https://doi.org/10.2352/issn.2169-2629.2019.27.11

Joint design of plane-dependent FM screens sets using DBS algorithm[1]

Yi Yang, Jan P. Allebach
Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

Abstract

Color Halftoning is a technique for generating a halftone image by using a limited number of colorants to simulate a continuous-tone image as perceived by a human viewer.

This paper describes an algorithm to jointly design three screens for Cyan, Magenta and Yellow colorants using the Direct Binary Search algorithm. The results show that high-quality color halftone images can be obtained using the screens sets, and the computational complexity will be greatly reduced.

Introduction:

Digital halftoning refers to the process of transforming a continuous-tone image that contains a full range of values between 0 and 255, for example, to a representation that is either binary, 0 or 255, or that contains only a small number of levels between 0 and 255. For simplicity, we will only consider binary output devices in this paper. Digital halftoning is an especially critical step in the imaging pipeline for most printing technologies. In this case, the print mechanism places colorant dots, for example, Cyan (C), Magenta (M), or Yellow (Y) for three-color printers, on the media and the perceived colors are the result of subtractive interaction between these colorants.

Digital halftoning algorithms can be broadly classified into one of three groups [1], according to their computational complexity. In the first group, known as *screening* [2], for each colorant plane, the binary value in the halftone pattern at each pixel is determined by comparing the colorant amount at that pixel with a spatially varying threshold. The matrix of these thresholds is called the screen. Thus, the halftone value at each pixel only depends on the colorant value at that pixel location. This approach is very widely used in printers due to its computational simplicity. At the next level of complexity, we have methods in the second group, which are based on *error diffusion* [3]. Here, the binary value of each colorant is determined by thresholding a colorant amount that has been modified by the diffusion of errors made at previously binarized pixels. Thus, the halftone value at a given pixel depends on the values of the continuous-tone input in a causal neighborhood of the current pixel. Despite its increased computational complexity, error diffusion is also widely used in printers. Finally, the halftoning methods in the third group are *search-based methods* [1], [4]. These methods make several passes through the image to minimize a metric of the perceived error between the continuous-tone and halftone images. While such methods can arguably yield the best overall image quality, their computational complexity has thus far prevented their use in printers to directly halftone images. However, these methods have been frequently used as the basis for design of the parameters for simpler methods, such as screening or error diffusion that are used in printers. The Direct Binary Search or DBS algorithm is perhaps the best-known of these methods [4].

In this paper, we shall only consider halftoning by screening. In particular, for color printers, we assume that the printer independently processes each colorant plane by performing comparisons pixel-by-pixel with a screen designed specifically for that colorant plane. This architecture is a limitation imposed by many printers, ranging from low-end devices to very high-end digital presses used for commercial printing. Although the work reported in this paper is only targeted to this class of printers, it is important to note that there are other variants in this class that appear promising. In particular, the recently proposed PARAWACS algorithm explicitly controls the placement of Neugebauer primaries on a pixel-by-pixel basis [5], [6], via a single *selection matrix that is analogous to a screen*. However, this method fails the test of independently processing each colorant plane to determine the halftone output at each pixel.

In addition, we only consider 3-color printers that can place a dot of one or more of the three colorants C, M, or Y at each pixel. Such devices are also commercially important. At the low-end, this restriction lowers the cost of the printer mechanism. At the high-end, the use of ink can be reduced, and the throughput increased, by not using the black (K) colorant, even when it is available. This reduces the overall cost of a print job.

Since the design of the screens is an offline process, we are free to use extensive computational resources in the design process. Here, we use DBS to jointly design a set of three screens that when used to halftone color images will yield good quality, which exceeds that resulting when the same color images are halftoned using screens that have each been independently designed using DBS.

Joint screen design using DBS was first proposed in Refs.[7] and [8]. The authors introduced a cost metric that was based on a full spatiochromatic model for the human visual system [14]. The metric was quite general in that at any level, it could account for quality of the halftone patterns consisting of any desired subset of the constituent Neugebauer primaries comprising the composite halftone pattern. Then, Refs.[10] and [11] proposed a different strategy for color halftoning based on DBS. Here, the cost function was simply based on Nasenen's

[1] Research supported by HP Inc., Barcelona, SPAIN

monochrome human visual system response [12], [13]; and the interaction of colorants was controlled by constraints that varied based on the target color value. However, these two references did not propose a screen design method, and only considered joint design of the C and M halftone patterns.

In the present paper, we propose a method for joint design of CMY color screen sets based on DBS that is philosophically most similar to the approach taken in Refs.[10] and [11]. In addition to proposing a detailed process for screen design, which is not given by Refs.[10] and [11], the present paper explicitly considers all three colorants C, M, and Y in the design process, and also more completely controls the printing of secondary colors and composite black (CMY) across the tone scale than do Refs.[10] and [11].

Joint Design Algorithm

To design a screen level-by-level, we select a certain level as the starting level, generate an initial pattern, and then apply our DBS-based optimization algorithm to obtain a homogeneous halftone pattern for this level. Then, based on the starting level, we design lighter levels by gradually removing dots and optimizing patterns, and design darker levels by gradually adding dots and optimizing patterns.

In this paper, the patterns for each level need to satisfy the following conditions [10], [11] for obtaining homogeneous halftone images.

(1). If we consider dots of each colorant (C, M, or Y) individually, we would like these dots to be arranged as uniformly as possible.

(2). We also would like the overall composite pattern that consists of all three colorants dots to be as uniform as possible.

(3). We need to reduce the dot-on-dot (overlapping dots) as much as possible. That is, keep dot-off-dot as much as possible during the process of design screens.

1. Optimization Procedure for Designing the Initial Level

We assume that the halftone screens have size $M \times M$, and the total number of levels of the screens is L. For all the examples shown in this paper, the size of the screens is 256×256, and the number of levels is $L = 256$. To simplify the notation, and without loss of generality, we will assume that the number of levels L is divisible by 3. If this requirement is not met, the procedure described below can be executed with minor modifications to the notation.

We consider a CMY color pattern as a combination of three different C, M and Y patterns. To meet the conditions in the preceding paragraph, we propose our algorithm with the following steps:

First, select a starting level. We choose $\frac{1}{3}L$ as the starting level in this paper since we consider three colorants: C, M and Y. The size of the screen determine the total number of dots for the three colorants (C+M+Y). With starting level $\frac{1}{3}L$, each colorant pattern receives $\frac{256^2}{3}$ dots for a total number of 256^2 dots. We generate an initial pattern based on the total number of dots (all C, M and Y dots are treated as black dots, and others are treated as white holes). Then, we apply the monochrome DBS algorithm to this pattern to achieve a uniform pattern. However, if the starting level is $\frac{1}{3}L$, every pixel in the initial halftone pattern will have been assigned a colorant dot, either C, M, or Y. Thus, there will be no white holes; and this first stage of optimization is not needed. It will only be required if the initial level is $< \frac{1}{3}L$.

Secondly, we randomly and evenly divide the positions of black dots in the uniform pattern into three same-size patterns representing the C pattern, M pattern and Y pattern, respectively, guaranteeing that the three patterns each have the same number of dots. By this step, we can prevent dot overlap. Especially in lighter levels, this step enforces the three patterns to be totally dot-off-dot.

Then, we apply the swap-only DBS algorithm to iteratively compute the joint $\Delta\varepsilon$ between two different patterns, adjusting the position of dots to minimize the error. Since every pixel is assigned one of the three colorants C, M, and Y, the only option is to swap one colorant pixel with another pixel of a different colorant. We start by scanning the C pattern until we find a C dot, then we scan the M pattern until we find an M dot. We swap these two pixels(exchange the position indices of these two pixels), and evaluate the change in error. We repeat this process until we have considered swapping every M pixel with this C pixel. If one or more of these trial swaps decreased the error, we keep the one that decreased the error the most. Otherwise, we don't accept any swaps with this C pixel. Then we go on to the next Cyan pixel, and repeat the process. After we have scanned through the entire halftone pattern, we repeat the process, and scan the halftone pattern again. When no change is accepted during an entire scan of the halftone pattern, this phase of the optimization has converged.

Next, we consider swaps between C pixels and Y pixels, following the procedure described in the preceding paragraph. Finally, we consider swaps between M pixels and Y pixels, again following the procedure described in the preceding paragraph. Once we have completed these three sequential optimization processes, we repeat the entire cycle: C with M, C with Y, and M with Y. When no changes are kept during an entire cycle, the overall optimization of the starting level of the three halftone patterns is complete.

In summary, the first step ensures that the screen superimposed by three colorants is uniform, and the next steps ensure that each of the three screens is uniform. Thus, patterns that satisfy those conditions can be obtained.

To illustrate the pattern uniformity, we presented initial and optimized pattern of C and M in *Fig. 1* and *Fig. 2*, initial and optimized combination of C and M patterns in *Fig. 3*, a small part of the initial and optimized combination of C, M and Y patterns in *Fig 4*, and the entire optimized pattern in *Fig. 5*, which is the uniform starting pattern.

2. Error Metrics

The Human Visual System (HVS) can be modeled as a linear, shift-invariant system with a point spread function given by

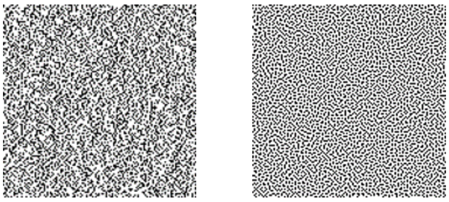


Figure 1. *The initial and optimized pattern of Cyan.*

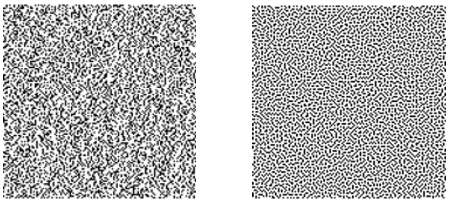


Figure 2. *The initial and optimized pattern of Magenta.*

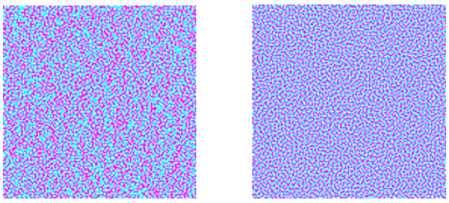


Figure 3. *The initial and optimized pattern of combination of Cyan and Magenta.*

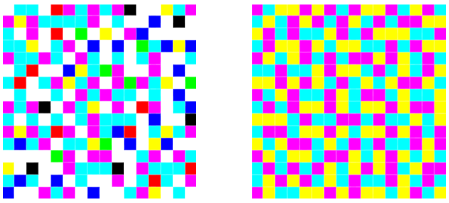


Figure 4. *The zoomed-in details (16 × 16) of a random pattern and our optimized pattern superimposed by Cyan, Magenta, and Yellow dots.*

the inverse Fourier transform of the contrast sensitivity function of the human viewer [14]. Kim and Allebach compared Nasanen's model with three other HVS models and concluded that Nasanen's model gave the best overall halftone quality when incorporated in the DBS algorithm [12]. In this paper, we used the contrast sensitivity function based on Nasanen's model [13] to filter the continuous tone image and halftone image.

We use $(\mathbf{x}) = (x,y)$ and $[\mathbf{m}] = [m,n]$ to denote continuous and discrete spatial coordinates, respectively. Then, we let $f_i(\mathbf{x})$ denote the continuous-tone image, and $g_i[\mathbf{m}]$ denote the halftone images of $f_i(\mathbf{x})$, where, *i = C, M, Y* denotes the colorant of the Cyan, Magenta, and Yellow components. We assume $f_i(\mathbf{x})$ takes values between 0 and 1 $(0, \frac{1}{L-1}, \frac{2}{L-1}, ..., 1)$, where L indicates the total number of grey levels mentioned previously. The digital halftone image $g_i[\mathbf{m}]$ takes binary values **0** or **1**.

We consider the patterns of Cyan, Magenta and Yellow to be monochrome, and the contrast sensitivity function based on Näsänen's model is used as the HVS filter. For an ideal printer which produces a square dot, the perceived rendered image $\tilde{g}_i(\mathbf{x})$ is

$$\tilde{g}_i(\mathbf{x}) = \sum_{m} g_i[\mathbf{m}]\tilde{p}(\mathbf{x}\text{-}\mathbf{mX}) \tag{1}$$

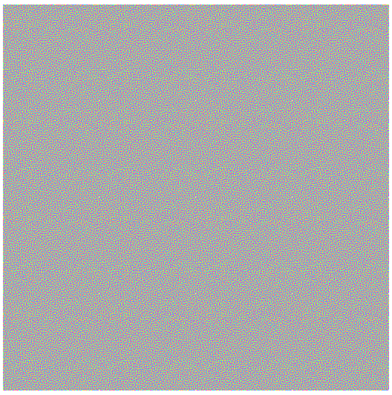


Figure 5. *The combination of optimized patterns of Cyan, Magenta and Yellow for the level $\frac{1}{3}L$. The reader is advised to zoom in to this pattern to see the individual colorant dots.*

where **X** is a diagonal matrix that represents the distance between printer addressable pixels on a rectangular grid, and $\widetilde{p(\mathbf{x})} = p_{hvs}(\mathbf{x}) * p_{dots}(\mathbf{x})$ indicates the effects of cascading the printer rendering and HVS models. So the perceived error is the filtered $e[\mathbf{m}]$, which can be written as

$$\tilde{e}(\mathbf{x}) = \sum_{\mathbf{m}} e[\mathbf{m}]\tilde{p}(\mathbf{x}\text{-}\mathbf{mX}) \tag{2}$$

where $e[\mathbf{m}] = g[\mathbf{m}] - f[\mathbf{m}]$ is the error image.

In a monochrome halftone image, when we swap two pixels at indices $\mathbf{m_0}$ and $\mathbf{m_1}$, we can write the change in error due to the swap as:

$$\begin{aligned}\Delta\varepsilon =& (a_0^2 + a_1^2)c_{\tilde{p}\tilde{p}}[\mathbf{0}] \\ &+ 2(a_0 c_{\tilde{p}\tilde{e}}[\mathbf{m_0}] + a_1 c_{\tilde{p}\tilde{e}}[\mathbf{m_1}]) \\ &- 2(a_0 a_1)c_{\tilde{p}\tilde{p}}[\mathbf{m_1} - \mathbf{m_0}]\end{aligned} \tag{3}$$

where

$$c_{\tilde{p}\tilde{e}}(\mathbf{x}) = \int \tilde{p}(\mathbf{y})\tilde{e}(\mathbf{y}\text{+}\mathbf{x})d\mathbf{y} \tag{4}$$

$$c_{\tilde{p}\tilde{p}}(\mathbf{x}) = \int \tilde{p}(\mathbf{y})\tilde{p}(\mathbf{y}\text{+}\mathbf{x})d\mathbf{y} \tag{5}$$

and $c_{\tilde{p}\tilde{e}}[\mathbf{m}] = c_{\tilde{p}\tilde{e}}(\mathbf{mX})$, $c_{\tilde{p}\tilde{p}}[\mathbf{m}] = c_{\tilde{p}\tilde{p}}(\mathbf{mX})$.

When $g[\mathbf{m}]$ is changed from 1 to 0, $a_0 = -1$, and when $g[\mathbf{m}]$ is changed from 0 to 1, $a_0 = 1$. The change rule of a_1 is the reverse of that for a_0, so $a_1 = -a_0$.

When we apply a swap-operation between a pixel that is Cyan and a pixel that is Magenta, we consider the dual metric,

$$\Delta\varepsilon = \alpha\Delta\varepsilon_{Cyan} + \beta\Delta\varepsilon_{Magenta} \tag{6}$$

We need to ensure that the patterns for the different colorants have the same quality, so we use equal weights:

$$\alpha = \beta = 1$$

Assume that we have an acceptable swap trial when we exchange a Cyan dot $(g_C[\mathbf{m}_0])$ with a Magenta dot $(g_M[\mathbf{m}_1])$. This swap indicates $g_C[\mathbf{m}_0]$ will be changed from 1 to 0, and $g_C[\mathbf{m}_1]$

will be changed from 0 to 1. Meanwhile, $g_M[\mathbf{m}_0]$ will be changed from 0 to 1 and $g_M[\mathbf{m}_1]$ will be changed from 1 to 0.

The pair of swapping dots need to have different binary states. So,

$$a_0^{M[\mathbf{m}_0]} = a_0^{C[\mathbf{m}_1]} = a_1^{M[\mathbf{m}_1]} = a_1^{C[\mathbf{m}_0]} = 1$$

$$a_0^{M[\mathbf{m}_1]} = a_0^{C[\mathbf{m}_0]} = a_1^{M[\mathbf{m}_0]} = a_1^{C[\mathbf{m}_1]} = -1$$

More generally, if we are considering swaps between colorant i and colorant j, where $i, j \subset C, M, Y$, then the parameters can be written as,

$$a_0^{i[\mathbf{m}_0]} = a_0^{j[\mathbf{m}_1]} = a_1^{i[\mathbf{m}_1]} = a_1^{j[\mathbf{m}_0]} = 1$$

$$a_0^{i[\mathbf{m}_1]} = a_0^{j[\mathbf{m}_0]} = a_1^{i[\mathbf{m}_0]} = a_1^{j[\mathbf{m}_1]} = -1$$

where $i, j \subset C, M, Y$

Substituting these relationships into Eq. (6), we can get the swap-only dual metric of DBS:

$$\begin{aligned}\Delta\varepsilon =& 2(\alpha+\beta)c_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2(\alpha+\beta)c_{\tilde{p}\tilde{p}}[\mathbf{m_1}-\mathbf{m_0}] \\ &+ 2a_0^{j[\mathbf{m}_0]}(\alpha c^{j}_{\tilde{p}\tilde{e}}[\mathbf{m_0}] - \beta c^{i}_{\tilde{p}\tilde{e}}[\mathbf{m_0}]) \\ &+ 2a_0^{j[\mathbf{m}_1]}(\alpha c^{j}_{\tilde{p}\tilde{e}}[\mathbf{m_1}] - \beta c^{i}_{\tilde{p}\tilde{e}}[\mathbf{m_1}]) \\ =& 4c_{\tilde{p}\tilde{p}}[\mathbf{0}] + 4c_{\tilde{p}\tilde{p}}[\mathbf{m_1}-\mathbf{m_0}] \\ &+ 2a_0^{j[\mathbf{m}_0]}(c^{j}_{\tilde{p}\tilde{e}}[\mathbf{m_0}] - c^{i}_{\tilde{p}\tilde{e}}[\mathbf{m_0}]) \\ &+ 2a_0^{j[\mathbf{m}_1]}(c^{j}_{\tilde{p}\tilde{e}}[\mathbf{m_1}] - c^{i}_{\tilde{p}\tilde{e}}[\mathbf{m_1}])\end{aligned} \tag{7}$$

3. Design of Levels Below and Above the Initial Level

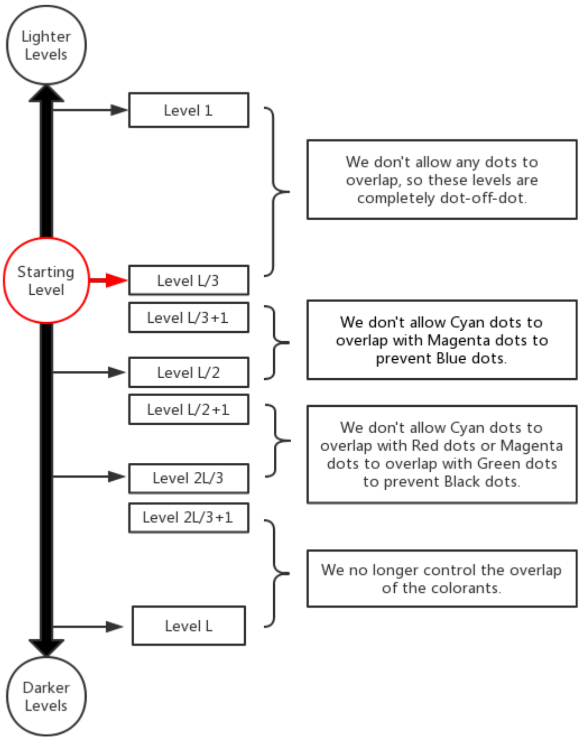

Lighter Levels
Level 1
We don't allow any dots to overlap, so these levels are completely dot-off-dot.
Starting Level
Level L/3
Level L/3+1
We don't allow Cyan dots to overlap with Magenta dots to prevent Blue dots.
Level L/2
Level L/2+1
We don't allow Cyan dots to overlap with Red dots or Magenta dots to overlap with Green dots to prevent Black dots.
Level 2L/3
Level 2L/3+1
We no longer control the overlap of the colorants.
Level L
Darker Levels

Figure 6. Design order and colorants control.

The process of generating halftone patterns for all levels is not the same. However, all L levels will depend on the uniform starting pattern for $\frac{1}{3}L$ that we designed in the preceding section. The color pattern at this level is full dot-off-dot without any holes or overlaps. Then, we generate lighter levels by gradually removing the dots and generate darker levels by gradually adding dots. The number of dots added or deleted per level is equal for all levels and each colorant. Since we need to keep the stacking constraint, only the newly added or deleted dots can be moved, as we design each succeeding level.

To design the first lighter level, we first choose at random M^2/L colorant dots to remove from each of the three starting patterns, which takes us to the next lighter level $\frac{1}{3}L-1$. Then, we have a composite halftone pattern consisting of an equal number $M^2(1-\frac{1}{L})$ of C, M, and Y pixels, plus $3M^2/L$ White pixels. At this point, to satisfy the stacking constraint, we may only consider swaps between a given colorant pixel and a White pixel that was previously of the same colorant value as the colorant pixel to be swapped. As before, we scan through all C pixels, considering possible swaps of each C pixel with every White pixel that was previously Cyan. For each C pixel, we retain that swap, if any, that most reduces the error. We repeat this process, scanning through all the C pixels until no swap is retained during an entire iteration through the halftone pattern. We then conduct the same process considering swaps of M pixels with White pixels that were previously Magenta, and then again, considering swaps of Y pixels with White pixels that were previously Yellow. Once this cycle of three sequential optimization processes is complete, we repeat it until no changes are accepted during an entire cycle. Then, the optimization of the first lighter level ($\frac{1}{3}L-1$) is complete.

To design the second lighter level ($\frac{1}{3}L-2$), we again choose at random M^2/L colorant dots to remove from each of the three halftone patterns. At this point, to again satisfy the stacking constraint, we can only swap a colorant pixel with a White pixel that had that colorant value at the previous level. Pixels that were White at the previous level cannot be changed. Once again, we repeatedly go through the cycle of three sequential optimizations until no change is accepted.

Then, we continue designing the next lighter levels $\frac{1}{3}L-3, \frac{1}{3}L-4, \ldots, 1, 0$ by repeating this process until there are no colorant dots to be removed. At this point, we have completed the design of all the levels lighter than the starting level $\frac{1}{3}L$. For all levels that lighter than $\frac{1}{3}L$, we can guarantee patterns are completely dot-off-dot, and also, the three colorants dots are uniformly distributed in the halftone patterns at each level, as illustrated in Fig. 7 for the Magenta colorant.

From the levels greater than $\frac{1}{3}L$, secondary colors will appear due to overlaps between colorants. Our algorithm directly operates only on the Cyan, Magenta and Yellow colorants; so we need to control the secondary colors such as Blue, Green, and Red in our processing.

We know that for the printer color space:

Cyan + Magenta = Blue
Cyan + Yellow = Green

Figure 7. The uniform patterns for Magenta colorant in the 1st, 2nd, and 3rd levels.

Magenta + Yellow = Red

We hypothesize that the secondary colors Red and Green blend well with the primaries that contribute most strongly to their lightness and hue [10], [11]. That is, Red harmonizes well with Magenta, and Green harmonizes well with Cyan. However, Blue dots are more problematic than Red and Green dots. Fig. 8 illustrates this phenomenon, although it is important to note that here the colorants and colors are generated by additive combinations of Red, Green and Blue primaries, not subtractive combinations of actual printed colorants Cyan, Magenta and Yellow. Therefore, when we design levels darker than $\frac{1}{3}L$, we prefer to let the Yellow dots first overlap with the other two colorant dots, but we don't allow Cyan dots to overlap with Magenta dots to prevent the Blue dots.

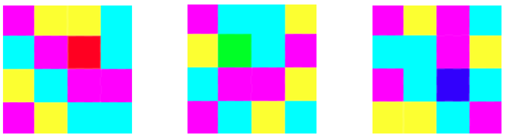


Figure 8. The secondary colors Red, Green and Blue with the primary colors Cyan, Magenta and Yellow.

To design the level $\frac{1}{3}L+1$, we start by adding, at random, M^2/L colorant dots to each of the three starting patterns. Secondary colors appear when we add new colorant dots. However, until we pass the level $\frac{1}{2}L$, we can prevent the occurrence of Blue dots, by not allowing Cyan and Magenta dots to be placed at the same pixel location. Thus every new Cyan dot or new Magenta dot will have to be placed on a Yellow dot, resulting in either a Green or a Red pixel, respectively. And each new Yellow dot must be placed on either a Cyan or Magenta dot. After the initial random assignment is made, we follow the same three-stage sequential optimization described in the preceding paragraphs, but only swap Green and Yellow pixels or Red and Yellow pixels. When this process is completed, we have the optimized design of the three halftone patterns for level $\frac{1}{3}L+1$.

To design the halftone patterns for level $\frac{1}{3}L+2$, we again add, at random M^2/L colorant dots to each of the three halftone patterns for level $\frac{1}{3}L+1$, following the same constraints as we did for level $\frac{1}{3}L+1$. However, during each of the three stages of the optimization process, we may only swap dots that were newly added at the beginning of the design procedure for this level. Again, we do not allow Cyan and Magenta dots to be placed at the same location.

Once we reach level $\frac{1}{2}L$, half of the M^2 pixels in the composite halftone pattern contain Cyan dots; and the other half of the pixels contain Magenta dots. Of these pixels, $\frac{1}{4}$ will be Green, $\frac{1}{4}$ will be Red, $\frac{1}{4}$ will be Cyan, and $\frac{1}{4}$ will be Magenta. Thus, when we design the halftone patterns for levels $\frac{1}{2}L+1$ and above, we have to allow the occurrence of Blue dots. In fact, for level $\frac{1}{2}L+1$ each new Cyan dot can be placed on an existing Magenta pixel and each new Magenta dot can be placed on an existing Cyan dot. With this constraint, we can avoid the occurrence of Black pixels, which would result if a new Cyan dot is placed on a Red pixel, or if a new Magenta dot is placed on a Green pixel. We can avoid having any Black pixels until we reach the level $\frac{2}{3}L$. At this point, every pixel will be either Red, Green, or Blue.

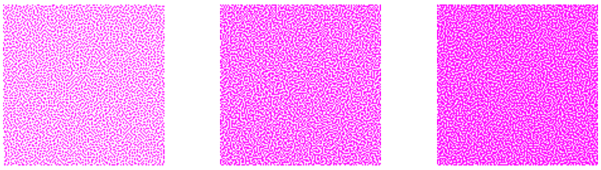


Figure 9. The uniform patterns for the Magenta colorant in the 85th, 130th and 160th level uniform patterns for Magenta colorant. Here the first Magenta pattern is designed under the constraint that there are no Blue dots in the composite halftone pattern; the second and third Magenta patterns are designed under the constraint that there are no Black dots in the composite halftone pattern. When viewing these patterns, the reader is advised to zoom in until the individual dots become visible. Otherwise, there may be sampling artifacts.

Results

In this section, we show sample halftone images obtained using a CMY screen set that has been jointly designed following the procedures described in previously in this paper. Each screen contain $M \times M = 256 \times 256$ pixels and $L = 256$ levels. The halftoning is implemented in CMY color space. We apply gamma de-correction to each channel of a color image, convert RGB to CMY components by using the formula:

C = 255 - R
M = 255 - G
Y = 255 - B

and use the halftone rule below:

If $\mathrm{C_{image}}[\mathrm{i,j}] < \mathrm{C_{screen}}[\mathrm{i,j}]$, print a Cyan dot at $[\mathrm{i,j}]$;
If $\mathrm{M_{image}}[\mathrm{i,j}] < \mathrm{M_{screen}}[\mathrm{i,j}]$, print a Magenta dot at $[\mathrm{i,j}]$;
If $\mathrm{Y_{image}}[\mathrm{i,j}] < \mathrm{Y_{screen}}[\mathrm{i,j}]$, print a Yellow dot at $[\mathrm{i,j}]$;

Using the set of Cyan, Magenta and Yellow screens, the color halftone processing can be simplified to a channel-independent comparison operation on a pixel-by-pixel basis.

We present a continuous-tone image in *Fig. 10* and the halftone image obtained using our jointly designed screen set in *Fig. 11* to illustrate the halftone quality. Also, we compare our result with a halftone image obtained using independently designed screens in *Fig. 12*. To better compare the uniformity of the halftone images, we zoom in on the same parts of the two halftoned images. From the detailed comparison in *Fig. 13*, it is clear that the color image halftoned by the jointly designed C, M and Y screen set greatly reduces the occurrence of secondary colors and white holes. Hence, it appears less grainy.

Summary

This paper describes an algorithm to jointly design screen sets for Cyan, Magenta and Yellow colorants using the DBS algorithm. The results show that the color images halftoned using the jointly designed screen set are more uniform than those halftoned using independently designed screens, due to the reduction in secondary colors and white holes. Also, compared with halftoning a color image directly using DBS [11] , jointly designed screens sets can greatly reduce the computational complexity.

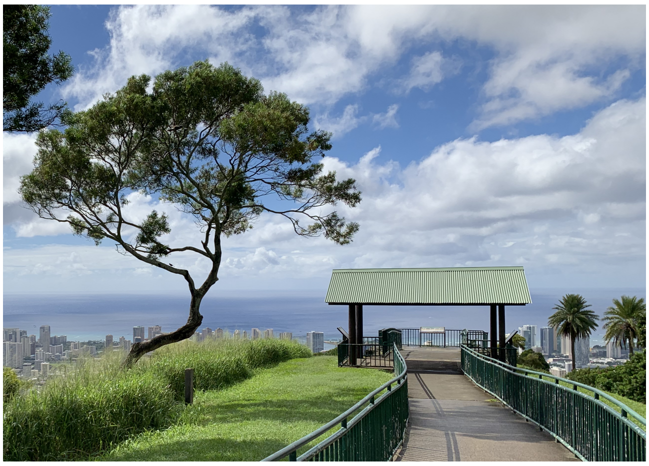


Figure 10. *Continuous-tone color image*

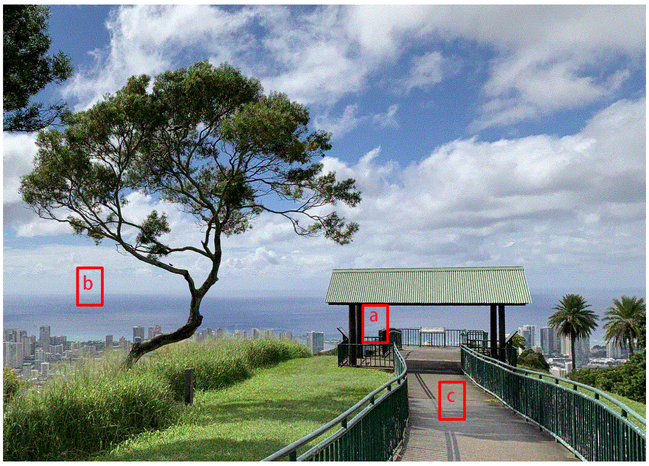


Figure 11. *Image halftoned using jointly designed screen set.*

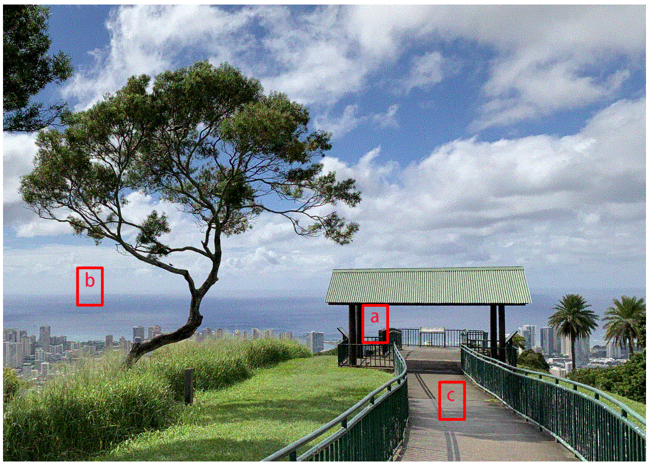


Figure 12. *Image halftoned using three independently designed screens.*

References

[1] Jan P. Allebach, “DBS: retrospective and future directions”. *Color Imaging: Device Independent Color, Color Hardcopy, and Graphic Arts.* Vol. SPIE-4300, 2001.

[2] Jan P. Allebach and R. N. Stradling, “Computer-aided design of dither signals for binary display of images”. *Applied Optics.* Vol. 18, 1979.

[3] Robert W. Floyd and Louis S. Steinberg, “An adaptive algorithm for spatial grayscale”. *Proceedings of the Society of Information Display.* Vol. 17, 1976.

[4] David J. Lieberman and Jan P. Allebach, “A dual interpretation for direct binary search and its implications for tone reproduction and texture quality”. *IEEE Transactions on Image Processing.* Vol. 9, 2000.

[5] Peter Morovic, Jan Morovic, Jay Gondek and Robert Ulichney, “ Direct Pattern Control Halftoning of Neugebauer Primaries”. *IEEE Transactions on Image Processing*. Vol. 26, 2017.

[6] Peter Morovic, Jan Morovic, Jay Gondek and Robert Ulichney, “ PARAWACS: color halftoning with a single selector matrix”. *Color and Imaging Conference*. pp. 41-46 2016.

[7] Qian Lin and Jan P. Allebach, “Color FM screen design using DBS algorithm”. *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts.* Vol.SPIE-3300, 1998.

[8] Jan P. Allebach and Qian Lin, “FM screen design using DBS algorithm”. *Proceedings of 3rd IEEE International Conference on Image Processing.* Vol. 1, 1996.

[9] Thomas J. Flohr, Bernd W. Kolpatzik, Raja Balasubramanian, David A. Carrara, Charles A. Bouman and Jan P. Allebach, “Model-based color image quantization”. *Proceedings of the 1993 IST/SPIE International Symposium on Electronic Imaging Science and Technology.* Vol. 1913, 1993.

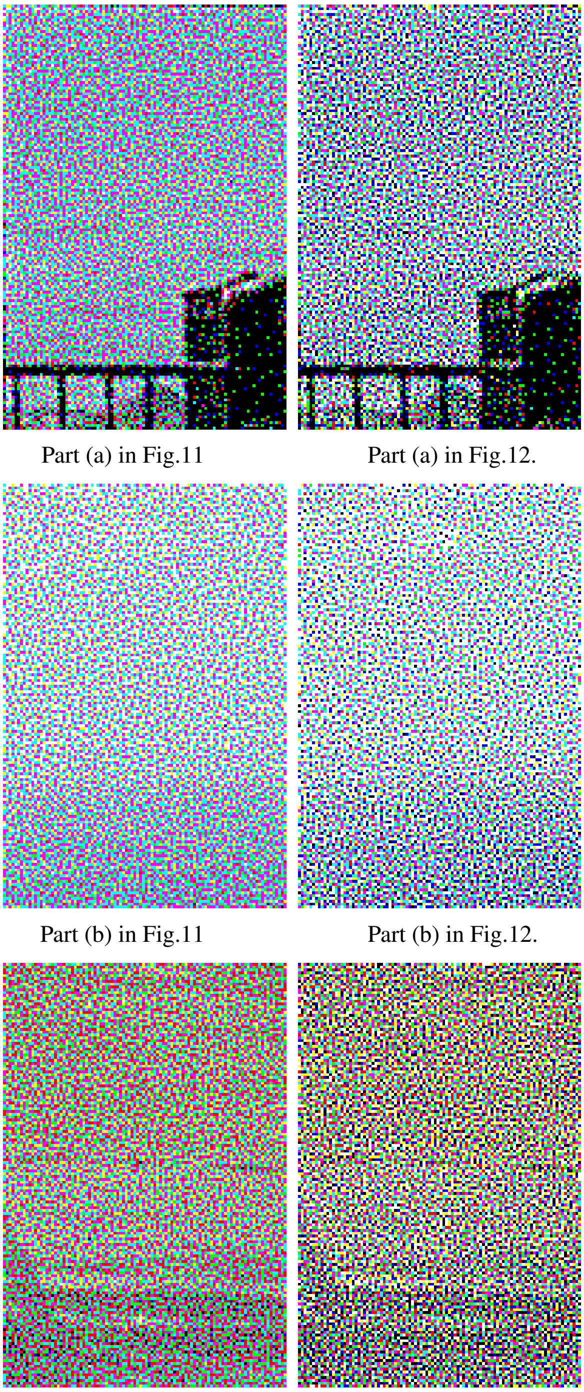
[10] Je H. Lee and Jan P. Allebach, “Colorant-based direct binary search halftoning”. *Journal of Electronic Imaging.* Vol. 11, 2002.

[11] Je H. Lee and Jan P. Allebach, “CMYK Halftoning Algorithm Based on Direct Binary Search”. *Color and Imaging Conference.* Vol. 6, 2001.

[12] Sang H. Kim and Jan P. Allebach, “Impact of HVS models on model-based halftoning”. *IEEE Transactions on Image Processing.* Vol. 11, 2002.

[13] Risto Näsänen, “Visibility of halftone dot textures”. *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. 6, 1984.

[14] Brian A. Wandell, *Foundations Of Vision.* Sinauer Associates, 1995.



Part (a) in Fig.11

Part (a) in Fig.12.

Part (b) in Fig.11

Part (b) in Fig.12.

Part (c) in Fig.11

Part (c) in Fig.12.

Figure 13. *Comparison of the zoomed-in detail of three regions in the two halftone images shown in Fig. 11 and Fig. 12. (The right is halftoned by independently designed screens, and the left is halftoned by jointly designed screens).*