

CMYK Halftoning Algorithm Based on Direct Binary Search[†]

Je-Ho Lee^{‡§} and Jan P. Allebach
*School of Electrical and Computer Engineering
 Purdue University, West Lafayette, IN 47907-1285*

Abstract

In this paper, we propose color design criteria for uniform color textures, and present a new colorant-based halftoning technique which meets these criteria, and hence generates high quality color halftones. The basic idea behind our halftoning method is to control the quality of each colorant texture separately along with the total dot distribution. In order to achieve this, we first set the total dot arrangement and then color the dots optimally without altering the total dot arrangement. This corresponds to a constrained optimization problem, and we solve this via the swap-only direct binary search (DBS) heuristic. Examples of color halftone images using our technique are compared to images obtained by a plane-independent color halftoning algorithm and by the HP 970 Cx inkjet printer driver.

1. Introduction

Digital color halftoning is the process of generating a pattern of pixels with a limited number of colors that creates the illusion of a continuous-tone image. Color halftoning presents many problems that we encounter in monochrome halftoning. However, it also presents many problems that are unique to color, mainly due to the interactions between color planes. A recent trend in color halftoning research has been toward approaches which explicitly account for the differences in pattern visibility between textures comprised of different sets of colorants to reduce artifacts that are unique to color halftoning. Take, for example, a solid patch of 50% gray. The patch could be rendered with black and white, yellow and blue, cyan and red, or magenta and green dots. The color of the halftoned patch will, theoretically, be the same in all cases. However, noise effects will be different. In order to yield smoother, less visible textures, several authors have considered approaches to constraining the primaries that are locally printed to yield a particular visually averaged color [1] [2] [3] [4] [5]. Lau

et al introduced the concept of stochastic moiré in their recent paper [6], which is the aperiodic interference pattern created by superimposing the aperiodic halftones of frequency modulated screens. This stochastic moiré characterizes the spatial fluctuations in color/texture of a halftone image. The concept of stochastic moiré is closely related to the color design criteria that we present in this paper.

In this paper, we take a DBS based color halftoning approach as in [7] [8]. But whereas these references use a complex color HVS model and operate in a linearized version of the CIE $L^*a^*b^*$ uniform color space, our algorithm operates directly in the *CMY* colorant space. This gives us explicit control over the individual colorant textures and the way in which these textures interact to determine the overall visual appearance of the halftone. It allows us to directly minimize dot-on-dot printing. In contrast, references [7] and [8] rely completely on the color HVS model to guide DBS toward good choices for the individual and combined colorant textures. Explicit control of the amount of each colorant used for a given *RGB* triplet is achieved by a pre-halftoning calibration process. In [1], Lin and Allebach took a similar approach when they jointly designed a set of threshold arrays for color screening. They designed the screens along the white-black neutral axis under the criterion that the monochrome pattern, as well as its decompositional color patterns be smooth and uniform. They used a color HVS model similar to that employed in [7] [8]. We employ only a luminance HVS model to enforce exclusion of visually non-homogeneous patterns for the total dot distribution, as well as for each individual colorant texture. We also present a technique for mapping the color design criteria defined in terms of constant patches to real images.

The remainder of this paper is organized as follows. In Sec. 2, we describe our uniform color criteria and the colorant-based halftoning algorithm. We also discuss how we have used DBS with the uniform color criteria. Then, we show some experimental results in Sec. 3. We will conclude the paper in Sec. 4.

[†]Research supported by the Hewlett-Packard Company.

[‡]This work was performed while Je-Ho Lee was a graduate student at Purdue University.

[§]The author is now with the Hewlett-Packard Company, 18110 SE 34th Street, Vancouver, WA 98607.

2. Colorant-based DBS algorithm

2.1. Criteria for Uniform Color Textures

In this paper, we restrict our discussion to bi-level 3-colorant *CMY* printers. Our algorithm, however, can also be applied directly to bi-level 4-colorant *CMYK* printers where full undercolor removal is employed, *i.e.* at a given pixel a black dot replaces cyan, magenta, and yellow dots if and only if all three are present there. The proposed halftoning algorithm is based upon the *CMY* colorant space. Therefore, we first need to perform *RGB-to-CMY* conversion on the original continuous-tone color image. Aside from the confounding complexity of color interpretation by the HVS, *RGB-to-CMY* conversion for printing has proven difficult because of the non-linear and non-ideal behavior of printer colorants [9]. Generally, halftoning researchers have favored simpler methods such as the ‘one-minus-*RGB*’ method that is based on an idealized model of block colorants with unity transmittance in all but non-overlapping bands, *i.e.* only one colorant absorbs at any wavelength [8] [10]. We also employ this one-minus-*RGB* method for the conversion:

$$\begin{aligned} C &= 1 - R, \\ M &= 1 - B, \\ Y &= 1 - G. \end{aligned} \quad (1)$$

We denote a continuous-tone color image by 3 scalar-valued functions $f_i[\mathbf{m}] : \mathcal{R}^2 \mapsto \mathcal{R}$, $i = C, M, Y$. Similarly, we denote a color halftone image by $g_i[\mathbf{m}]$, $i = C, M, Y$. We assume $f_i[\mathbf{m}]$ takes values in units of absorbance between 0 and 1, where 0 means that there is no colorant and 1 means that there is a maximum amount of colorant present on the paper. The digital halftone image $g_i[\mathbf{m}] = 0$ or 1.

As mentioned in Sec. 1, the approach that we take here to color halftoning focuses explicitly on the appearance of the textures corresponding to each colorant, both separately and in combination. We treat the dots of each colorant as a distinct type of object, and strive to arrange these objects in as uniform and homogeneous a manner as possible, mostly without regard for the colorimetric properties of the individual colorants. Our HVS model is only used to generate a measure of the visual quality of the manner in which the objects are arranged. We have identified three criteria of texture quality that are appropriate to this approach:

1. If we consider the dots of each color (C , M , Y , R , G , B , or K) individually, we would like these dots to be arranged as uniformly as possible.
2. If we treat the dots of the seven possible colors as being an identical type of object, we would like the overall composite texture consisting of all these objects to be as uniform as possible.
3. We wish to minimize dot-on-dot printing as much as possible.

The first two criteria are fairly self-evident. To understand the third criterion, consider a situation in which we wish to print a 100×100 pixel patch of constant gray with 7% absorbance. To achieve this level of gray, we will need to print 700 dots each of cyan, magenta, and yellow. If we constrain the printing so that the cyan, magenta, and yellow dots are printed on top of each other, we will have 700 black dots, which by enforcing Criteria 1 and 2 above, can be arranged in a uniform pattern with 7% fill. This situation is maximally *dot-on-dot*. At the other extreme, we constrain the textures so that no dots of different colorants may be printed on top of each other. In this case, enforcement of Criteria 1 and 2 will result in a homogeneous texture consisting of a mixture of 2100 cyan, magenta, and yellow dots. The overall fill is now 21%, which will be perceived as a more uniform texture than a 7% fill, regardless of the color of the individual dots. In addition, each of the cyan, magenta, or yellow dots will contrast less with the white substrate than will the black dots, further reducing the overall visibility of the halftone texture. We refer to this case as maximally *dot-off-dot*. Figure 1 compares dot-on-dot and dot-off-dot printing for a 7% gray level. In addition to producing a smoother, more homogeneous texture [1] [11], dot-off-dot printing yields a larger color gamut [11]. For darker colors, it is necessary to have some dot-on-dot printing. This results in dots with secondary colors, such as the blue obtained by printing cyan on top of magenta. According to Criterion 1, we would like each texture resulting from just the dots with a single secondary color to be arranged as uniformly as possible.

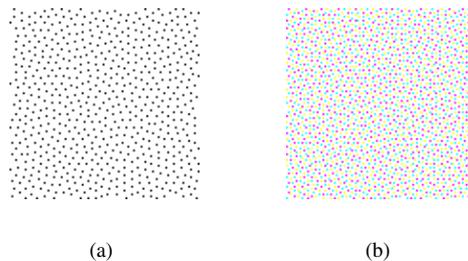


Figure 1: Two halftone patterns illustrating (a) dot-on-dot, and (b) dot-off-dot printing for a 7% gray level printed at 100 dpi.

Now let us consider in some detail how we actually can design halftone textures for constant patches according to our criteria. Suppose we want to halftone a color image which is a constant patch consisting of 20% cyan, 20% magenta, and 40% yellow. Without any dot-on-dot printing, the dot area coverage will be $20 + 20 + 40 = 80\%$. According to our criteria, we want to generate *CMY* halftones which are individually homogeneous and for which the total dot distribution covering 80% of the entire image is also as uniform as possible. To achieve this goal, we first set a

homogeneous binary pattern for the total dot arrangement, and then color the dots so that each plane has uniform texture, without altering the total dot distribution.

However, we have found that better overall halftone quality is obtained when we control only the total distribution of the C and M dots, and halftone Y completely independently. This is because the Y colorant is much less visible than those of C and M . By not including Y in the constraint for C and M , we obtain a better total CM texture. Therefore, in our color halftoning scheme, we jointly halftone only the C and M planes, do the Y plane independently, and finally combine their halftones. For the above halftoning example, we establish the dot arrangement for CM by generating a $20 + 20 = 40\%$ fill bitmap image, color the bitmap with C and M so that each of C and M halftones has uniform texture, and overlay the independently generated Y halftone onto them.

Before extending our analysis to more general situations, let us consider another halftoning example for a constant-tone image which consists of 80% cyan, 50% magenta, and 0% yellow. We exclude Y from this example since we are interested now only in the C and M components for the total dot distribution constraint. The sum of $C + M$ is greater than 100%, which implies that at least some portions of C and M have to be piled up together thereby introducing the color blue (B). Let $C' = C - B$ and $M' = M - B$ denote the portions of pure cyan and magenta colors in the halftone image, respectively. To avoid dot-on-dot printing as much as possible, C' , M' , and B are given by

$$\begin{aligned} C' &= 1 - M, \\ M' &= 1 - C, \\ B &= C + M - 1. \end{aligned} \quad (2)$$

The color halftoning process for this example is quite similar to that for the previous example. We set the dots for $C' + M'$, optimally color each of dots either by cyan or magenta, and finally fill the holes by blue. Figure 2 shows bitmaps for this example produced at the four stages of our halftoning scheme.

2.2. Extension to General Images

The above two examples give us an idea how to extend our approach to real imagery. We begin by defining additional grayscale functions

$$\begin{aligned} f_{C'}[\mathbf{m}] &= \begin{cases} f_C[\mathbf{m}], & \text{if } f_C[\mathbf{m}] + f_M[\mathbf{m}] \leq 1, \\ 1 - f_M[\mathbf{m}], & \text{otherwise,} \end{cases} \\ f_{M'}[\mathbf{m}] &= \begin{cases} f_M[\mathbf{m}], & \text{if } f_C[\mathbf{m}] + f_M[\mathbf{m}] \leq 1, \\ 1 - f_C[\mathbf{m}], & \text{otherwise,} \end{cases} \\ f_{CM}[\mathbf{m}] &= f_{C'}[\mathbf{m}] + f_{M'}[\mathbf{m}], \end{aligned} \quad (3)$$

where $f_{C'}[\mathbf{m}]$ and $f_{M'}[\mathbf{m}]$, respectively, are the amount of the cyan and magenta components which do not contribute

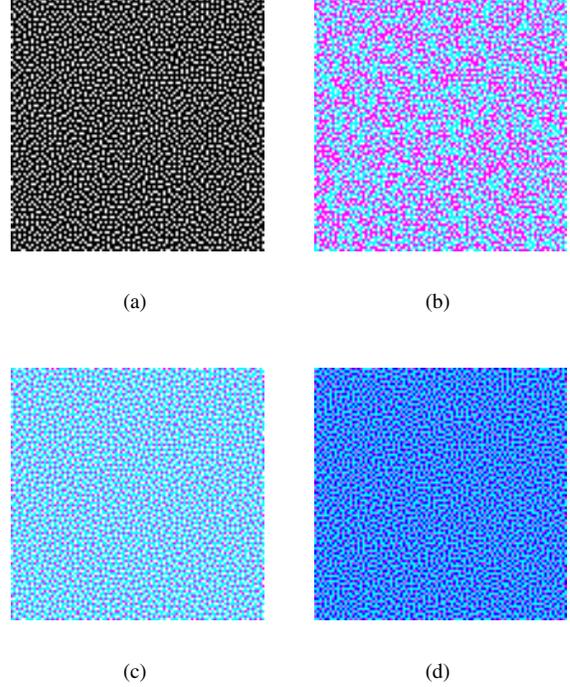


Figure 2: The steps of our color halftoning algorithm for the case where $C = 50\%$, $M = 80\%$, and $Y = 0\%$: (a) set overall dot arrangement, (b) color the dots randomly, (c) refine the halftones for C' and M' , and (d) fill holes by B .

to blue. Note that $0 \leq f_{CM}[\mathbf{m}] \leq 1$. In order to optimally set dots for CM , we apply the monochrome DBS halftoning algorithm to $f_{CM}[\mathbf{m}]$. We denote the halftone version of $f_{CM}[\mathbf{m}]$ by $g_{CM}[\mathbf{m}]$. Then, the 0-pixels of $g_{CM}[\mathbf{m}]$ will be either W or B , while the 1-pixels of $g_{CM}[\mathbf{m}]$ will be either C or M . Let $s_{CM}[\mathbf{m}] : \mathcal{R}^2 \mapsto \{W, C, M, B\}$ be the function that represents the coloring of $g_{CM}[\mathbf{m}]$. To the 0-pixels of $g_{CM}[\mathbf{m}]$, we assign B if $f_C[\mathbf{m}] + f_M[\mathbf{m}] \geq 1$; otherwise we assign W . We initially color the 1-pixels by random thresholding. At this stage, $s_{CM}[\mathbf{m}]$ is given by

$$s_{CM}[\mathbf{m}] = \begin{cases} W, & \text{if } g_{CM}[\mathbf{m}] = 0 \text{ and } f_C[\mathbf{m}] + f_M[\mathbf{m}] < 1, \\ B, & \text{if } g_{CM}[\mathbf{m}] = 0 \text{ and } f_C[\mathbf{m}] + f_M[\mathbf{m}] \geq 1, \\ C, & \text{if } g_{CM}[\mathbf{m}] = 1 \text{ and } \frac{f'_C[\mathbf{m}]}{f_C[\mathbf{m}] + f'_M[\mathbf{m}]} > r, \\ M, & \text{otherwise.} \end{cases} \quad (4)$$

Here r is a random variable which is uniformly distributed in the interval $(0, 1)$. Starting with this random initial pattern, we iteratively seek an optimal color pattern for $s_{CM}[\mathbf{m}]$ by swapping C and M pixels to reduce an error metric. It should be pointed out that during this process, the monochrome dot pattern of $g_{CM}[\mathbf{m}]$ is not altered. Every 0- and 1-pixel retains its original state, since we only consider swaps between C and M pixels, both of which belong to the 1-

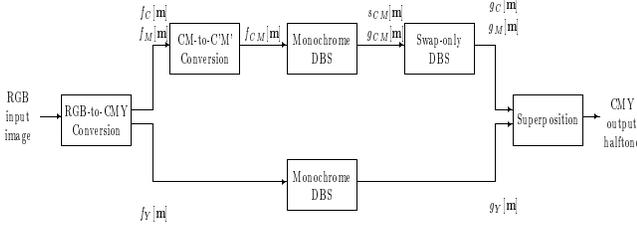


Figure 3: Block diagram for the flow of the proposed halftoning algorithm.

pixels of $g_{CM}[\mathbf{m}]$. Thus, the dot distributions of W and B remain unchanged. After the algorithm has converged, *i.e.* no changes are made during a single iteration, $g_C[\mathbf{m}]$ and $g_M[\mathbf{m}]$ are obtained from $s_{CM}[\mathbf{m}]$ as follows:

$$\begin{aligned} g_C[\mathbf{m}] &= \begin{cases} 1, & \text{if } s_{CM}[\mathbf{m}] = C \text{ or } B, \\ 0, & \text{otherwise,} \end{cases} \\ g_M[\mathbf{m}] &= \begin{cases} 1, & \text{if } s_{CM}[\mathbf{m}] = M \text{ or } B, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (5)$$

Figure 3 summarizes the flow of the proposed halftoning algorithm. In the next section, we will describe the error metric and the error minimization technique used in the proposed algorithm.

2.3. Error Metrics

Let $g_{C'}[\mathbf{m}]$ and $g_{M'}[\mathbf{m}]$ denote the halftone images for $f_{C'}[\mathbf{m}]$ and $f_{M'}[\mathbf{m}]$, respectively. It is important to appreciate the relationships among $g_{C'}[\mathbf{m}]$, $g_{M'}[\mathbf{m}]$, and $s_{CM}[\mathbf{m}]$:

$$\begin{aligned} g_{C'}[\mathbf{m}] &= 1 \quad \text{iff } s_{CM}[\mathbf{m}] = C, \\ g_{M'}[\mathbf{m}] &= 1 \quad \text{iff } s_{CM}[\mathbf{m}] = M. \end{aligned} \quad (6)$$

We assume that the printer spot overlap is insignificant or the overlap is additive if it is not negligible. It is possible to relax these assumptions by using non-linear dot interaction models, as is done in [8] [10]. The halftone image rendered by the printer is then given by

$$g_i(\mathbf{x}) = \sum_{\mathbf{m}} g_i[\mathbf{m}]p(\mathbf{x} - \mathbf{X}\mathbf{m}), \quad i = C', M', \quad (7)$$

where $p(\mathbf{x})$ is the spot function of the printer and \mathbf{X} is the periodicity matrix whose columns comprise the basis for the lattice of printer addressable dots. We define the perceived error images as

$$\tilde{e}_i(\mathbf{x}) = h(\mathbf{x}) * (g_i(\mathbf{x}) - f_i(\mathbf{x})), \quad i = C', M'. \quad (8)$$

Here $*$ denotes the convolution operation, $h(\mathbf{x})$ represents the point spread function of the HVS, and $f_i(\mathbf{x})$ is the continuous-tone original image. We will assume that the continuous-tone original images $f_i(\mathbf{x})$ can also be expressed

in terms of their samples $f_i[\mathbf{m}]$ according to (7). Equation (8) can then be expressed as

$$\tilde{e}_i(\mathbf{x}) = \sum_{\mathbf{m}} e_i[\mathbf{m}]\tilde{p}(\mathbf{x} - \mathbf{X}\mathbf{m}), \quad i = C', M', \quad (9)$$

where

$$e_i[\mathbf{m}] = g_i[\mathbf{m}] - f_i[\mathbf{m}], \quad i = C', M', \quad (10)$$

and $\tilde{p}(\mathbf{x}) = h(\mathbf{x}) * p(\mathbf{x})$ embodies the effect of cascading the printer rendering and HVS models. In general, the printer spot profile $p(\mathbf{x})$ is sufficiently narrow compared to $h(\mathbf{x})$ so that $\tilde{p}(\mathbf{x}) \approx h(\mathbf{x})$. This is the situation when output from a printer with at least moderate resolution is viewed at a normal distance. Throughout this paper, we will assume that this relationship holds and refer to the point spread function of the HVS as $\tilde{p}(\mathbf{x})$. Our error measure is based on the total squared value of (9), which is given by

$$\mathcal{E}_i = \int |\tilde{e}_i(\mathbf{x})|^2 d\mathbf{x}, \quad i = C', M'. \quad (11)$$

The problem of optimally coloring $s_{CM}[\mathbf{m}]$ is to minimize $\mathcal{E}_{C'}$ and $\mathcal{E}_{M'}$ within the context of $g_{CM}[\mathbf{m}]$. This problem can be stated as

$$\text{minimize } \mathcal{E}, \quad (12a)$$

$$\text{subject to } g_{C'}[\mathbf{m}] + g_{M'}[\mathbf{m}] = g_{CM}[\mathbf{m}], \quad (12b)$$

where $\mathcal{E} = \alpha\mathcal{E}_{C'} + \beta\mathcal{E}_{M'}$ is a dual error measure which penalizes the use of both C and M halftone textures that lead to a visually unpleasing appearance. Here α and β are weights for C and M , respectively. The constraint (12b) ensures that the monochrome pattern of $g_{CM}[\mathbf{m}]$ remains unchanged.

2.4. Error Minimization

As mentioned earlier, the method we use to solve the constrained optimization problem (12a) is swap-only DBS. Starting with the initial $s_{CM}[\mathbf{m}]$ which has random C and M assignments, we iteratively visit each C and M pixel and swap it with any of its neighbors within a 7×7 window that differ in color state. The swap takes place only between C and M pixels. With smaller window sizes such as a 3×3 window, we observed color shifts due to an insufficient search area. We accept that trial change, if any, which most reduces the value of \mathcal{E} . When no changes are accepted during a single iteration, the algorithm has converged. Note that the swap between pixels \mathbf{m}_0 and \mathbf{m}_1 in $s_{CM}[\mathbf{m}]$ causes changes to $g_{C'}[\mathbf{m}_0]$, $g_{C'}[\mathbf{m}_1]$, $g_{M'}[\mathbf{m}_0]$, and $g_{M'}[\mathbf{m}_1]$. We can reduce the complexity for computing the effect of trial halftone changes by using an approach similar to that proposed in [12] [13]. To develop

this more efficient method to evaluate the effect of a trial change, we first define the autocorrelation function

$$c_{\tilde{p}\tilde{p}}(\mathbf{x}) = \int \tilde{p}(\mathbf{y})\tilde{p}(\mathbf{x} + \mathbf{y})d\mathbf{y}. \quad (13)$$

Then we can express (11) in the discrete form

$$\mathcal{E}_i = \sum_{\mathbf{m}} \sum_{\mathbf{n}} e_i[\mathbf{m}]e_i[\mathbf{n}]c_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{n}], \quad i = C', M', \quad (14)$$

with the correlation function $c_{\tilde{p}\tilde{p}}[\mathbf{m}] = c_{\tilde{p}\tilde{p}}(\mathbf{X}\mathbf{m})$. Consider a trial swap between pixels \mathbf{m}_0 and \mathbf{m}_1 . The changes in $g_{C'}[\mathbf{m}]$ and $g_{M'}[\mathbf{m}]$ are described by

$$\Delta g_i[\mathbf{m}] = a_i^0 \delta[\mathbf{m} - \mathbf{m}_0] + a_i^1 \delta[\mathbf{m} - \mathbf{m}_1], \quad i = C', M', \quad (15)$$

where a_i^0 and a_i^1 , $i = C', M'$ are given by

$$a_i^j = \begin{cases} 1, & \text{if } g_i[\mathbf{m}_j] \text{ is changed from 0 to 1,} \\ -1, & \text{if } g_i[\mathbf{m}_j] \text{ is changed from 1 to 0,} \end{cases} \quad j = 0, 1. \quad (16)$$

Since the pairs $(g_{C'}[\mathbf{m}_0], g_{C'}[\mathbf{m}_1])$, $(g_{M'}[\mathbf{m}_0], g_{M'}[\mathbf{m}_1])$, $(g_{C'}[\mathbf{m}_0], g_{M'}[\mathbf{m}_0])$, and $(g_{C'}[\mathbf{m}_1], g_{M'}[\mathbf{m}_1])$ must all have different binary states, we have the following relationships among a_i^0 and a_i^1 , $i = C', M'$,

$$a_{C'}^0 = a_{M'}^1 = -a_{C'}^1 = -a_{M'}^0. \quad (17)$$

It can be shown [12] [13] that the changes in error $\Delta\mathcal{E}'_{C'}$ and $\Delta\mathcal{E}'_{M'}$ due to a trial swap described by (15) are given by

$$\Delta\mathcal{E}_i = ((a_i^0)^2 + (a_i^1)^2)c_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2a_i^0 c_{\tilde{p}\tilde{e}}^i[\mathbf{m}_0] + 2a_i^1 c_{\tilde{p}\tilde{e}}^i[\mathbf{m}_1] + 2a_i^0 a_i^1 c_{\tilde{p}\tilde{p}}[\mathbf{m}_1 - \mathbf{m}_0], \quad i = C', M', \quad (18)$$

where $c_{\tilde{p}\tilde{e}}^i[\mathbf{m}]$ is the cross-correlation between $e_i[\mathbf{m}]$ and $c_{\tilde{p}\tilde{p}}[\mathbf{m}]$

$$c_{\tilde{p}\tilde{e}}^i[\mathbf{m}] = \sum_{\mathbf{n}} e_i[\mathbf{n}]c_{\tilde{p}\tilde{p}}[\mathbf{n} - \mathbf{m}], \quad i = C', M'. \quad (19)$$

From (12b), (16), (17), and (18), the change in our dual metric $\Delta\mathcal{E}$ can be written as

$$\begin{aligned} \Delta\mathcal{E} = & 2(\alpha + \beta)c_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2a_{C'}^0(\alpha c_{\tilde{p}\tilde{e}}^{C'}[\mathbf{m}_0] - \beta c_{\tilde{p}\tilde{e}}^{M'}[\mathbf{m}_0]) \\ & + 2a_{C'}^1(\alpha c_{\tilde{p}\tilde{e}}^{C'}[\mathbf{m}_1] - \beta c_{\tilde{p}\tilde{e}}^{M'}[\mathbf{m}_1]) \\ & - 2(\alpha + \beta)c_{\tilde{p}\tilde{p}}[\mathbf{m}_1 - \mathbf{m}_0]. \end{aligned} \quad (20)$$

The cost of computing $\Delta\mathcal{E}$ can be greatly reduced by keeping LUTs for $c_{\tilde{p}\tilde{p}}$ and $c_{\tilde{p}\tilde{e}}^i$, $i = C', M'$ as in [12] [13].

3. Experimental Results

We compare color halftones from the proposed algorithm *CMYK*-DBS, plane-independent DBS, and the HP 970 Cx inkjet printer driver, which is based on error diffusion. We use equal weights for C and M , *i.e.* $\alpha = \beta = 1$, and employ Näsänen's model [14] to compute $\tilde{p}(\mathbf{x})$ as in [13]. For plane-independent DBS, we apply monochrome DBS to each *CMY* plane independently, and use the same $\tilde{p}(\mathbf{x})$ for all three color planes as in the proposed algorithm. The HP 970 Cx inkjet printer is used in binary-only mode for this experiment.

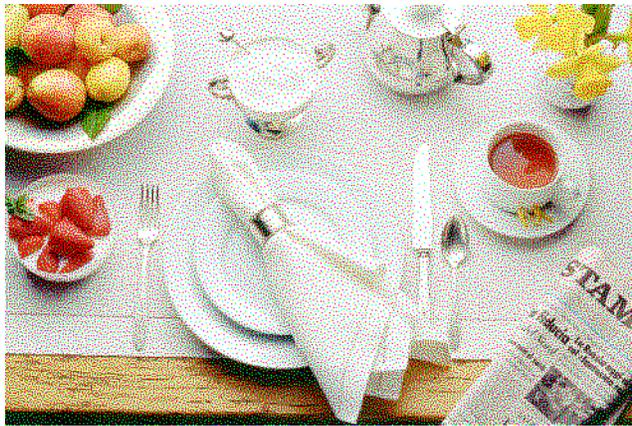
Figure 4 shows halftone images for the ‘‘Dining table’’ image. Due to excessive color interference, the result of plane-independent DBS (a) is overall much grainier than that of the HP 970 Cx driver (b) or *CMYK*-DBS (c). It also suffers from mottle. Comparing the halftones generated by the HP 970 Cx driver and *CMYK*-DBS, we see that they are much closer to each other in appearance than to the halftone from plane-independent DBS. However, the HP 970 Cx driver halftone is less homogeneous. *CMYK*-DBS yields smoother (apples and pears), more homogeneous (plates and table cloth), and less visible (coffee and flowers) textures than the other two methods.

4. Summary and Conclusions

In this paper, we introduced a new color halftoning algorithm based on the *CMY* colorant space. In order to achieve high quality color halftones, we first formulated color design criteria for uniform color textures, and then developed a colorant-based color halftoning technique which meets these criteria. We also presented a technique for mapping the color design criteria, defined in terms of constant-tone patches to real images. The basic ideal behind our halftoning method is to set the total dot arrangement first, and then to color the dots optimally without altering the total dot arrangement. This corresponds to a constrained minimization problem which we solved via the swap-only direct binary search (DBS) heuristic. We showed that the algorithm can be efficiently implemented by recursively evaluating the effect of trial changes. Since the algorithm operates in the printer colorant space, it controls the quality of each colorant texture directly. At the same time, the algorithm maintains the uniformity of the total dot distribution and minimizes dot-on-dot printing under constraints defined by the criteria. To account for the fact that the *CMY* colorants are not equally dark, we jointly halftoned the C and M planes, which have relatively high visibility. We halftoned the Y plane independently since it has relatively low visibility. The proposed algorithm yields visually smooth and fine halftone textures.



(a)



(b)



(c)

Figure 4: Halftones for the dining table image: (a) generated by plane-independent DBS, (b) generated by the HP 970 Cx inkjet printer driver, and (c) generated by CMYK-DBS. All images are printed at 150 dpi.

Acknowledgment

The authors would like to thank the Hewlett-Packard Company for their support of this research.

References

- [1] Q. Lin and J.P. Allebach, "Color FM screen design using DBS algorithm," in *Color Imaging: Device-independent Color, Color Hardcopy, and Graphic Arts III*. Proc. SPIE 3300, 1998, pp. 353 – 361.
- [2] H. Haneishi, H. Yaguichi, and Y. Miyake, "A new method of color reproduction in digital halftone image," in *Proc. IS&T 46th Annual Conf., Final Program and Advance Printing of Paper Summaries*, Cambridge, MA, May 1993.
- [3] R.V. Klassen, R. Eschbach, and K. Bharat, "Vector error diffusion in a distorted color space," in *Proc. IS&T's 47th Annual Conf./ICPS*, Rochester, NY, May 1994.
- [4] J. Shu, "Error diffusion with vivid color enhancement and noise reduction," in *Human Vision and Electronic Imaging*. Proc. SPIE 2657, 1996, pp. 464 – 470.
- [5] D. Shaked, N. Arad, A. Fitzhugh, and I. Sobel, "Ink relocation for color halftones," in *Proc. IS&T PICS*, 1998, pp. 340 – 343.
- [6] D.L. Lau, A.M. Khan, and G.R. Arce, "Stochastic moiré," in *Proc. IS&T PICS*, 2001, pp. 96 – 100.
- [7] T.J. Flohr, B.W. Kolpatzik, R. Balasubramanian, D.A. Carrara, C.A. Bouman, and J.P. Allebach, "Model based color image quantization," in *Human Vision, Visual Processing, and Digital Display IV*. Proc. SPIE 1913, 1993, pp. 270 – 281.
- [8] A.U. Agar and J.P. Allebach, "Model-based color halftoning using direct binary search," in *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V*. Proc. SPIE 3963, 2000, pp. 521 – 535.
- [9] M. Rodriguez, "A graphic arts perspective on rgb-to-cmyk conversion," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 1995, vol. II, pp. 319 – 322.
- [10] T.N. Pappas, "Model-based halftoning of color images," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1014 – 1024, July. 1997.
- [11] V. Ostromoukhov, P. Emmel, N. Rudaz, I. Amidror, and R.D. Hersch, "Multi-level colour halftoning algorithms," in *Imaging Sciences and Display Technologies*. Proc. SPIE Vol. 2949, 1996, pp. 332 – 340.
- [12] M. Analoui and J.P. Allebach, "Model based halftoning using direct binary search," in *Human Vision, Visual Processing, and Digital Display III*. Proc. SPIE 1666, 1992, pp. 96 – 108.
- [13] D.J. Lieberman and J.P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1950 – 1963, Nov. 2000.
- [14] R. Näsänen, "Visibility of halftone dot textures," *IEEE Trans. Syst., Man, Cyber.*, vol. 14, no. 6, pp. 920 – 924, 1984.