Color Diffusion: Error-Diffusion for Color Halftones *

Doron Shaked[†] Nur Arad[‡]
Hewlett Packard Laboratories Israel
Andrew Fitzhugh[§] Irwin Sobel[¶]
Hewlett Packard Laboratories Palo Alto

Keywords: Color halftoning, error diffusion, minimal brightness variation criterion.

Abstract

Error Diffusion is a high-performance halftoning method in which quantization errors are diffused to "future" pixels. Originally intended for grayscale images, it is traditionally extended to color images by Error-Diffusing each of the three color planes independently (separable Error Diffusion). In this report we show that adding to the Error Diffusion paradigm a simple design rule which is based on certain characteristics of human color perception results in a novel color halftoning algorithm, whose output is of considerable higher quality compared to separable Error Diffusion. The algorithm presented requires no additional memory and entails a reasonable increase in run-time.

---

*Patent application submitted. Monochrome version to appear at SPIE 3648

†HP Labs Israel, Technion City, Haifa 32000, Israel. dorons@hp.technion.ac.il

‡Current address: Orbotech, P.O.B. 215, Yavne, 81102, Israel. nur@orbotech.co.il

§Current address: Hewlett-Packard Internet Imaging Operation fitzhugh@cup.hp.com

¶HP Labs, 1501 Page Mill Rd. Palo-Alto, Ca. sobel@hpl.hp.com

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 binary halftones (*Black* and *White*), this factor cannot be mitigated. Current color halftoning algorithms are usually a Cartesian product 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 specifications are optimally met: (1) The placement pattern is visually unnoticeable. (2) The local average color is the desired color. (3) The colors used reduce the notice-ability 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.

The third design rule makes sure that the proper halftone colors are used in rendering a given input color. Arguments relating to the participating halftone colors were already made in [3], where they motivated a distortion of the color space as a pre-process to Error Diffusion.

In a previous Technical Report [4] we presented the *Minimal Brightness Variation Criterion (MBVC)*, a comprehensive formulation of the third design rule, based on some characteristics of the human visual system. The MBVC characterizes the set of participating halftone colors for given input color. Based on the MBVC, we derived *Ink Relocation*, a postprocess to arbitrary halftoning algorithms which was shown to improve their visual quality. Being a post-precess Ink Relocation balances the delicate trade-off between compliance with a new design criterion (the MBVC), and not wanting to modify the original halftone pattern. Thus, its output complies with the MBVC only in some local sense. In this report we propose *Color Diffusion*, a Vector Error Diffusion algorithm incorporating the MBVC design rule. Color Diffusion provides both full compliance with the MBVC and optimal halftone pattern (optimal in the Error Diffusion sense [5]).

The next section reviews briefly the MBVC and the motivation behind it. Section 3 determines the set of participating halftone colors complying with the MBVC for any given input color. Section 4 describes the proposed Color Diffusion algorithm, and Section 5 presents experimental results comparing Color Diffusion to separable Error Diffusion before and after enhancement due to the Ink Relocation postprocess.

2 The Minimal Brightness Variation Criterion

In this section we review the MBVC and its derivation. 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 cube. Actually, any color may be rendered using no more than 4 colors, different colors requiring different quadruples. Moreover the quadruple corresponding to a specific color is, in general, not unique (in a linear color space, any quadruple whose convex hull contains the desired color will do). The issue we raise 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 in [3], it was mainly as an example of how bad things can get. The MBVC 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 (up to 600 dpi) can still be resolved by the human visual system, thus still higher frequencies will 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 Minimal Brightness Variation Criterion for halftoning of solid color patches:

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 criterion is not equivalent to choosing the set whose maximal difference measure is minimal. The rationale 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 for large solid color patches. We, on the other hand, consider colors in a high frequency pattern. Chrominance differences between participating colors play 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, see Figure 1, and [4] for a short discussion of the subject.

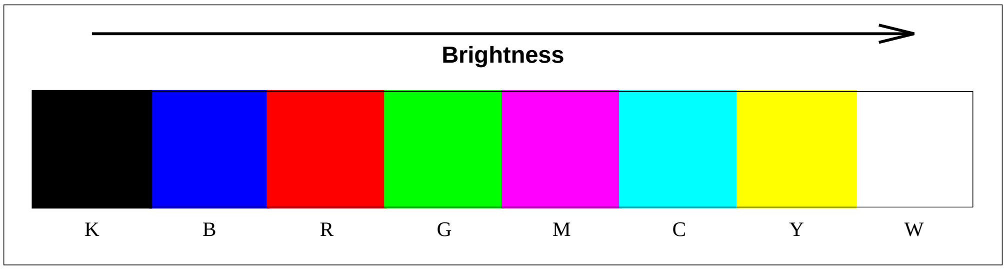

Figure 1: The brightness scale of the eight basic colors rendered using *hp* inks.

An interesting expected byproduct of using the MBVC is that color patches appear more saturated. Improved color saturation is expected because due to the MBVC, neutral dots (*K* or *W*) are used less, and wherever possible saturated dots (*R*, *G*, *B*, *C*, *M*, or *Y*) are used instead. Thus rendered patches appear far from the neutral (*Gray*) axis.

3 Minimal Brightness Variation Quadruples

According to the MBVC, colors have to be rendered using the halftone set whose brightness variation is minimal. In this section we determine the required set for any given input color.

Let us consider once more the simple example of large patches of solid color. In Separable Error Diffusion practically all 8 basic colors are used in rendering any solid color patch, their appearance ratio being some decreasing function of their distance from the desired color. However, the use of 8 colors (where 4 would suffice) stands in blunt contradiction to the MBVC (since for almost any solid color, *Black* and *White*, whose brightness variation is maximal, will be used).

In order to find the halftone set required by the MBVC, consider a large halftone pattern rendering an arbitrary solid input color. We show that transforming the halftone pattern so as to preserve the average color, and reduce the brightness variation, reduces the number of participating halftone colors to the minimum (i.e: 4). The resulting halftone quadruple is the required halftone set. We denote it as *Minimal Brightness Variation Quadruple* (MBVQ).

All this is done with Ink-Relocation-type transformations. In Ink Relocation [4], neighboring halftone couples are transformed to minimize their brightness variation, while preserving their average color. The notation

$$KW \rightarrow MG. \tag{1}$$

was used for the transformation of a *Black* and *White* halftone couple to *Green* and *Magenta*.

Since we do not need to preserve resemblance to the original halftone pattern, we perform (1) extensively, on all the halftones we can pair. The process ends when there are no more of either *Black* or *White* halftones (or both). This transformation reduces a great deal of the brightness variation, however it reduced the number of halftone colors participating in the rendering by one (to 7).

We proceed as follows: If (1) eliminated the *White* halftones, we transform couples of a *Black* and a primary colored halftone to couples of secondary colored halftones:

$$KY \rightarrow RG; \quad KC \rightarrow BG; \quad KM \rightarrow BR. \tag{2}$$

Now if a *Black* halftone still remains, the MBVQ is *RGBK*.

Alternatively, if (1) eliminated the *Black* halftones, we transform couples of a *White* and a secondary colored halftone to couples of primary colored halftones:

$$WB \rightarrow CM; \quad WR \rightarrow YM; \quad WG \rightarrow YC. \tag{3}$$

Now if a *White* halftone still remains, the MBVQ is *WCMY*.

If neither *Black* nor *White* remained, we are left with a halftone pattern consisting of potentially 6 halftone colors. To eliminate the two last halftone colors, the following two transformations select between *Blue* and *Yellow*, and between *Red* and *Cyan*:

$$BY \rightarrow MG; \quad RC \rightarrow MG. \tag{4}$$

This leaves us with 4 possible MBVQ's (according to which halftone colors were eliminated in (4)): MYGC, RGMY, RGBM, and CMGB.

We conclude that we can render every input color using one of six halftone quadruples: RGBK, WCMY, MYGC, RGMY, RGBM, or CMGB, each with obviously minimal brightness variation. Those quadruples partition the *RGB* cube into the six tetrahedra shown in Figure 2 (they cover the whole cube with no overlap). Considering the fact that a halftone quadruple can render only colors within its convex hull, it is obvious that the MBVQ of a given input color is the set of vertices of the tetrahedron it is located in. Thus, given an *RGB* triplet, one may compute its MBVQ by point location:

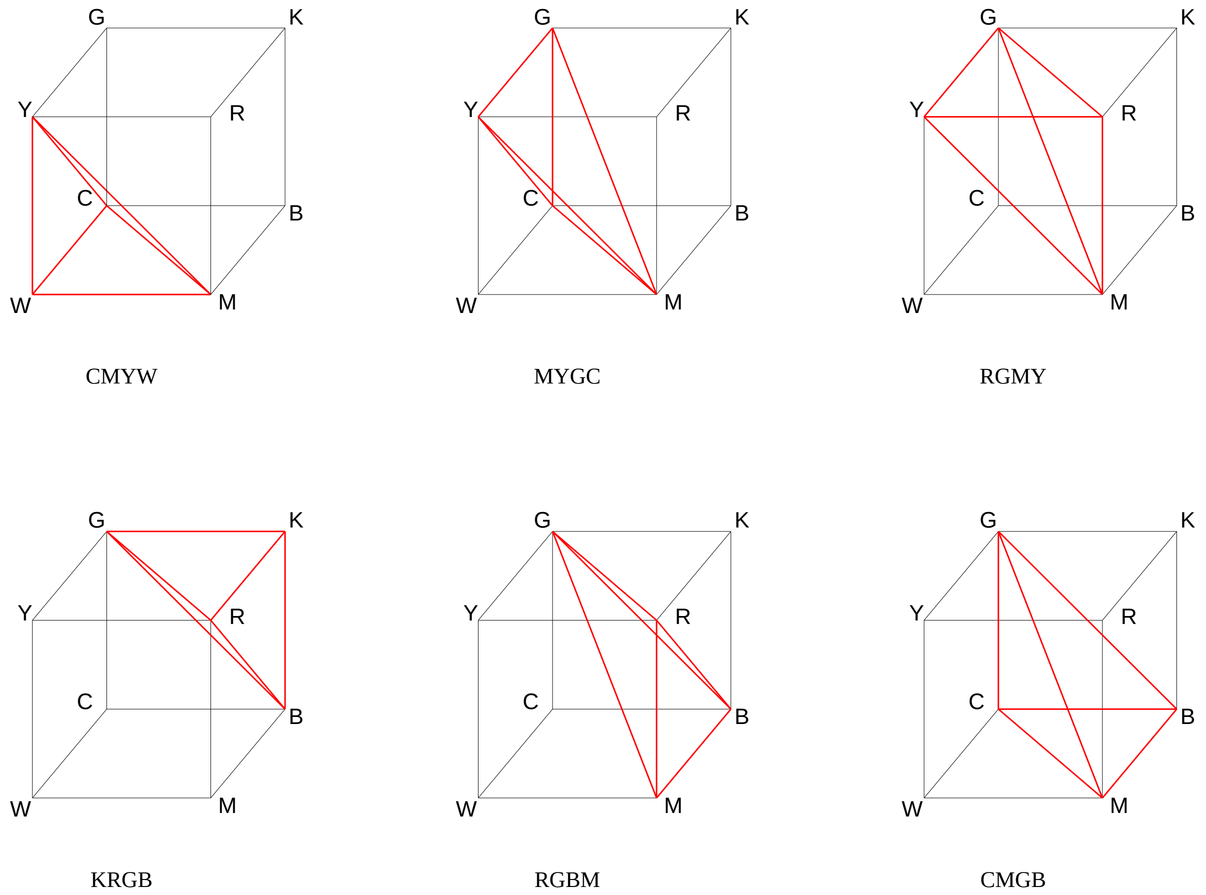

Figure 2: The partition of the *RGB* de toube to six tetrahedral volumes, each of which the convex hull of the MBVQ used to render colors in it. Note that all the tetrahedra are of equal volume, but are not congruent.

```
pyramid MBVQ(BYTE R, BYTE G, BYTE B)
{
  if((R+G) > 255)
    if((G+B) > 255)
```

```
      if((R+G+B) > 510)    return CMYW;
      else                 return MYGC;
    else                   return RGMY;
  else
    if(!((G+B) > 255))
      if(!((R+G+B) > 255)) return KRGB;
      else                 return RGBM;
    else                   return CMGB;
}
```

4 The Color Diffusion Algorithm

In the this section we introduce Color Diffusion, a vector modification to the standard Error Diffusion algorithm.

In the following, denote by $RGB(i, j)$ the RGB value at pixel (i, j) and by $e(i, j)$ the accumulated error at pixel (i, j). Color Diffusion may be formalized as follows:

For each pixel (i, j) in the image do:

1. Determine MBVQ($RGB(i, j)$).
2. Find the vertex $v \in$ MBVQ which is closest to $RGB(i, j) + e(i, j)$.
3. Compute the quantization error $RGB(i, j) + e(i, j) - v$.
4. Distribute the error to "future" pixels.

The only difference between separable Error Diffusion and Color Diffusion is in step (2), where the algorithm looks for the closest vertex in the MBVQ of the color, as opposed to the closest of the eight vertices of the cube. Thus, any separable Error Diffusion algorithm (regardless of the exact manner in which pixels are ordered and errors are computed or distributed) may be modified to Color Diffusion. In our implementation we used the Floyd Steinberg Error Diffusion [1].

The issue of finding the closest vertex $v \in$ MBVQ deserves special attention. When applying separable Error Diffusion each component of the RGB value is compared to the threshold value, 127, and a tessellation of $I\!R^3$ with respect to the eight vertices is formed. Note that the norm used to form the tessellation is not explicitly stated, but a closer look reveals that this is not necessary: Due to symmetry properties between the eight vertices any L^p ($1 \leq p \leq \infty$) norm gives rise to the same tessellation of $\mathbb{R}^3$ with respect to the eight vertices of the cube. The same does not hold when a tessellation relative to a proper subset of the eight vertices is called for: When restricted to the *RGB* cube itself, any given quadruple gives rise to the same tessellation regardless of the L^p norm used, however outside the cube these tessellations may differ. Easiest to compute is the L^2 tessellation, in which the decision planes inside the cube are actually valid for all of $\mathbb{R}^3$. Thus, for each of the six pyramids, determining the closest vertex to a given point entails traversing a decision tree of depth 3. A description of one of the trees is shown in Figure 3.

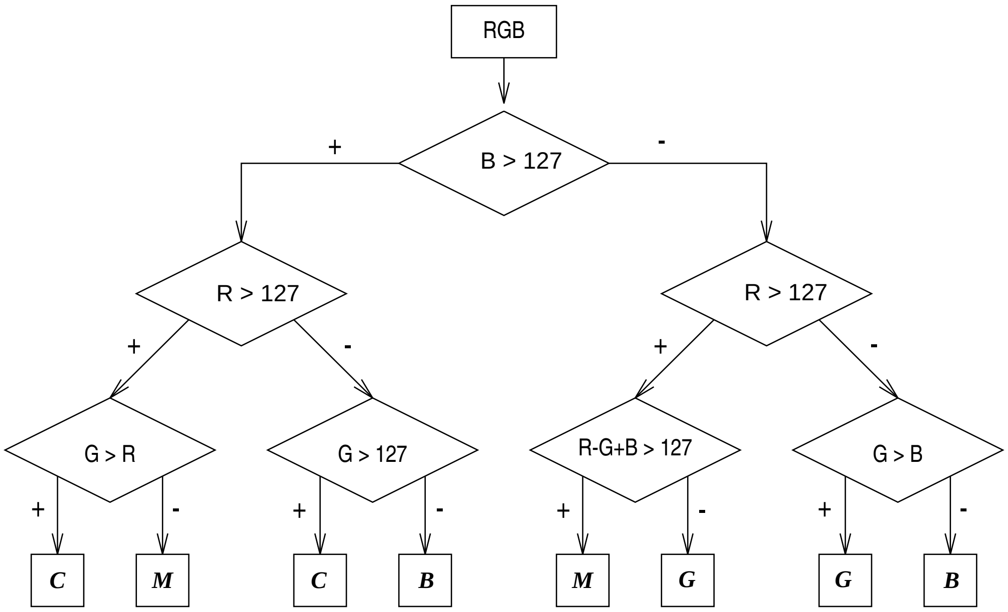

Figure 3: Decision tree for determining the tessellation of space relative to the vertices of the CMGB pyramid. All comparisons are of the type $x > 127$ or $x > y$ or $x - y + z > 127$, and the third type appears only once in every tree.

5 Examples

Figure 4 shows the application of several halftoning algorithms to a solid patch with value $RGB = (210, 40, 230)$, and printed at 75 dpi. The separable Error Diffusion algorithm (Figure 4(a)) renders the patch with all eight halftones. Dark halftones adjacent to light ones are abundant. Application of the Ink Relocation post-process (Figure 4(b)) reduces brightness variation, however, rendering is still done with all eight halftones (*Black* appears only once - due to boundary effects, and *White* is very rare). When applying Color Diffusion to the original patch, only 4 colors (B, C, G, M) are used, and halftone noise is virtually brought to a minimum, also the dot pattern is more regular (in the Error Diffusion sense [5]).

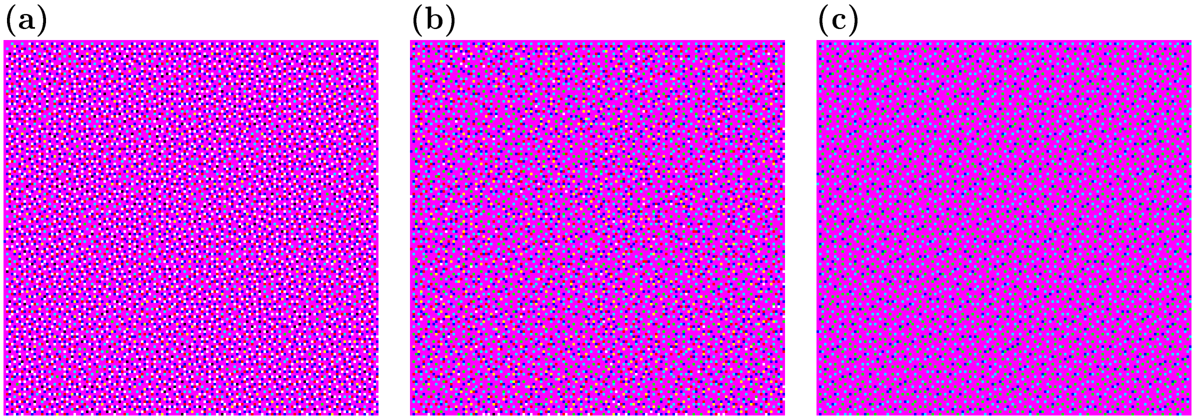

Figure 4: A solid color patch rendered at 75 dpi using different halftoning methods. **(a)** Separable Error Diffusion. **(b)** The Ink Relocation post-process applied to separable Error Diffusion. **(c)** Color Diffusion. Note the decrease in halftone noise from left to right.

Figure 5 shows the application of the same halftoning methods to high-resolution natural images.

Regarding run-time, our results are as follows: Supposing separable Error Diffusion takes one unit of time, adding the Ink Relocation post-process takes 1.4 units, and Color Diffusion takes 1.55 units.

Acknowledgment: The authors thank Michael McGuire of Hewlett Packard Laboratories for the discussions we had and for his implementation ideas.

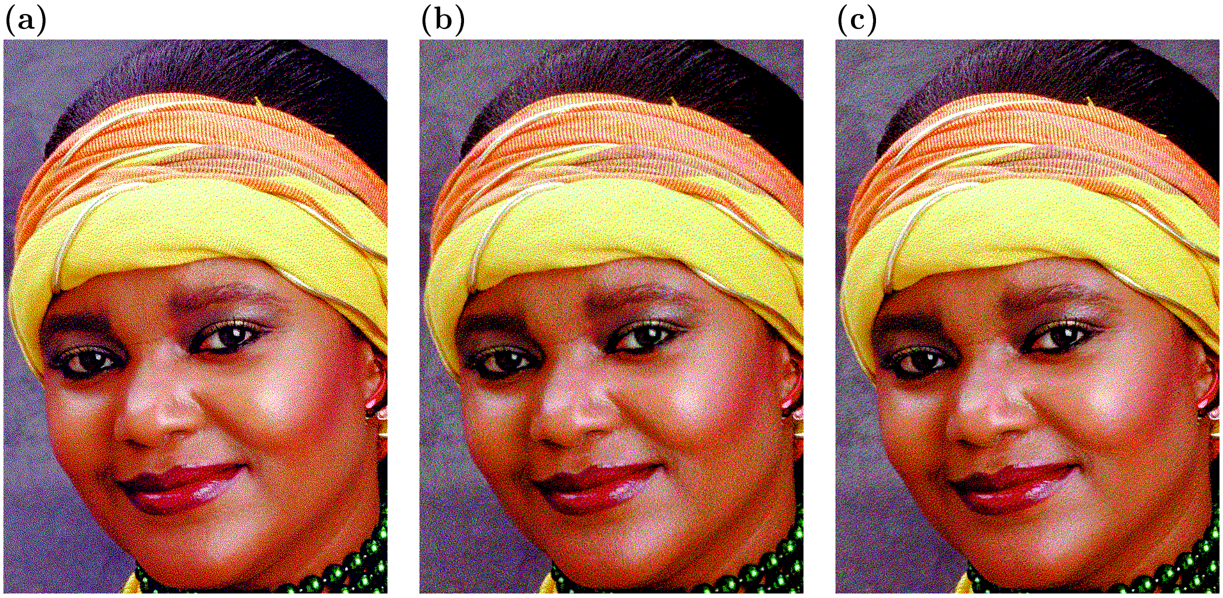

Figure 5: A high-resolution image rendered using different halftoning methods. **(a)** Separable Error Diffusion. **(b)** The Ink Relocation post-process applied to separable Error Diffusion. **(c)** Color Diffusion. Note the decrease in halftone noise from left to right.

5 References

[1] R. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Gray Scale", in *SID Symposium*, pp. 36–37, 1975.

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

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

[4] D. Shaked, N. Arad, A. Fitzhugh, and I. Sobel, "Ink Relocation for Color Halftones", *Hewlett-Packard Laboratories Technical Report*, August 1996. Proceedings version in *Proc. of IS&T's Image Processing Image Quality and Image Capture Conf.*, May 1998.

[5] R. A. Ulichney, "Dithering with Blue Noise", *Proc. of the IEEE*, Vol. 76, pp. 56–79, 1988.

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

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