https://doi.org/10.2352/ISSN.2470-1173.2018.16.COLOR-429

Novel color halftoning algorithm for ink savings*

***Wanling Jiang*[1], *Alex Veis*[2], *Robert Ulichney*[3], *and Jan P. Allebach*[1]**
[1]***Electronic Imaging Systems Laboratory, School of Electrical and Computer Engineering, Purdue University,***
West Lafayette, IN-47906, U.S.A. {jiang267,allebach}@purdue.edu;
[2]***Hewlett-Packard Scitex Division, Israel. alex.veis@hp.com;***
[3]***Hewlett-Packard Laboratories USA, Cambridge, MA 02142, U.S.A. u@hp.com***

Abstract

Abstract—In this paper, we propose a novel halftoning method that we call Ink-Saving, Single-Frequency, Single-Angle, Multi-Drop (IS-SF-SA-MD) halftoning. The application target for our algorithm is high-volume production ink-jet printing in which the user will value a reduction in ink usage. Unlike commercial offset printing in which four-colorant printing is achieved by rotating a single screen to four different angles, our method uses a single frequency screen at a single angle, and depends on accurate registration between colorant planes to minimize dot-overlap especially between the black (K) colorant and the other colorants (C, M, and Y). To increase the number of graylevels for each colorant, we exploit the multidrop capabilities of the target writing system. We also use the hybrid screening method to yield improved halftone texture in the highlights and shadows.

I. Introduction

Digital halftoning is the process of converting a continuous-tone image into a binary image. When observed from the intended viewing distance, the goal of the halftoning algorithm is to provide the viewer with a percept of continuous tones ranging smoothly from highlight through mid-grays to shadows. The viewer should not perceive the binary textures that comprise the halftone image. To achieve this goal, the halftoning algorithm relies on the limited ability of the human visual system to resolve fine spatial detail.

There are three major types of halftoning methods. The most computationally expensive methods are iterative, such as Direct Binary Search (DBS)[1], where the image is visited multiple times in multiple passes. In each pass, multiple swaps and/or toggles are evaluated. For each toggle/swap, depending on whether the change can reduce the error metric or not, we accept or discard the change. The operation is repeated until the algorithm converges. There are dispersed-dot and clustered-dot DBS algorithms. Color halftoning based on Neugebauer Primary area coverage [2] is an example of a dispersed-dot color halftoning method based on DBS, and Multi-Stage, Multi-Pass, clustered-DBS [3] is an example of a clustered-dot DBS algorithm.

The second type of halftoning method also requires the information of neighboring pixels, but only one pass of the image is performed. Error diffusion is an example of this method. The image quality may not be quite as good as that of an iterative method, but it requires less computation. The quantization residual for each pixel is diffused to its neighboring pixels. An example of error diffusion is Floyd-Steinberg error diffusion [4].

The third type of halftoning method is screening. This method requires the least amount of computation. For traditional color halftoning using a screening method, the value of the continuous-tone image in each channel at every pixel is compared with the threshold at that pixel in its corresponding screen matrix. If the threshold is exceeded, the colorant for that channel will be printed at that pixel. If it is not exceeded, that colorant will not be printed there. With traditional 4-color printing, the color halftone result is the combination of the halftone results of the four colorants, cyan (C), magenta (M), Yellow (Y), and black (K).

Halftone images are categorized according to their texture. They are distinguished by the attributes of the dots comprising the halftone, which are dispersed-dot and clustered-dot. Another way to categorize the halftone is by the periodicity of the halftone texture, whether it is periodic or aperiodic. The halftone can be any one of the four combination of these two categories. In this paper, we primarily discuss clustered-dot periodic halftones. We create such a halftone by generating a halftone screen that is clustered-dot and periodic. the clustered-dot screen has an advantage over dispersed-dot screens due to the fact that it is beneficial for printers with unstable output.

The target resolution for a given printing device requires that the frequency of the screen to be high enough to have good quality at a normal viewing distance. However, the higher the frequency of the screen the smaller the number of graylevels the screen can produce. Thus there will be contouring artifacts due to the limited number of graylevels. The supercell method solves this problem by allowing more graylevels, while retaining the clusters for better print stability. The hybrid screen further optimizes the texture in highlights and shadows.

Multilevel halftoning is possible with both laser electrophotographic and inkjet printers. With the former, either laser amplitude modulation or pulse-width modulation is possible. With the latter, multiple drops can be placed at a single printer-addressable pixel.

Our work is a novel halftoning method that is targeted for inkjet. It reduces the overlap between different colorants and saves ink at the same time. The algorithm

*Research sponsored by HP Inc., Scitex Division, Netanya, Israel

incorporates a multilevel halftone method to increase the number of gray levels, and to reduce the visibility of dots in highlight regions. The traditional halftone screen is replaced by a hybrid screen to optimize the texture of the printed halftone patterns. The hybrid screen design is based on Lee and Allebach's supercell method.

The rest of the paper is organized as follows. In Section II, we discuss the major components of the ink-saving, single-frequency, single-angle, multidrop (IS-SF-SA-MD) halftoning algorithm. In Section III, we introduce the novel halftone method with ink limitation. Experimental results are presented in Section IV. And our conclusion is made in Section V.

II. Major components

In this section, we discuss the major components of our ink-saving, single-frequency, single-angle multidrop (IS-SF-SA-MD) halftoning algorithm in the order of the steps we took for the novel halftoning algorithm.

II.A. Color management

In order to perform halftoning on an sRGB image using our hybrid screen based method in CMYK color space, we need to convert the sRGB image to the CMYK color space we work in. Photoshop is used in the conversion, and U.S. Web Coated (SWOP) v2 color profile is selected. According to the specification of SWOP, the nominal screen ruling should be 133 lines per inch.

II.B. Gray component replacement (GCR)

The gray component is a mixture of three colorants: cyan (C), magenta (M), and yellow (Y). With different amounts of the CMY mixture, we have different levels of lightness. The idea of substituting some amount of black for the gray component is known as gray component replacement, it reduces the expense of printing stabilize the printing process[6]. The GCR method in the work of Davor et al. is content dependent, depending on whether the content of images contain many replaceable colors or not, ink savings due to GCR can vary[7]. Content-dependent GCR method use more aggressive GCR in detail regions where K dots are not so visible and they provide higher contrast rendering of detail. GCR can also be tone-dependent.

In our halftoning process, GCR and ink limitation are applied before the halftoning step. And we perform GCR first, before enforcing ink limits, because GCR will reduce the total number of drops, and thus move us away from the gamut boundary due to ink limits. Intuitively, we expect that due to ink limitations, GCR may allow us to print colors that we otherwise would not be able to print. The input to GCR is $I_{request}$, and output of GCR is I_{post_gcr}. GCR involves subtracting equal amounts of C, M, Y, K and transfering this to K. This can change the ratios of C, M, Y. i.e. it is possible that:

$$\frac{I_{post\_gcr}}{J_{post\_gcr}} \neq \frac{I_{request}}{J_{request}} \tag{1}$$

where $I,J = C,M,Y$ and $I \neq J$.
Regardless of the amount of GCR, if $I_{request} = J_{request}$ then $I_{post\_gcr} = J_{post\_gcr}$.

Next, we discuss the way in which the GCR is performed. Let $R^{max} = min(C_{request}, M_{request}, Y_{request}, 1 - K_{request})$ be the maximum amount of colorant to be replaced. And let $0 \leq \alpha \leq 1$ be the fractional amount that is actually to be performed. Then $R^{post\_gcr} = \alpha \cdot R^{max}$, and we have

$$I_{post\_gcr} = I_{request} - \alpha \cdot R^{max} \tag{2}$$

$$K_{post\_gcr} = K_{request} + \alpha \cdot R^{max} \tag{3}$$

where $I = C,M,Y$.
Note that we are removing R_{post_gcr} from each of C, M and Y, i.e. 3 different colorants, but we only add R_{post_gcr} to one colorant, i.e. K. This implies that n drops of K is equivalent to n drops each of C, M, and Y.

II.C. Ink limitation

After GCR, we enforce ink limits with continuous-tone image prior to halftoning. Let $C_{post\_gcr}, M_{post\_gcr}, Y_{post\_gcr}, K_{post\_gcr}$ be the ink amounts after GCR. And the summation is calculated as

$$\begin{aligned} \bar{I}^{total}_{post\_gcr} = C_{post\_gcr} + M_{post\_gcr} \\ + Y_{post\_gcr} + K_{post\_gcr} \end{aligned} \tag{4}$$

Let $C_{actual}, M_{actual}, Y_{actual}, K_{actual}$ be the actual ink amounts that are printed. If $I^{total}_{post\_gcr} \leq \frac{4}{3}$, no ink limitation is necessary, because theoretically, we can put a maximum of 4 drops at any given pixel. However, the halftoning algorithm will restrict the gamut even more. And we will discuss this in Section IV. In the case where ink limitation is not needed, we have

$$\begin{cases} C_{actual} = C_{post\_gcr}, \\ M_{actual} = M_{post\_gcr}, \\ Y_{actual} = Y_{post\_gcr}, \\ K_{actual} = K_{post\_gcr} \end{cases} \tag{5}$$

When $I^{total}_{post\_gcr} > \frac{4}{3}$ We perform ink limitation. The process of how we perform ink limitation is as follows:
The actual amount of colorant that is input to halftoning algorithm is

$$I_{actual} = I_{post\_gcr} \frac{4/3}{\bar{I}^{total}_{post\_gcr}} \tag{6}$$

where $I = C,M,Y,K$. Then we have

$$\bar{I}^{total}_{actual} = C_{actual} + M_{actual} + Y_{actual} + K_{actual} \tag{7}$$

And we have $\bar{I}^{total}_{actual} \equiv \frac{4}{3}$. Thus all ratios of ink amounts are preserved, e.g.

$$\frac{I_{actual}}{J_{actual}} = \frac{I_{post\_gcr} \frac{4/3}{\bar{I}^{total}_{post\_gcr}}}{J_{post\_gcr} \frac{4/3}{\bar{J}^{total}_{post\_gcr}}} = \frac{I_{post\_gcr}}{I_{post\_gcr}} \tag{8}$$

where $I,J = C,M,Y$ or K and $I \neq J$.

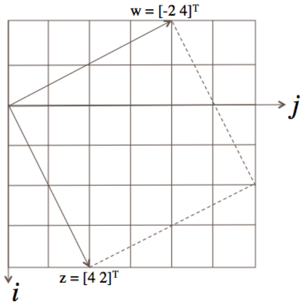


Figure 1. Continuous-parameter halftone cell (CPHC) of microcell design with parameters $\mathbf{z} = [4,2]$ *and* $\mathbf{w} = [-2,4]$.

II.D. Halftoning (IS-SF-SA-MD)

In this section, we discuss traditional concepts for design of periodic clustered-dot halftone screens, including specifying lpi and angle for the screen, choosing tile vectors, determining microcell size and shape, determining BSB, and determining dot growth rules for the microscreen. Then, we discuss the generalization of traditional halftoning methods to supercell and other possibilities. First, we discuss the traditional concepts.

II.D.i. Traditional concepts

1. Specify lpi and angle

First of all, our target frequency is 133 lpi, which is on the lower side of the screen frequencies that are typically used for printing. And the target angle is between 15 and 30 degrees. We avoid having 0, 45 and 90 degree angle screens because they will create a visible pattern. Based on these parameters, we choose the tile vectors for our regular screen.

2. Choose tile vectors

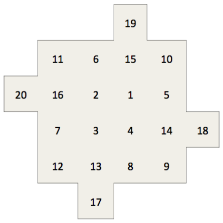
A periodic screen can be defined by two vectors, $\mathbf{z} = [z_i, z_j]$ and $\mathbf{w} = [w_i, w_j]$ as shown in Fig. 1. Here, vectors $\mathbf{z}$ and $\mathbf{w}$ are screen tile vectors, and the shape formed by these vectors defined the continuous-parameter halftone cell (CPHC). Screen tile vectors have the property that if we shift and copy the CPHC by $n_1\mathbf{z} + n_2\mathbf{w}, (n_1, n_2) \in \mathbb{Z}^2$, then we can cover the entire spatial domain.

The restriction for the screen tile vectors is that they are integer pairs, i.e.,$\mathbf{z}, \mathbf{w} \in \mathbb{Z}^2$ in a bilevel screen design. There are different types of screen tile vectors depending on whether the tile vectors are orthogonal or not, and the length of the tile vectors are equal or not. If $\mathbf{z} \cdot \mathbf{w} = 0$, then we call the screen an orthogonal screen. If $\mathbf{z} \cdot \mathbf{w} \neq 0$ then we call the screen a nonorthogonal screen. And if $\mathbf{z}$ and $\mathbf{w}$ satisfy $|\mathbf{z}| = |\mathbf{w}|$ and $\mathbf{z} \cdot \mathbf{w} = 0$, then we have a square screen. If $\mathbf{z}$ and $\mathbf{w}$ satisfy $|\mathbf{z}| = |\mathbf{w}|$, but $\mathbf{z} \cdot \mathbf{w} \neq 0$, then we have a parallelogram screen with equal sides. Further, if $|\mathbf{z}| \neq |\mathbf{w}|$ and $\mathbf{z} \cdot \mathbf{w} = 0$, we have a rectangular screen, and if $|\mathbf{z}| \neq |\mathbf{w}|$ and $\mathbf{z} \cdot \mathbf{w} \neq 0$, we have a parallelogram screen with unequal sides.

In our design, we choose a square screen. Since the tile vectors are $\mathbf{z} = [4,2]$ and $\mathbf{w} = [-2,4]$, the periodicity matrix is defined as $\mathbf{N} = [\mathbf{z}^T | \mathbf{w}^T]$. The number of pixels in the screen is thus $det|\mathbf{N}| = 20$ in our design. The screen frequencies in the $\mathbf{z}$ and $\mathbf{w}$ directions are defined as $f_{\mathbf{z}}$ and $f_{\mathbf{w}}$, respectively. We have $f_{\mathbf{z}} = R/||\mathbf{z}||$ and $f_{\mathbf{w}} = R/||\mathbf{w}||$, where $R = 600$ dpi is the printer resolution, and the units of $f_{\mathbf{z}}$ and $f_{\mathbf{w}}$ are lines/inch (lpi). In our design, $f_{\mathbf{z}} = f_{\mathbf{w}} = 134.16$ lpi. This frequency is on the lower side of screen frequencies. The screen angle is defined as the angle between one of the tile vectors and the $+j$ axis. In our case, since the screen is orthogonal, the screen angle is defined to be the angle between vector $\mathbf{w}$ and the $+j$ axis. So we have $\theta = arctan(-w_i/w_j)$. For our design, $\theta = 26.57$ degrees.

3. Determine the discrete microcell size and shape

Based on the CPHC design, we need to determine the shape of the block with boundaries lying on the grid of printer-addressable points, which also matches with the CPHC. This is defined to be the discrete-parameter halftone cell (DPHC) design corresponding to the CPHC, with the same frequency and angle parameters. The DPHC is not rectangular shaped, but in order to be used for computation purpose, we need to tile the microcell to form a rectangular screen using the Holladay method. The halftone cell defined by CPHC has fraction of a pixel inside the halftone cell, the DPHC cell is chosen based on the shape of CPHC. Fractional area of the pixel covered by the CPHC is not calculated, and the pixels that has more area covered by the CPHC may not be the ones chosen to be included. In the DPHC cell chosen, the total area of the CPHC and the DPHC are the same.



			19		
	11	6	15	10	
20	16	2	1	5	
	7	3	4	14	18
	12	13	8	9	
		17			

Figure 2. Discrete-parameter halftone cell (DPHC) of microcell designed with parameters $\mathbf{z} = [4,2]$ *and* $\mathbf{w} = [-2,4]$.

4. Determine BSB

The basic screen block (BSB) is the smallest rectangle that can be tiled vertically and horizontally. The height H_B and width W_B of the BSB is

$$H_B = \left| \frac{N_m}{gcd(z_j, w_j)} \right|, W_B = \left| \frac{N_m}{gcd(z_i, w_i)} \right| \tag{9}$$

where N_m is the number of pixels in the microcell. For our design, specified in Fig. 1, we have $H_B = W_B = 10$.

5. Determine the dot growth rules

In order to determine the dot growth rules (microscreen index array), we need to consider the growth rules. The design rules for microcell are:

- Maintain existing dot- or hole-clusters by adding new

dots on the periphery of the dot-cluster and adding new holes on the periphery of the hole-cluster. This rule is essential for maintaining the clustered-dot/clustered-hole structure of the halftone patterns.
• Maintain symmetry of dot and hole clusters as much as possible. This is especially essential in midtones, where a perfectly complementary-symmetric design will form a checkerboard pattern.
• Take into account the periodicity of the halftone screen.
According to these rules, we designed the microcell growing sequence, numbered 1 to 20, as shown in Fig. 2. Note that the design is not unique.

II.D.ii. Generalizations

After discussing the traditional concepts for design of periodic clustered-dot halftone screens, we now generalize to supercell. In this method, we need to design a macroscreen sequence. Combining this with the microcell sequence, we are able to generate the supercell. We also discuss hybrid screen for the purpose of improving the screen quality. Further, we discuss a more generalized concept of core.

1. Supercell design

Supercell is designed by tiling a number of microcells together. In order to have rectangular shape for the supercell to simplify computation, we tile several BSBs together to form a rectangular supercell. The supercell height and width can be multiples of the BSB size:

$$H = M_H H_B, W = M_W W_B \tag{10}$$

where M_H and M_W are arbitrary integer multiplication factors in the horizontal and vertical directions respectively. To achieve acceptable image quality, we choose the size of supercell to be 260×260.

Next, we need to design the macrocell sequence. This sequence is the order in which dots are added to each microcell. For example, starting in the highlights, we turn on the first pixel (numbered 1 in each microcell) in the macrocell sequence. In order to have a good quality of the first dot turn-on sequence, we use DBS to determine the macrocell sequence.

2. Hybrid screen

Our hybrid screen creation is similar to Lee and Allebach's hybrid screen generation method[9]. The target hybrid screen has the property of complementary symmetry.

2.1 Highlights and shadows generation

The highlight region, corresponding to pixels with indices 1 through 4 in Fig. 2, is generated using DBS. The total number of levels L is the total number of pixels in the hybrid screen plus one.

$$L = L_M N_m + 1 \tag{11}$$

where L_M is the length of the macrodot growing sequence. In order to generate the index matrix $d[\mathbf{m}]$, we generate the dot profile function for highlights, we start by generating $p[\mathbf{m}, 2L_M/(L-1)]$ using DBS. The dots in the cores are randomly initialized, and swap only within all cores are performed to obtain the best halftone texture for the dot profile. Other dot profiles are generated sequentially after the first dot profile is generated, i.e. from $p[\mathbf{m}, 2L_M/(L-1)]$ to $p[\mathbf{m}, 1/(L-1)]$, then from $p[\mathbf{m}, (2L_M+1)/(L-1)]$ to $p[\mathbf{m}, 4L_M/(L-1)]$. Here, it is assumed that each highlight core contains four pixels. The dot profiles are generated following the stacking constraint. The index matrix for highlights can be generated using the dot profile.

$$d^h[\mathbf{m}] = \begin{cases} N_h L_M - \sum_{i-1}^{N_h L_M} p[\mathbf{m}; \frac{i}{L-1}] & \text{if } \mathbf{m} \in \Omega_h \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where Ω_h is the highlight core region.

The shadow core region, corresponding to pixels 17 through 20 in Fig. 4, can be generated in similar way as the highlight core. Following Lee and Allebach's approach, the shadow core region is given by:

$$d^s[\mathbf{m}] = \begin{cases} L-1-d^h[(\mathbf{m}-\mathbf{m}_c) mod[H,W]] & \text{if } \mathbf{m} \in \Omega_s \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Where $\mathbf{m}_c$ is the offset vector relative to the highlight core, and Ω_s is the shadow core region.

2.2 Hybrid screen generation

We described the macrodot growing sequence $d_M[\mathbf{m}] \in \{0, 1, ..., L_M - 1\}$ in the supercell design section. Following Lee and Allebach's hybrid screen generation method, we need both microcell and macrocell growing sequence, the microcell cluster growing index is $d_m[\mathbf{m}] \in \{N_h, N_h + 1, .., N_m - N_s - 1\}$, where N_h and N_s are the number of pixels in the highlight and shadow cores, respectively. Given the macrodot growing index $d_M[\mathbf{m}]$, which we generated before, we can express the index matrix for midtone region as

$$d^m[\mathbf{m}] = \begin{cases} L_M d_m[\mathbf{m}] + d_M[\mathbf{m}] & \text{if } \mathbf{m} \in \Omega_m \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where Ω_m is the midtone region. Now that we have the highlight, midtone, and shadow index matrices $d^h[\mathbf{m}]$, $d^m[\mathbf{m}]$, and $d^s[\mathbf{m}]$, we can generate the complete index matrix by summing the three index matrices.

$$d[\mathbf{m}] = d^h[\mathbf{m}] + d^m[\mathbf{m}] + d^s[\mathbf{m}] \tag{15}$$

This index matrix is used for our novel halftoning algorithm, which we will discuss in Section V.

2.3 Generalize concept of core

We may further optimize the hybrid screen method by generalizing to multiple microscreen index arrays designed by DBS jointly with the macroscreen index array.

II.E. Multilevel halftoning

The traditional multilevel halftoning method [8], shown in Fig. 3, is incorporated in our new halftoning algorithm. The idea in the traditional multilevel halftoning method is that we have no drop (absorptance 0), 1 drop (absorptance $\frac{1}{3}$), 2 drops (absorptance $\frac{2}{3}$), and 3 drops (absorptance 1). The total number of levels as a result of multilevel halftoning is $D \cdot N + 1$ without supercell, where D is maximum number of drop per pixel, and N is number of pixels in the microcell.

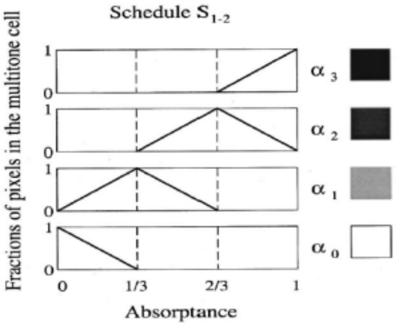


Figure 3. *Schedule S_{1-2} for traditional one-two tone mixture multitone screen [8]*

II.F. Display image

In order to display different number of drops of ink, we use graylevel 0, 33, 67, 100 to represent, no drop, one drop, two drops and three drops, respectively for display purposes.

II.G. Estimate ink saving

We estimate the ink savings by counting the average number of drops/pixel for the halftoned image compared with that required for a conventional periodic, clustered dot screen.

III Novel halftoning method

After the GCR and ink limitation steps, which serve as preprocessing for the halftoning algorithm, we now discuss the novel halftoning method. The novel idea is to minimize overlap between the C, M, Y, and K colorants with multilevel halftoning and ink limitation. The general idea is shown in Fig. 4.

We place black ink and the C, M and Y inks separately, because black overlapping with other colorants reduces the effectiveness of the C, M, and Y colorants. In our method, we also separate the C, M, and Y colorants as much as possible, as shown in Fig. 4.

In our novel halftoning method, the design rules we follow are:

1. Minimize the overlap between C,M,Y,K colorants.
2. All colorants follow the traditional 1-2 schedule for multilevel halftoning.
3. We will not print 2 drops on any pixel until every pixel has at least 1 drop. We will not print 3 drops on any pixel until every pixel has at least 2 drops. We will not print 4 drops on any pixel until every pixel has at least 3 drops (Leveling Rule).

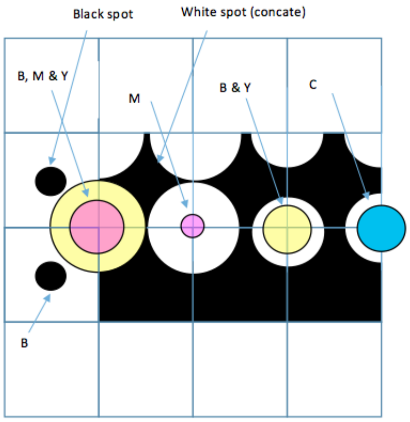
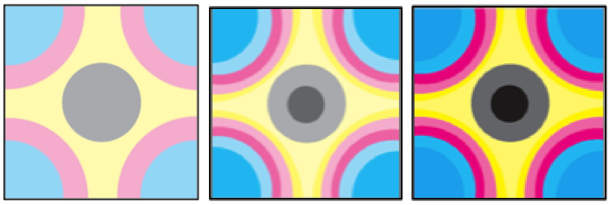


Figure 4. *Novel halftoning idea.*



(a) Case 1 (b) Case 2 (c) Case 3

Figure 5. *Halftone growth rules for Cases 1, 2 and 3.*

4. We will not print anything on 3 drops of K (Ink Efficiency Rule).
5. We cannot print more than 4 drops on any pixel (Ink Limit Rule).
6. Save ink.

According to the above rules, we halftone according to $\bar{I}^{total}_{half}$, which is the summation of C, M, Y, K colorant amounts, C_{half}, M_{half}, Y_{half} and K_{half}; and the novel halftoning method is divided into 4 cases.

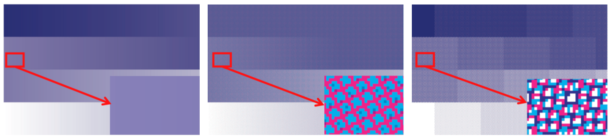
- Case 1: when $0 \le \bar{I}^{total}_{half} \le \frac{1}{3}$, we can print the requested amounts of C, M, Y, K with only 1 drop at each pixel of the halftone cell.
- Case 2: when $\frac{1}{3} < \bar{I}^{total}_{half} \le \frac{2}{3}$, we can satisfy the request by printing 1 or 2 drops at every pixel in the halftone cell.
- Case 3: when $\frac{2}{3} < \bar{I}^{total}_{half} \le 1$, we can satisfy the request by printing 2 or 3 drops at every pixel in the halftone cell.
- Case 4: when $1 < \bar{I}^{total}_{half} \le \frac{4}{3}$, we can satisfy the request by printing 3 or 4 drops at every pixel in the halftone cell.

As shown in Fig. 5, in Case 1, we place all colorants separate from each other, black grows from the center of K, and cyan grows from the center of the C, M, and Y colorants, followed by magenta and yellow, as illustrated in Fig. 4. To determine the exact location of C, M, Y and K, we refer to the hybrid index matrix $d_K[\mathbf{m}]$ as the hybrid screen for K, and create another hybrid screen $d_{CMY}[\mathbf{m}]$, centered at the location of the C, M and Y colorants, which is located 45° relative to the center of CMY. The placement of C, M, Y, K is based on the idea that each colorant is assigned to a set of pixels in the hybrid index matrix according to its relative percentage. For black, we calculate the percentage of K_{half} relative to $\bar{I}_{half}^{total}$ and assign $L \cdot \frac{K_{half}}{\bar{I}_{half}^{total}}$ pixels starting at the center of the black region to the black colorant. For cyan, we assign indices $0, 1, ...L \cdot \frac{C_{half}}{\bar{I}_{half}^{total}} - 1$ in $d_{CMY}[\mathbf{m}]$ to the cyan colorant. For magenta, we assign $L \cdot \frac{M_{half}}{\bar{I}_{half}^{total}}$ pixels starting at index $L \cdot \frac{C_{half}}{I_{half}^{total}}$ to magenta. For yellow, we assign $L \cdot \frac{Y_{half}}{\bar{I}_{half}^{total}}$ pixels starting at index $L \cdot \frac{C_{half}+M_{half}}{I_{half}^{total}}$ to yellow.

As $\bar{I}_{half}^{total}$ grows, we follow the same relative placement for all the colorants, and put the same colorant on itself, thus reducing the overlap between different colorants. As $\bar{I}_{half}^{total}$ keeps increasing, in Case 3, we have a mixture of 2 drops and 3 drops for each colorant. However, when $\bar{I}_{half}^{total} > 1$, we need to add the 4th drop to each colorant in a different manner, since we cannot print 4 layers of the same colorant on any pixel. We place colorant I on top of colorant J, and keep the ratio of colorant I and J constant, where $I, J = C, M, Y, K$, and $I \neq J$. We also avoid putting any colorant on top of 3 drops of colorant K.

IV Experimental results

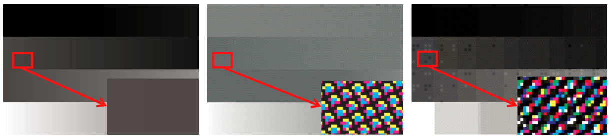
We compare our halftoning algorithm result with a conventional clustered-dot channel-independent halftoning method [10][11] result using a comparable screen frequency. We first compare halftoned images of a ramp that consists of cyan and magenta equally increasing from absorptance 0 to 1. The comparison between the novel and traditional halftoning algorithms is shown in Fig. 6.



(a) Continuous-tone (b) Novel halftone (c) Traditional

Figure 6. Comparison of ramp images with C and M equally increasing.

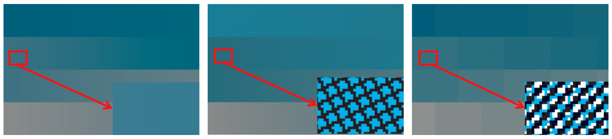
Figure 7 shows the halftoned images of a ramp with C, M, Y, K equally increasing from absorptance 0 to 1.



(a) Continuous-tone (b) Novel halftone (c) Traditional

Figure 7. Comparison of ramp images with K, C, M, Y equally increasing.

Figure 8 shows the halftoned images of a ramp with K fixed (absorptance is 0.5), while C increases from absorptance 0 to 1.



(a) Continuous-tone (b) Novel halftone (c) Traditional

Figure 8. Comparison of ramp images with K fixed, C increasing.

The ink savings for the ramps shown in Figs. 6-8 is listed in Table. 1.

Table 1: Ink savings with novel halftoning algorithm compared to the traditional algorithm for the ramps shown in Figs. 6-8.

	Percentage of ink saving (%)			
Fig.	C	M	Y	K
6	10.48	10.51	0	0
7	66.52	66.48	66.03	32.48
8	15.07	0	0	8.47

Note that the traditional halftone images shown in Figs. 6-8 are not based on a supercell. So they exhibit contouring artifacts due to quantization. Also, the novel halftone ramps have not been color managed. As a consequence of the rules described in Sec. III, these ramps are not color matched to the continuous-tone images. At this point, we have not investigated what is the gamut for our novel halftoning algorithm. If it is deficient in some aspects, it may be necessary to modify the design rules described in Sec. III.

V Conclusions

We have introduced a novel single-angle periodic, clustered-dot halftoning method that eliminates the traditional rosette artifacts due to overlapping halftone patterns for each colorant. Colorant overlap is minimized in order to improve ink efficiency. Our method supports multi-drop halftoning. We show significant potential ink savings.

References

[1] T. N. Pappas, J. P. Allebach, and D. L. Neuhoff, ”Model-based digital halftoning,” *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 14-27, Jul. 2003.

[2] W. Jiang, W. Xi, U. Sarkar, R. Ulichney, and J. P. Allebach, ”Color halftoning based on neugebauer primary area coverage”, in *Proc. of IS&T Symposium of Electronic Imaging, Color Imaging XXII,* pp. 91-102(12), Burlingame, US, Jan. 2017.

[3] W. Xi, T. Frank, Y. Ben-Shoshan, R. Ulichney, and J. P. Allebach, ”Color halftoning based on multi-stage, multi-pass, clustered-DBS”, in *Proc. of IS&T Symposium of Electronic Imaging, Color Imaging XXII,* pp. 103-113(11), Burlingame, US, Jan. 2017.

[4] Franklin, C. (1985). A quantitative analysis of the Floyd-Steinberg (error diffusion) halftoning algorithm. 143-5.

[5] R. Ulichney, ”The void-and-cluster method for dither array generation,” in Proc. SPIE, Human Vision, Visual Processing and Digital Display IV, 1993, vol. 1913, pp. 332-343.

[6] Grey component replacement (2017, June 12). In Wikipedia, The Free Encyclopedia. Retrieved 22:19, Dec.

19, 2017, fromhttps://en.wikipedia.org/w/index.php?title=Grey_component_replacement&oldid=785274671
Spiridonov, M. Shopova, R. Boeva, ”Study the effect of gray component replacement level on reflectance spectra and color reproduction accuracy,” in Proc. SPIE 8770, 17th International School on Quantum Electronics: Laser Physics and Applications, 87700W (15 March 2013)

[7] Davor Donevski, Ante Poljicak, Maja Strgar Kurecic, Colorimetrically accurate gray component replacement using the additive model, In *Journal of Visual Communication and Image Representation*, vol. 44, pp. 40-49, 2017

[8] G.-Y. Lin and J. P. Allebach, ”Multilevel screen design using direct binary search”, *J. Opt. Soc. Amer. A*, vol. 19, no. 10, pp. 1969-1982. Oct. 2002

[9] C. Lee and J. P. Allebach, ”The hybrid screen - Improving the breed”. *IEEE Trans. on Image Processing*. Vol. 19, pp. 435-450. Feb. 2010.

[10] F. A. Baqai and J. P. Allebach, ”Computer-aided design of clustered-dot color screens based on a human visual system model”, in *Proceedings of the IEEE*, vol. 90, no. 1, pp. 104-122. Jan. 2002

[11] X. Zhang, A. Veis, R. Ulichney and J. P. Allebach, ”Multilevel halftone screen design: Keeping texture or keeping smoothness?”, *2012 19th IEEE International Conference on Image Processing*, Orlando, FL. pp. 829-832. 2012.

Author Biography

Wanling Jiang received her BS in Measurement and control technology and instrumentation from Dalian University of Technology (2011) and her MS degree in Electrical Engineering from University of Southern California (2013). Now, she is working on halftoning at Purdue University. Her work has focused on the optimization of halftone qualities.