error correction. In Proceedings of the Society for Information Display, 27, pp. 305–308, 1986.