Ink Relocation for Color Halftones

Doron Shaked*, Nur Arad[†], Andrew Fitzhugh[‡], Irwin Sobel[††]
HP Laboratories Israel*

HPL-96-127(R.1)
May, 1999

E-mail: dorons@hp.technion.ac.il
nur@orbotech.co.il
fitzhugh@cup.hp.com
sobel@hpl.hp.com

color halftoning,
halftone noise reduction,
color saturation

Current methodologies for color image halftoning produce prominent halftoning noise. In this paper we argue that brightness variation between color dots placed at neighboring locations is a major cause of color halftone noise. To correct this flaw we propose ***Ink Relocation***, a postprocess to arbitrary color halftoning algorithms. The proposed postprocess relocates ink drops between neighboring drop locations in order to reduce local brightness variation.

*Technion City, Haifa 32000, Israel
[†] Orbotech, P.O.B. 215, Yavne 81102, Israel
[‡] Hewlett-Packard Internet Imaging Operation, Cupertino, CA
[††] HP Laboratories, Palo Alto, CA

1 Introduction

Monochrome halftone algorithms are carefully designed to reduce visible artifacts. One of the most important factors producing those artifacts is the variation in the brightness of the dots. In monochrome (*Black* and *White*) halftones, this factor cannot be mitigated. Current color halftoning algorithms are usually Cartesian products of three halftoned monochrome planes corresponding to the color components of the image [7]. This generalization of monochrome algorithms overlooks the fact that colored dots are not equally bright.

To produce a good color halftone one has to place colored dots so that the following criteria are optimally met: (1) The placement pattern is visually unnoticeable. (2) The local average color is the desired one. (3) The colors used reduce the noticeability of the pattern. The first two design criteria are easily carried over from monochrome algorithms. However, the third cannot be satisfied by a simple Cartesian product generalization of monochrome halftoning.

Take for example, a solid patch of 50% *Gray*. Suppose some dot pattern (e.g., checkerboard) is selected. The patch could be equally rendered with *Black* and *White* dots as with *Blue* and *Yellow*, *Red* and *Cyan*, or *Green* and *Magenta* dots. As can be seen in Figure 1, the color of the halftoned patch will, theoretically, be the same in all cases. The noise effect, however, will be different, *Green* and *Magenta* being almost equally bright, in contrast to, for example, *Black* and *White*.

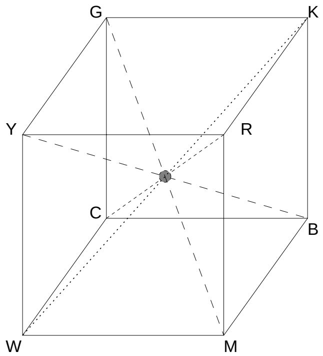


Figure 1: The *RGB* cube and its 4 main diagonals, meeting on 50% gray.

Arguments similar to the above have already been made in [5], where they motivated a distortion of the color space (as a pre-process to Error Diffusion).

In this report we analyze the problem systematically, and generalize it to arbitrary colors (rather than the simplistic example of gray tones or the extreme examples of not wanting dark halftones while rendering bright colors and vise-versa). Having proposed a criterion for optimal halftone colors, we present an algorithm to modify arbitrary halftones, optimizing them in terms of the new criterion.

Section 2 formulates the additional color criterion in practical terms by examining the simple case of rendering arbitrary solid color patches. Section 3 presents *Ink Relocation*, a postprocess which transforms arbitrary halftones to halftones conforming to the new color design criterion. Ink Relocation examples are presented and discussed in Section 4.

2 The Minimal Brightness Variation Criterion

In this section we formulate the color design criterion in more concrete terms. To this end we analyze it in a special case of rendering large patches of arbitrary solid colors. It is known that given a color in the *RGB* cube, it may be rendered using the 8 basic colors located at the vertices of the *RGB* cube. Actually, any particular color may be rendered using no more than 4 colors (in a linear color space, any color-quadruple whose convex hull contains the desired color will do). The issue raised in this section is: Suppose we want to print a patch of solid color, what colors should we use? Note that in previous work done on halftoning the issue was mainly what pattern should the dots be placed in, and when the issue of the participating halftone color was raised [5], it was mainly as an example of how bad things can get (e.g. Using *Black*halftones for bright colors). The following criterion gives the issue a full answer.

Consider the basic rationale of halftoning: When presented with high frequency patterns, the human visual system "applies" a low-pass filter and perceives only their average. Current inkjet printing resolution can still be resolved by the human visual system, thus still higher resolutions might have to be achieved. Relevant to the problem at hand is the fact that the human visual system is more sensitive to changes in brightness than to changes in the chrominance, which average at much lower frequencies. Thus we arrive at the following formulation of the third criterion:

The Minimal Brightness Variation Criterion (MBVC)
To reduce halftone noise, select from within all halftone sets by which the desired color may be rendered, the one whose brightness variation is minimal.

There are several standard "visually uniform" color spaces, and standard color difference measures [6]. The proposed Minimal Brightness Variation Criterion is not equivalent to choosing the set whose maximal difference measure is minimal. The rational behind our preference of an apparent one-dimensional projection (on the luminance axis) of a more general measure is that the visually uniform color spaces and the resulting color difference measures were developed with large solid color patches in mind, whereas halftone images are high frequency color patterns. Chrominance difference between participating colors plays part. However, due to the stronger low-pass in the chrominance channel, it matters much less than is embodied in the standard color difference formulas. We maintain that at current printing resolution the Minimal Brightness Variation Criterion is a reasonable one.

To consider the brightness variation of color sets we only need to order the eight basic colors on a brightness scale. In color theory [3] the primary colors (*Cyan*, *Magenta* and *Yellow*), and secondary colors (*Red*, *Green* and *Blue*) have a specific brightness order: *Blue* is the darkest secondary color, and *Green* the brightest. Their complementary colors *Yellow* (complements *Blue*) is the brightest primary, and *Magenta* (complements *Green*) is the darkest. Hence, we have two color orders: The "dark" colors $\{KBRG\}$, and the "bright" colors $\{MCYW\}$. The question is what is the combined brightness order.

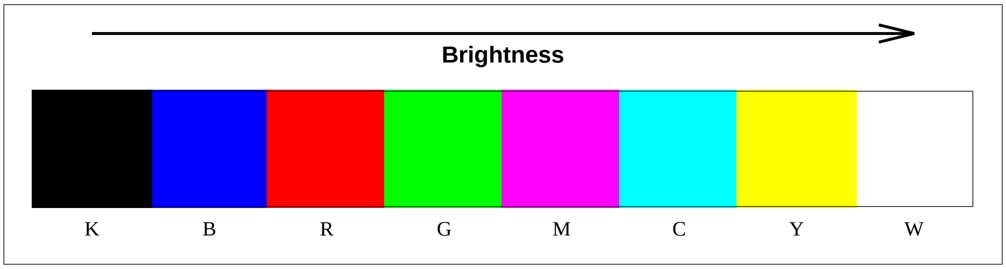


Figure 2: The brightness scale of the eight basic colors rendered using *HP* inkjet.

It would be only natural to assume that the bright colors are always brighter than the dark ones. Indeed, using *HP* inkjets this is the case, as can be seen in Figure 2. However, if other inks (or other media for that matter) are used, this may change. For example, colors rendered on a CRT screen have a different brightness order: $\{KBRMGCYW\}$, in which *Magenta* is darker than *Green*. It is easily seen that this permutation is actually the only one possible in three-color systems. Current printing technology produces *Red* as a two ink-drop overlay of *Magenta* and *Yellow*. Similarly Blue is an overlay of *Magenta* and *Cyan*. Thus *Magenta* cannot be darker than *Red* or *Blue*. *Green* cannot be brighter than *Cyan* or

Yellow, because it is produced as an overlay of the *Cyan* and *Yellow*. This argument is valid in a subtractive three-color system (printing). A similar argument may be formulated for additive color systems (CRTs).

3 The Ink Relocation Postprocess

In this section we introduce the Ink Relocation postprocess to arbitrary color halftone algorithms, e.g. Dithering [1], or Error-Diffusion [2]. Ink Relocation applies the Minimal Brightness Variation Criterion to arbitrary color halftones, so that the realization of first two design criteria embodied in the original halftoning algorithm is minimally interfered with.

Since we try to stick to the dot placement pattern of the original halftoning algorithm, we limit Ink Relocation's radius of influence to the minimum. The smallest digital neighborhood is a set of two adjacent pixels. Since we also need to maintain the local average color we limit Ink Relocation to relocation of ink-drops within sets of 2-pixel neighborhoods.

For example, consider the situation depicted in Figure 3, where two neighboring pixels are rendered *Black* and *White*(we model *Black* as composite *Black* - an overlay of *Cyan*, *Magenta*, and *Yellow*). In such a case, relocating the *Magenta* ink drop from the *Black* location to the empty *White* location constitutes a minimal infringement of the original dot placement pattern, and does not change the average color (50% *Gray*)[1]. The significant effect of the relocation is the reduced brightness variation (from *Black-White* to *Green-Magenta*).

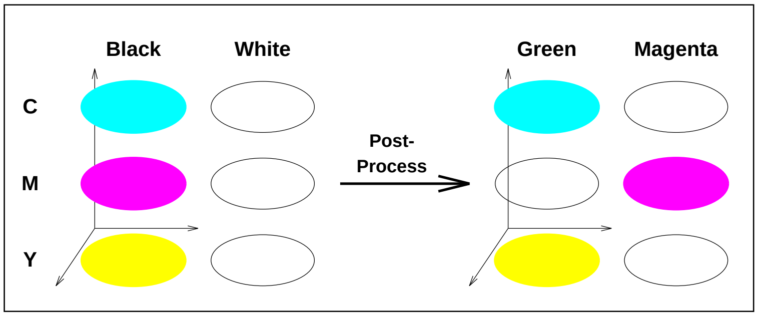


Figure 3: Relocation of the *Magenta* drop from the composite *Black* location to the empty *White* location.

---

[1] This is true only in a linear color space. In practice neither dot combinations is a true 50% *Gray*, and the average color is not identical. Standard tone correction is applied to correct for this illinearity.

Inspecting 2-pixel halftone neighborhoods, Ink Relocation considers whether the average color of the halftone couple can be rendered with a different halftone couple, whose brightness variation is smaller. It is easily verified that there are only 9 such halftone couples:

<table>
<tr><td>Black White</td><td colspan="6">KW → MG</td></tr>
<tr><td>Black Primary</td><td colspan="2">KY → RG</td><td colspan="2">KC → BG</td><td colspan="2">KM → BR</td></tr>
<tr><td>White Secondary</td><td colspan="2">WB → CM</td><td colspan="2">WR → YM</td><td colspan="2">WG → YC</td></tr>
<tr><td>Complementaries</td><td colspan="3">BY → MG</td><td colspan="3">RC → MG</td></tr>
</table>

All the 9 relocations amount to relocating a single ink drop from the high ink-load location to the neighboring low ink-load location. In some cases, like the example of Figure 3, there are several possible relocations. In creating the above table we made sure that the resulting halftone couple has the minimal possible brightness variation, and that the brightness gradient is maintained.

In the Ink Relocation postprocess, each pixel is sequentially compared to 4 of its immediate neighbors, as in Figure 4. The more general 8-neighborhood case can probably yield slightly better results, but was, nevertheless, ruled out because of computational considerations.

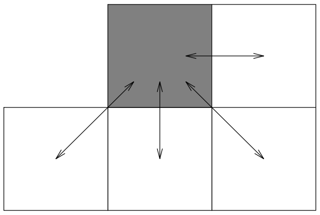


Figure 4: The neighborhood for Ink Relocation.

Figure 5 depicts the application of Ink Relocation to the outputs of two different halftoning algorithms (ordered dither and Error Diffusion). In both cases the dot placement characteristics have been preserved, as well as the average color, while the brightness variation has been significantly reduced.

The following section presents real size examples of both constant color patch examples and real images.

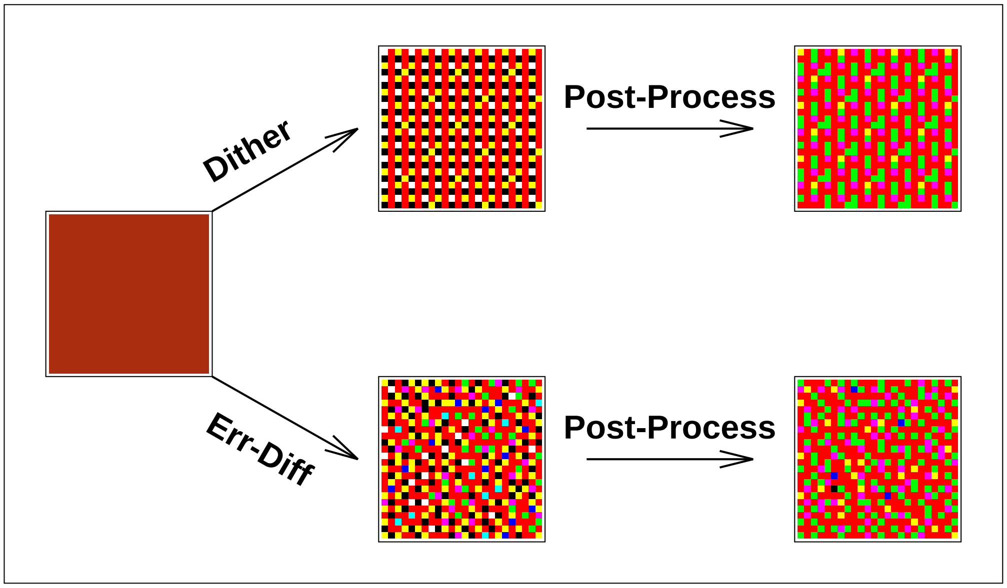


Figure 5: Enlarged example of the Ink Relocation postprocess, applied to a patch of solid color.

4 Results and Discussion

We conclude with a few Ink Relocation examples. Figure 6 presents three patches of solid color rendered with Error-Diffusion (top) and with Error-Diffusion+Ink Relocation (bottom). The reduction in halftone noise, and Error Diffusion artifacts (worms) is clearly visible (though highly dependent on the actual printer and media).

Figure 7 presents four samples of an image, rendered using 8×8 Bayer dither, Bayer dither+Ink Relocation, Error-Diffusion, and Error Diffusion+Ink Relocation. In this figure, the halftone noise and artifact reduction can be appreciated more easily than in Figure 6. The colors in Figure 7 where not corrected, though an initial tone correction was applied.

Alongside the improvement of halftone quality due to the Ink Relocation, postprocess Figure 7 depicts some of the problems introduced by the postprocess:

- Postprocessing of ordered dither halftones produces some disturbing artifacts. Those occur probably because of the interplay of the ordered dither with the structured

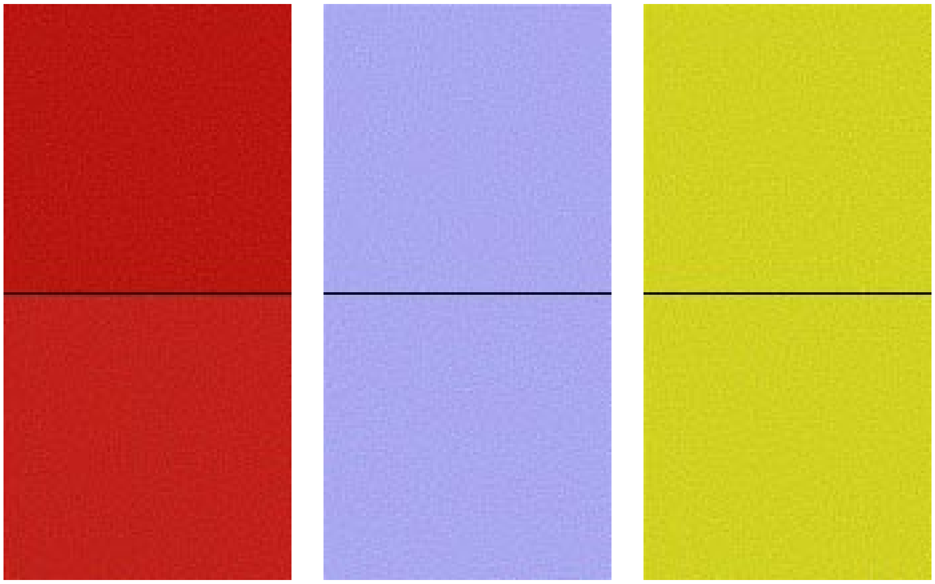


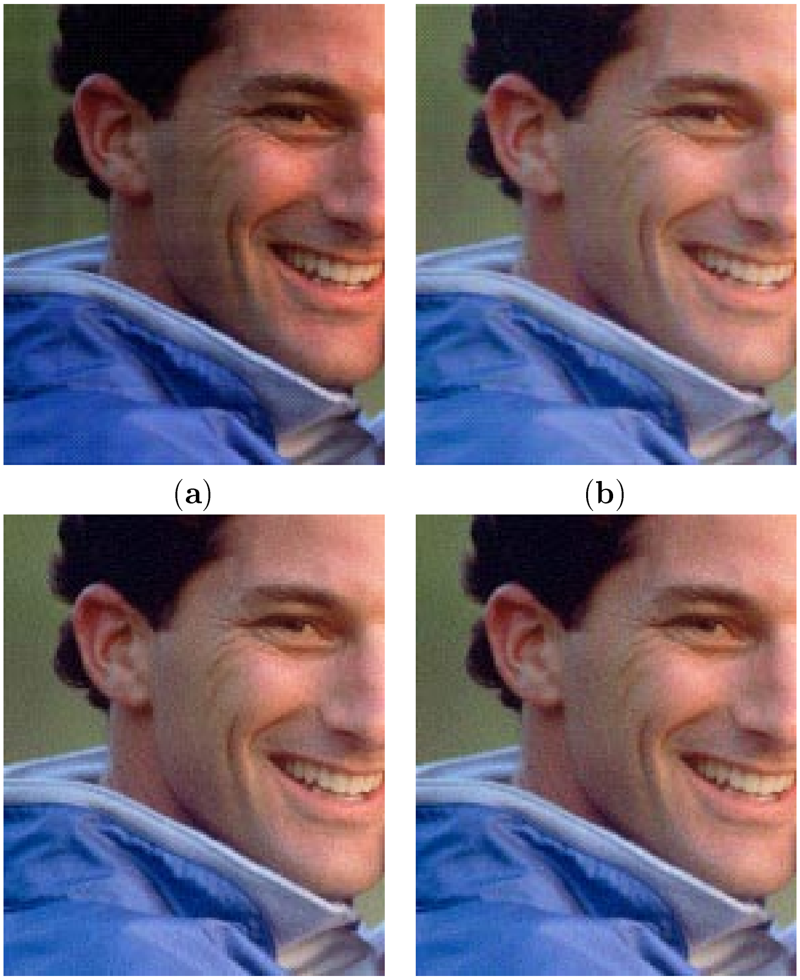
Figure 6: Three solid color patches rendered with Error-Diffusion (top), and Error Diffusion+Ink Relocation (bottom).

postprocess (the four neighbors in Figure 4 are visited in the same order). It may be possible to solve this problem if some randomization is introduced.

- An interesting expected byproduct of Ink Relocation is that colors are intrinsicly more saturated(see especially the ordered dither example in Figure 7(a)-(b)). This phenomenon is highly dependent on the media (paper type). Improved color saturation is expected because in Ink Relocation neutral dots (K and W) are discarded, in favor of saturated dots (R, G, B, C, M, or Y). Thus colors appear far from the neutral (*Gray*) axis (the dotted axis in Figure 1).

  It is interesting to note that similar claims about the colorfulness of "dull" tones had been made by the Neo-Impressionists more than a century ago [4]. Georges Seurat (1859–1891) an artist and art theoretician, developed his theories on pointilism (as an artistic method) at the dawn of color printing. Amongst others, he maintained that in order to render the effects of natural light and shadow, one has to apply dot combinations of complementary colors, rather than combinations involving *Black* and *White*. In fact, he and his followers banned *Black* from their pallet altogether.

- Any rendering method requires a color correction process. The uncorrected images in Figure 7 were intentionally left that way to present the intrinsic features of Ink Relo-



(**a**) (**b**)

(**c**) (**d**)

Figure 7: An image rendered using **(a)** Dither, **(b)** Dither+Ink Relocation, **(c)** Error Diffusion, **(d)** Error Diffusion+Ink Relocation.

cation. Figures 8(a) and 8(b) are the color corrected version of Figures 7(c) and 7(d) respectively.

- Ink Relocation is essentially a smoothing process. A relocated ink dot moves typically a single location, and may move up to two locations. Blurring effects are usually less disturbing than expected because of the edge enhancement implicit in halftone noise reduction. Nevertheless, blur can be noted at strong edges, where tone differences at the edge cause prominent ink relocation across it. The blur could be prevented if edges were detected, and Ink Relocation over them suppressed. Disabling the post-process over edges does not reduce the positive effect of Ink Relocation, which was designed to improve mainly the rendering of smooth areas. It is however important to note that the proposed blur reduction requires access to the original color values or the employment of a good inverse halftoning algorithm. In Figure 8(c) Ink Relocation was disabled on edges.

Figure 8: Color corrected results: **(a)** Error Diffusion, **(b)** Error Diffusion+Ink Relocation, **(c)** Ink Relocation disabled on edges.

Acknowledgment: The authors thank Prof. Bruckstein for calling their attention to the color theories of G. Seurat and the Neo-Impressionists.

5 References

[1] B. E. Bayer, “An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures”, in *Proceedings of the IEEE International Conference on Communication*, pages 11–26, 1973.

[2] R. Floyd and L. Steinberg, “An Adaptive Algorithm for Spatial Gray Scale”, in *SID Symposium*, pages 36–37, 1975.

[3] L. D. Grandis, *Theory and Use of Color*, Prentice Hall, Inc. and Harry N. Abrams, Inc., 1984, (Translated by J. Gilbert).

[4] W. I. Homer, *Seurat and the Science of Painting*, Hacker Art Books, 1985.

[5] R. V. Klassen, and R. Eschbach, Vector Error Diffusion in a Distorted Color Space”, in *Proc. IS&T’s 47th Annual Conf.*, 1994.

[6] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley and sons, 1982.

[7] A. Zakhor, S. Lin, and F. Eskafi, “A New Class of B/W and Color Halftoning Algorithms”, in *Proc. Int. Conf. on Acoustics Speech and Signal Processing*, 1991.