The following publication Yik-Hing Fung and Yuk-Hee Chan "Blue noise digital color halftoning with multiscale error diffusion," Journal of Electronic Imaging 23(6), 063013 (9 December 2014) is available at https://doi.org/10.1117/1.JEI.23.6.063013.

Journal of Electronic Imaging

JElectronicImaging.org

Blue noise digital color halftoning with multiscale error diffusion

Yik-Hing Fung
Yuk-Hee Chan

Journal of Electronic Imaging 23(6), 063013 (Nov/Dec 2014)

Blue noise digital color halftoning with multiscale error diffusion

Yik-Hing Fung and Yuk-Hee Chan*
The Hong Kong Polytechnic University, Department of Electronic and Information Engineering, Hong Kong, HKSAR

Abstract. A recent trend in color halftoning is to explicitly control color overlapping, dot positioning, and dot coloring in different stages of the halftoning process. As feature-preserving multiscale error diffusion (FMED) shows its capability and flexibility in dot positioning in both binary and multilevel halftoning applications, it naturally becomes a potential candidate in the development of new color halftoning algorithms. This paper presents an FMED-based color halftoning algorithm developed based on the aforementioned strategy. In contrast to the algorithms that adopt the same strategy, the proposed algorithm has no bias for specific Neugebauer primaries in dot coloring and has no fixed scanning path to place dots in dot positioning. These features allow the algorithm to place dots of the correct colors on the right positions with less constraint. Hence, the algorithm is better able to preserve image features. *© 2014 SPIE and IS&T* [DOI: 10.1117/1.JEI.23.6.063013]

Keywords: multiscale error diffusion; color error diffusion; halftoning; multiscale processing; color; printing; blue noise.
Paper 14407 received Jul. 17, 2014; accepted for publication Oct. 20, 2014; published online Dec. 9, 2014.

1 Introduction

Digital color halftoning is a technique used to render a continuous-tone color image with a limited number of binary color planes and has been widely used in color printing applications.[1–3] To a certain extent, it can be considered as an evolution of digital monochrome halftoning in which a continuous-tone monochrome image is rendered with a binary image.

There are various monochrome halftoning schemes, and each one of them can theoretically be evolved to realize color halftoning. These schemes can be roughly classified as screening-based schemes,[4–6] error diffusion based schemes,[7–10] and optimization-based schemes, which try to iteratively optimize an objective function.[11,12] Iterative schemes generally provide the best quality in terms of the objective function being optimized at the cost of complexity, while screening-based schemes are of the lowest complexity. When one opts for a good quality at a cost of affordable complexity, an error diffusion based monochrome halftoning scheme[7–10] is generally used.

Separable error diffusion (SED) is the simplest extension of error diffusion for color halftoning. It independently processes each color channel of a color image with error diffusion and then superimposes the generated halftones to form a color halftone. Because SED does not exploit the intercolor correlation, it generally leads to color artifacts and poor color rendition. To solve this problem, vector error diffusion takes the interchannel color correlation into account by either jointly quantizing all color channels or diffusing color errors across channels.[13–21] When the quantizer is overloaded, there can be other significant color artifacts, such as smear and slow response, which make it difficult to preserve high-frequency spatial features. Another approach is to halftone a color image in the colorant space to directly control the color halftone texture.[22–27] Readers can read Refs. 2 and 3 for a comprehensive review on color error diffusion.

Recently, He proposed a hierarchical error diffusion (HED) algorithm[28] for color error diffusion. HED is different from conventional color error diffusion algorithms because its output is produced through neither conventional joint quantization nor interchannel error diffusion. Instead, it explicitly controls color overlapping, dot positioning, and dot coloring at different stages to produce a color halftone. As HED is able to produce color halftones with superior quality, it shows a new direction for developing error diffusion based color halftoning algorithms. Shortly afterward, He extended the idea of HED and developed a hierarchical colorant based direct binary search (HCB-DBS) halftoning algorithm[29] which is able to provide color halftones of even higher quality.

Feature-preserving multiscale error diffusion (FMED)[30] is a monochrome halftoning technique developed based on multiscale error diffusion.[31] Extensive studies on FMED show that it is able to eliminate directional hysteresis, preserve spatial features, and provide outputs with good blue noise characteristics.[32–37] During its realization, pixels are not processed sequentially with a predefined order, which allows one to effectively and efficiently control dot positioning. Because controlling color overlapping, dot positioning, and dot coloring is the key to success in HED, we would like to explore whether fusing HED and FMED can create greater synergy in color halftoning. This paper presents an FMED-based color halftoning algorithm developed in our recent study of this issue.

The organization of this paper is as follows. Section 2 provides a brief review on HED. Section 3 presents our proposed FMED-based color halftoning algorithm. A tone-dependent error diffusion filter is proposed to support dot positioning control in the proposed color halftoning

*Address all correspondence to: Yuk-Hee Chan, E-mail: enyhchan@polyu.edu.hk

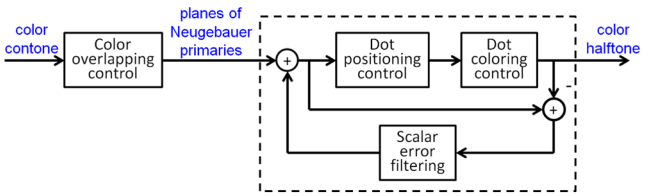


Fig. 1 General framework of hierarchical error diffusion (HED).

algorithm. The details of its design and the rationale of its design will be discussed in Sec. 4. Section 5 provides simulation results on real images for evaluation. Finally, a conclusion is given in Sec. 6.

2 Review on HED

Unlike conventional color error diffusion algorithms, HED explicitly controls color overlapping, dot positioning, and dot coloring at different stages to produce a color halftone as shown in Fig. 1.[28]

In HED, the color overlapping control module decomposes a color contone into a set of monochrome contones, one for each Neugebauer primary, to minimize the brightness variation at the pixel density level. The minimization of brightness variation color density comes from the idea that one should render an input color with the Neugebauer primaries of less brightness contrast instead of those of more brightness contrast to reduce the perception of different patterns in a color halftone. The color overlapping control algorithm suggested in HED, which is referred to as minimum brightness variation conversion (MBVC) hereafter, is independent of the subsequent halftoning step and does not enforce any halftone color restriction at the quantization step. Once it is carried out, no further color overlapping control is needed.

In the subsequent halftoning step, pixels are sequentially processed. For each pixel, a partial density sum vector (PDSV) is constructed and thresholded to determine whether a dot should be placed on the same pixel location in the output without concern for the color of the dot. Its color is determined only when the decision is positive. As dot positioning is decoupled from dot coloring, it allows one to control the positions of dots with less constraint and, conceptually, it would be easier to provide an output of blue noise characteristics.

In order to promote using colors of less brightness contrast to render an input color, Neugebauer primaries are ordered according to their brightness such that, when selecting the color of the dot to be placed during dot coloring, HED always selects the color within a predefined subset of Neugebauer primaries. This can introduce a bias in dot coloring control.

Another observation we have had is that HED pixels are processed sequentially according to a predefined scanning order. This limits the flexibility in dot positioning, and hence, spatial features in the original image are difficult to preserve in the color halftoning output.

3 Proposed Color Error Diffusion Algorithm

In this section, an FMED-based color error diffusion algorithm is proposed. As shown in Fig. 2, this algorithm consists of four operational stages. It naturally carries out its dot overlapping control, dot positioning control, and dot coloring control in various stages with the help of MBVC (Ref. 28) and FMED.[30]

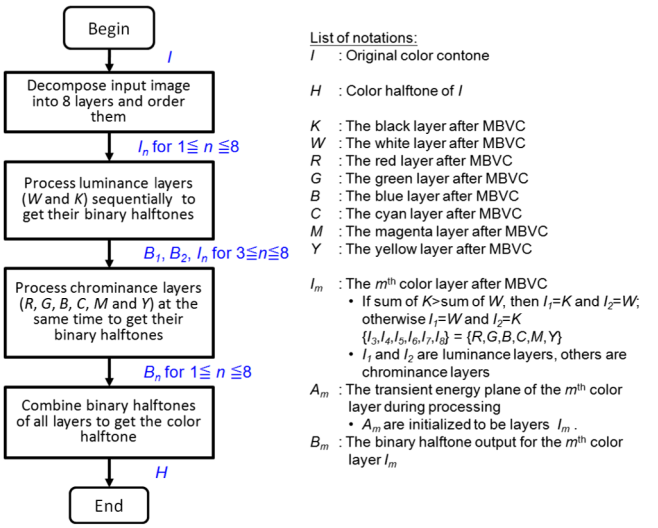


Fig. 2 Flow of the proposed algorithm.

3.1 *Overview of the Proposed Color Error Diffusion Algorithm*

Consider the case that we want to convert a full-color image I of size $N \times N$ pixels to a color halftone H. The intensity value of each color component of a pixel is normalized such that it is bounded in [0,1]. Without loss of generality, we assume that the input pixel color is in the CMYK specification as is the common case for digital printing applications.

The proposed color error diffusion algorithm consists of four operation stages as shown in Fig. 2. In its first stage, the input color image in the CMYK color space is first decomposed by MBVC (Ref. 28) to produce a set of multilevel intensity planes, each of which represents the spatial intensity distribution of a specific color channel of the input color image. These color channels include channels K, W, R, G, B, C, M, and Y (i.e., black, white, red, green, blue, cyan, magenta, and yellow). Accordingly, there are at most eight intensity planes at the output of the first stage. These color channels are classified into two groups. The channels that do not carry chrominance information (i.e., channels K and W) are classified as luminance channels, while the remaining channels are classified as chrominance channels.

The luminance channels are then processed in the second stage before the chrominance channels are processed. The processing of a specific channel is used to position the dots of the color associated with the channel. Luminance channels are processed first because the human visual system is more sensitive to the luminance contrast than the chrominance contrast. Positioning black and white dots is, hence, more critical than positioning chromatic dots, especially where feature preserving is concerned. This stage produces two binary halftones for channels K and W, respectively, based on their associated intensity planes. In practice, the binary halftones define where black or white pixels should appear at the final color halftoning output. Accordingly, they set a constraint for the algorithm to put chromatic dots on the color halftone in the next stage.

In the third stage, all chrominance channels are collectively processed to produce their corresponding binary halftones without a preset priority order. Chromatic dots are placed one by one and the intensity planes of the chrominance channels are adjusted accordingly. The most updated intensity planes of all chrominance channels are combined to form a cost map, which defines the extent to which a dot should be put on a specific pixel at that specific moment. Based on this cost map, the algorithm locates a pixel location that has not yet been occupied by any dots and then determines the color of the dot to be assigned to that location. Since there is no predefined preference for a particular chrominance channel when processing them, no bias is introduced to any of them. In short, the algorithm searches for the most critical position in the image and places a dot of the most appropriate color for that position in the corresponding pixel position of the halftoning output. This approach of dot positioning and coloring control is useful for handling a real image.

Conventional color halftoning algorithms generally prioritize the color channels and process them sequentially. Unlike a constant patch, a real image has spatially variant content, and hence the color priority should be adaptive to local regions. Adopting a fixed color priority order for the whole image is obviously not a good strategy with which to faithfully report the original local features of an image.

The output of stage three is six binary halftones, each of which defines a dot pattern for a particular chrominance channel. The final color halftone is produced by combining the binary halftones of all the color channels. Note that each binary halftone can be interpreted as a color mask in which 1 marks a pixel having the corresponding color and 0 marks a transparent pixel. Because a dot can only be assigned to a location that has not been occupied by other dots when producing the binary halftones, dots of different colors cannot overlap. Hence, the color halftone can be obtained by simply superimposing all color masks. Dot overlapping control is automatically achieved.

3.2 *Details of Stage 1*

In order to reduce the noticeable placement of the dots in a smooth region, it is necessary to render the color in the region with dots of the least contrast as far as possible. In Ref. 28, He proposed a CMYK to CMYRGBK transform named the MBVC to control the mixture ratio of different available colorants for producing a given color such that minimum brightness variation color density can be achieved in a smooth region. (In He's original proposal, MBVCD can produce both K (black) and K_p (composite black). However, as K and K_p are mutually exclusive in the final printout of an image, we also denote K_p as K in this paper to simplify the presentation. In other words, K actually means K_p when K is not supported).

To achieve the same goal, the proposed algorithm performs MBVC for each pixel to produce seven color components for composing the color of the pixel. For reference, the intensity plane of a particular color component is referred to as a layer in this paper. Accordingly, we have layers R, G, B, C, M, Y, and K. Let $\Phi_c = \{R, G, B, C, M, Y\}$ be the set of layers that carry chrominance information. An extra color layer W can then be defined as follows:

$$W(i,j) = 1 - \sum_{L \in \Phi_c} L(i,j) - K(i,j) \quad \forall\ (i,j), \tag{1}$$

where $W(i,j)$ is the intensity value of pixel (i,j) of layer W and $L(i,j)$ is that of layer L for $L \in \Phi_c$. Layers W and K do not carry any chrominance information and form a set denoted as Φ_l.

For reference purpose, all layers are indexed as follows:

$$I_1 = \begin{cases} W & \text{if } \sum_{(i,j)} W(i,j) \geq \sum_{(i,j)} K(i,j) \\ K & \text{else} \end{cases},$$

$$I_2 = \begin{cases} K & \text{if } \sum_{(i,j)} W(i,j) < \sum_{(i,j)} K(i,j) \\ W & \text{else} \end{cases},$$

$$I_3 = R, \quad I_4 = G, \quad I_5 = B, \quad I_6 = C,$$

$$I_7 = M, \quad \text{and} \quad I_8 = Y. \tag{2}$$

Accordingly, we have $\Phi_c = \{I_3, I_4, I_5, I_6, I_7, I_8\}$ and $\Phi_l = \{I_1, I_2\}$, where $\sum_{(i,j)} I_1(i,j) \geq \sum_{(i,j)} I_2(i,j)$. In the later discussion, we refer to the colorant associated with layer I_m as colorant m and its color as color m.

In short, the algorithm decomposes a color image into eight layers, classifies them into two groups, and indexes them for further processing in this stage.

By considering white as a color, producing a color halftone is now equivalent to placing dots of eight different colors on the output plane to fill up all its pixels. MBVC predefines the numbers of different color dots and guarantees that dots need not overlap.[28]

In general, the total number of pixels containing a specific color can be determined as

$$\mathrm{DB}_m = \sum_{(i,j)} I_m(i,j) \quad \text{for } 0 < m < 9, \tag{3}$$

where I_m is the layer associated with the color. Hence, DB_m is the dot budget of the color and can be interpreted as the number of 1's appearing in the binary halftoning result of layer I_m. MBVC guarantees that the total number of image pixels equals $\sum_{0<m<9} \mathrm{DB}_m$, and hence, no overlap of dots is required.

Note that MBVC has already helped to achieve minimum brightness variation color density by controlling the mixture ratio of different color dots. The major concern in subsequent stages will be, given the provided mixture ratio of the color dots, how to highlight the image details by controlling the positions of the available color dots.

3.3 *Details of Stage 2*

With the eight layers on hand, the proposed algorithm converts each one of them into a binary halftone, which defines the pixels containing a specific color in the color halftone.

Let B_m be the binary halftoning outputs of I_m for $0 < m < 9$. They are initialized such that $B_m(i,j) = -1$ for all (i,j), which means that the algorithm has not yet decided whether a dot should be placed on any one of the pixels in the output image at the beginning of stage two. In stages two and three, dots of different colors are placed on the output image one by one and the placements are registered in B_m as follows:

$$B_m(x_0,y_0)=\begin{cases}1 & \text{if a dot of color } m \text{ is placed on } (x_0,y_0)\\ 0 & \text{if a dot of other color is placed on } (x_0,y_0)\end{cases}, \tag{4}$$

where color m is the color associated with layer I_m. At the end of color halftoning, the output image is filled up with dots of different colors (as the sum of dot budgets = the sum of pixels) and all elements in B_m have value of either 0 or 1. B_m is then the final binary halftone of layer I_m.

Among all the layers, layers in $\Phi_l=\{I_1,I_2\}$ are processed in stage two first as our eyes are more sensitive to luminance contrast than chrominance contrast. Figure 3 summarizes the workflow of stage 2 in a form of pseudo code. Because dots cannot overlap, positioning black and white dots first allows the algorithm to have more freedom to place white and black dots, and hence, helps to preserve the local features in an image.

Layers I_1 and I_2 are handled sequentially. Layer I_1 is handled first because it carries more energy. Theoretically, one can use any binary halftoning algorithm to produce a halftone for a layer. However, FMED is used here because it has been proven to be good at preserving local features and flexible enough to carry out dot positioning control.[30,32–37]

Without loss of generality, consider the case that we are handling layer I_n. In general, FMED iteratively locates a pixel in I_n to quantize its intensity value to 1 and diffuses the quantization error to the neighboring pixels to update I_n. Accordingly, the intensity profile of layer I_n is adjusted in the course of iteration and the quantization is actually based on the transient intensity profile of I_n. In this paper, to discriminate the transient intensity profile from the original intensity profile of I_n, the former is referred to as A_n, while the latter is still referred to as I_n. In other words, I_n is fixed, while A_n changes with time during the halftoning process. The same definitions apply to other layers. Obviously, we have $A_m=I_m$ for $0<m<9$ before stage two starts.

Quantizing $A_n(x_0,y_0)$ to 1 is equivalent to placing a dot of color n on pixel (x_0,y_0) of the halftoning output. Accordingly, it consumes one dot budget of color n. During the iteration, dots of color n are consumed one by one until the dot budget DB_n is used up and then the iteration stops. For monitoring the progress of dot placement, a mask referred to as M_α is used to indicate which pixels have been placed as dots.

$$M_\alpha(i,j)=\begin{cases}0 & \text{if } (i,j) \text{ has been occupied}\\ 1 & \text{else}\end{cases}. \tag{5}$$

Note that (i,j) can be occupied by the dots of any colorant and mask M_α takes all of them into account. Mask M_α is initialized such that $M_\alpha(i,j)=1$ for all (i,j) before stage two begins.

In stage two, when searching for a pixel to place a dot of color n, the maximum intensity guidance[30] is adopted as follows. Starting with the transient intensity plane A_n as the region of interest, we repeatedly divide the region of interest into nine overlapped subregions of equal size and pick the subregion having the maximum total intensity value among them to be the new region of interest. We repeat the above steps to update the region of interest until a pixel location is reached. One can refer to Ref. 30 for more details.

Once a pixel is located, a dot of color n is placed and B_n is updated with Eq. (4) to record the placement. The transient intensity plane A_n has to be updated by diffusing the error $[=A_n(x_0,y_0)-B_n(x_0,y_0)]$ to pixel (x_0,y_0)'s neighbors in A_n with a diffusion filter $F_{(r_1,r_2)}$ as follows:

$$A_n(i,j)=\begin{cases}0, & \text{if } (i,j)=(x_0,y_0)\\ A_n(i,j)+f(i-x_0,j-y_0)\cdot M_\alpha(i,j)\cdot[A_n(x_0,y_0)-B_n(x_0,y_0)]/\kappa & \text{if } (i-x_0,j-y_0)\in\Omega\setminus\{(0,0)\}\end{cases}, \tag{6}$$

```
% Process layers K and W (i.e. I_1 and I_2) sequentially
%-----------------------------------------------------------------
For n=1,2
    BD_n = Σ_(x,y) I_n(x,y)                          % Get dot budget for layer I_n
    While BD_n≥0.5
        Locate a pixel in A_n via maximum intensity guidance   % Assume the located one is (x_0, y_0)
        B_n(x_0, y_0) = 1                            % Put a dot of color n in (x_0, y_0)
        Diffuse error of A_n(x_0, y_0) to its neighbors in A_n   % Use filter F_(0.7813,0.7813√2)
        For k = n+1:8
            B_k(x_0, y_0) = 0                        % No dot of other colors in (x_0, y_0)
            Diffuse error of A_k(x_0, y_0) to its neighbors in A_k   % Use filter (14)
        End
        BD_n = BD_n − 1                              % Update dot budget for layer A_n
        M_α(x_0, y_0) = 0                            % Update mask M_α
    End
    B_n(x,y) = 0   for all (x,y) in B_n not assigned 1
End
```

Fig. 3 Workflow of stage 2.

where $f(p,q)$ is a filter coefficient of filter $F_{(r_1,r_2)}$, $\Omega = \{(u,v)|u,v \in \mathrm{Z} \text{and} |u|,|v| \le r_2+1\}$ is the filter support, and

$$\kappa = \sum_{(i-x_0,j-y_0)\in\Omega} f(i-x_0,j-y_0)\cdot M_\alpha(i,j) \tag{7}$$

is for filter normalization. In particular, by assuming the error source is at (0,0), $f(p,q)$ is defined as

$$f(p,q) = \frac{A(p,q,r_2)-A(p,q,r_1)}{(r_2^2-r_1^2)\pi} \quad \text{for } (p,q)\in\Omega, \tag{8}$$

where p and q are the horizontal and vertical offsets from the error source in terms of numbers of pixels, r_1 and r_2 are, respectively, the inner and outer radii of a ring defined as $r_2 \ge \sqrt{x^2+y^2} > r_1$, and $A(p,q,r_k)$ for $r_k = r_1$, r_2 is the area covered by circle $\sqrt{x^2+y^2} \le r_k$ in the grid cell of pixel (p,q).

Placing a dot of color n on the output image not only affects B_n, but all B_k for $k \ne n$ as well. It is necessary to adjust the transient intensity planes of all layers with error diffusion. For the adjustment of A_n, diffusion filter $F_{(0.7813,0.7813\sqrt{2})}$ is used since it is found to be optimal for producing binary halftones of blue noise characteristics with FMED.[33] As for the adjustment of A_k for $k \ne n$, a tone-dependent diffusion filter is used instead. A dedicated discussion on the selection of this tone-dependent diffusion filter will be given later in Sec. 4.

After using up the dot budget of layer I_n, the positions of the dots of color n are all determined. All other pixel positions in the final color halftone should have dots of other colors. Accordingly, B_n should be updated as follows before its finalization:

$$B_n(i,j) = 0 \quad \text{if } M_\alpha(i,j) = 1. \tag{9}$$

3.4 *Details of Stage 3*

In order not to introduce a bias to any specific chromatic colors during the halftoning process, layers $I_n \in \Phi_c$ are processed all together without a predetermined order in stage three as summarized in Fig. 4. These chromatic layers are combined to form a cost map as

$$E = \sum_{2<n<9} A_n. \tag{10}$$

A search is then carried out to locate a pixel position to place a dot based on E with the maximum intensity guidance. The search is exactly the same as the one discussed in stage two except that the search is based on the cost map E instead of a particular transient intensity plane A_n.

Once a pixel position, say (x_0,y_0), is located, the color of the dot to be placed on pixel (x_0,y_0) is selected to be the color component that has the maximum transient intensity on that spot and, at the same time, has not yet used up its dot budget. In equation form, the index value of the selected color is determined as

$$s = \underset{k\in\Lambda}{\operatorname{argmax}} A_k(x_0,y_0), \tag{11}$$

where $\Lambda = \{k|2<k\le 8 \text{ and } \mathrm{BD}_k \ge 0.5\}$.

Placing a dot of color s on pixel (x_0,y_0) implies $B_s(x_0,y_0) = 1$. The error between $B_s(x_0,y_0)$ and the current $A_s(x_0,y_0)$ is then diffused with diffusion filter $F_{(0.7813,0.7813\sqrt{2})}$ to update A_s as in the case when we update A_n for $0<n<3$ after making $B_n(x_0,y_0)=1$ in stage two.

As for all other layers, we have $B_k(x_0,y_0)=0$ for $k \in \{3,4,5\ldots 8\}\setminus\{s\}$. Their transient intensity planes A_k are updated by diffusing their individual errors at pixel (x_0,y_0) $[=A_k(x_0,y_0)]$ in their corresponding A_k with a tone-dependent filter. The design of the tone-dependent filter used in this stage is the same as that of the one used in stage two; details are provided in Sec. 4. The update of A_k accordingly leads to an update of E.

```
% Process layers R, G, B, C, M and Y (i.e. I3, I4, ... and I8) without preference
%-------------------------------------------------------------------------------
For n=3,4...8
    BDn = Σ(x,y) In(x,y)                                  % Get dot budget for layer In
End
While Σn=3,4..8 BDn ≥0.5
    E = A3 + A4 + ... + A8                                % Combine all An to form plane E
    Locate a pixel in E via maximum intensity guidance    % Assume the located one is (x0, y0)
    s = arg max(3≤m≤8) Am(x0, y0) under constraint BDm≥0.5   % Determine the color of the dot to be placed
    Bs(x0, y0) = 1                                        % Put a dot of color s in (x0, y0)
    Diffuse error of As(x0, y0) to its neighbors in As    % Use filter F(0.7813,0.7813√2)
    For k ∈ {3:8}\{s}
        Bk(x0, y0) = 0                                    % No dot of other colors in (x0, y0)
        Diffuse error of Ak(x0, y0) to its neighbors in Ak % Use filter (14)
    End
    BDs = BDs − 1                                         % Update dot budget for layer As
    Mα(x0, y0) = 0                                        % Update mask Mα
End
```

Fig. 4 Workflow of stage 3.

After placing a dot of color s, the dot budget of color s is reduced by 1. The above dot positioning and color selecting procedures are repeated until all color dot budgets are used up. At the end, each of the pixels is assigned a dot of specific color and B_n planes are binary halftones of I_n for all n.

3.5 *Details of Stage 4*

At this stage, we have eight binary halftones (i.e., B_n for $0 < n < 9$), each of which defines a dot pattern for a particular layer I_n. In equation form, we have $\sum_{n=1}^{8} B_n(x, y) = 1$ for all pixels (x, y), which implies none of the dot patterns overlap. The final color halftone is produced by directly superimposing all eight dot patterns.

4 Tone-Dependent Diffusion Filter

In both stages two and three, once a dot of color s is placed on pixel (x_0, y_0), all $B_k(x_0, y_0)$ should be updated as

$$B_k(x_0, y_0) = \begin{cases} 1 & \text{if } k = s \\ 0 & \text{else} \end{cases} \quad \text{for } 0 < k < 9 \tag{12}$$

because no dots of other colors should be placed on pixel (x_0, y_0). Accordingly, error diffusion should be carried out in all layers to update A_k for $0 < k < 9$. While diffusion filter $F_{(0.7813,0.7813\sqrt{2})}$ is used to update A_s, a tone-dependent diffusion filter is used to update A_k for $k \neq s$. This arrangement allows for controlling the noise characteristics of the final color halftoning output.

The tone-dependent diffusion filter for updating a specific A_k for $k \neq s$ is defined based on color s, color k, and color β, where color β is the background color of pixel (x_0, y_0) and is selected based on the following criterion:

$$I_\beta(x_0, y_0) \geq I_m(x_0, y_0) \quad \text{for } 0 < m < 9. \tag{13}$$

When there is more than one color that satisfies Eq. (13), the one that dominates the local region is selected. We note that the background color that we refer to here is pixel-oriented. It changes from pixel to pixel.

In our proposal, the diffusion filter for adjusting A_k for $k \neq s$ is defined as

$$F = \begin{cases} F_{[d(x_0,y_0)-1/\sqrt{2},d(x_0,y_0)+1/\sqrt{2}]} & \text{if } s \text{ and } k \neq \beta \\ F_{(1/\sqrt{2},3/\sqrt{2})} & \text{if } s \text{ or } k = \beta \end{cases}, \tag{14}$$

where

$$d(x_0,y_0) = \begin{cases} 1/\sqrt{1 - I_\beta(x_0, y_0)} & \text{if } 1 > I_\beta(x_0, y_0) > 0.5 \\ \sqrt{2} & \text{else} \end{cases}. \tag{15}$$

The rationale for this proposal is as follows. Consider the case where we are color-halftoning a constant color patch whose color is $I(x_0, y_0)$. When MBVC is exploited in the halftoning, the mixture ratio of color dots that appear in the resultant halftoning output should be $I_1(x_0, y_0)$: $I_2(x_0, y_0)$ $\cdots$ $I_8(x_0, y_0)$. Dots in the resultant halftoning output can be divided into two groups. The background group contains dots of background color β, while the foreground group contains dots of other colors. In an ideal halftoning output, dots from the foreground group (referred to as foreground dots hereafter) should be homogenously and uniformly distributed over the halftone with an average distance. Hence, they should maintain a certain distance from each other. It is not necessary, however, for a foreground dot to maintain a certain distance from a background dot (i.e., a dot of color β).

4.1 *Case A: Both Dots of Color s and k Are Foreground Dots [i.e., (s ≠ k) and (k and s ≠ β)]*

According to the blue noise model proposed by Ulichney,[7] in an ideal binary halftone that renders a constant patch of gray level g, dots should distribute homogenously over the halftone with an average distance λ from each other, where λ is defined as

$$\lambda = \begin{cases} 1/\sqrt{g} & \text{for } 0 < g \leq 0.5 \\ 1/\sqrt{1-g} & \text{for } 0.5 < g < 1 \end{cases}. \tag{16}$$

Based on the same idea, when rendering a constant patch of color $I(x_0, y_0)$, the foreground dots should also be homogenously distributed with an average distance $d'(x_0, y_0) = 1/\sqrt{1 - I_\beta(x_0, y_0)}$, where $[1 - I_\beta(x_0, y_0)]$ is the total intensity of all colors other than the background color in a pixel. Here, we assume that $I_\beta(x_0, y_0) \geq 0.5$.

In view of this, a tentative diffusion filter defined as $F_{[d'(x_0,y_0)-1/\sqrt{2},d'(x_0,y_0)+1/\sqrt{2}]}$ is suggested in our proposal for handling case A. This filter is a discrete approximation of a ring-shaped filter whose filter support in the continuous domain is shown in Fig. 5(a). Because we have $B_k(x_0, y_0) = 0$, the quantization error $[=A_k(x_0, y_0) - B_k(x_0, y_0) = A_k(x_0, y_0)]$ will be diffused to the ring region

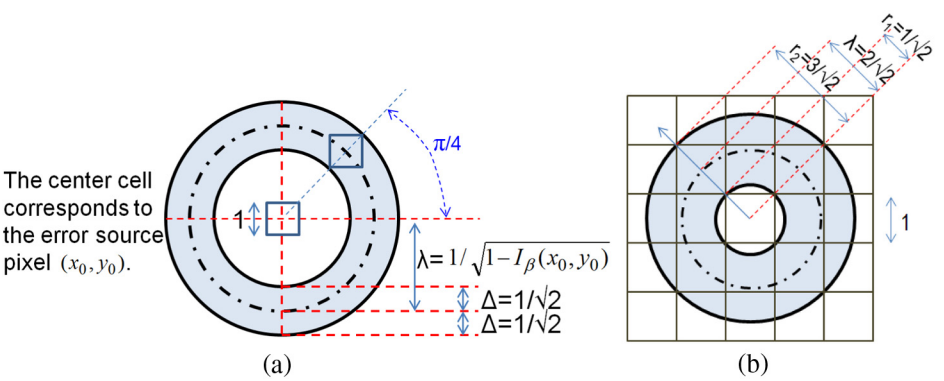


Fig. 5 Filter support of a tone-dependent ring-shaped diffusion filter: (a) general case: $\lambda > \sqrt{2}$ and (b) extreme case: $\lambda = \sqrt{2}$.

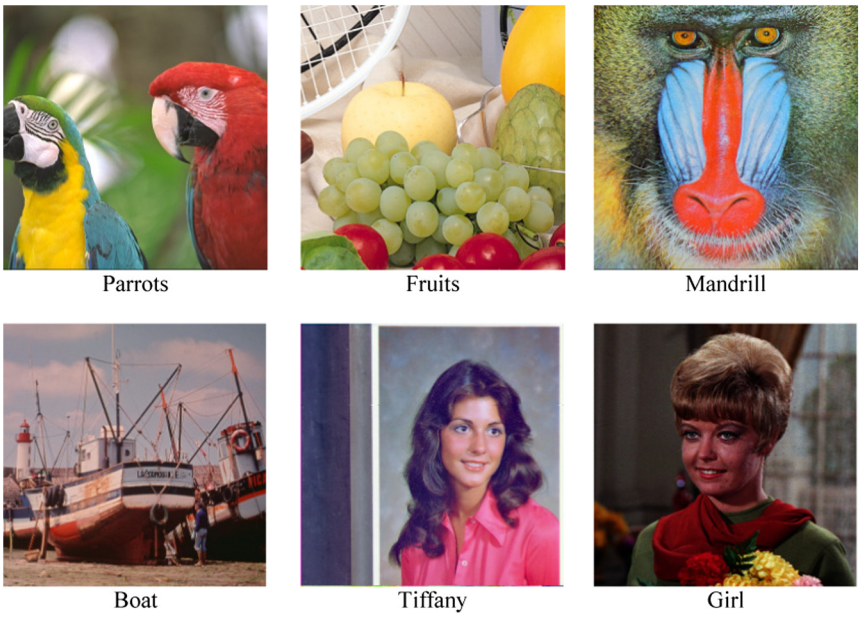


Fig. 6 Testing images.

and increases the potential for having a dot of color k in the ring region in the future. In other words, the diffusion encourages a dot of color k to be at a distance of $d'(x_0, y_0)$ from a dot of color s as long as none of their colors is the background color.

In error diffusion, all errors must be diffused away from the error-source pixel. When $I_\beta(x_0, y_0)$ is <0.5, distance $d'(x_0, y_0)$ is so small that we have $d'(x_0, y_0) - 1/\sqrt{2} < 1/\sqrt{2}$. This makes the filter support of the ring-shaped filter approximated by $F_{[d'(x_0,y_0)-1/\sqrt{2},d'(x_0,y_0)+1/\sqrt{2}]}$ cover the grid cell of (x_0, y_0) and part of the intensity of $A_k(x_0, y_0)$ will be diffused back into pixel (x_0, y_0). To prevent this, the average distance between two foreground dots is bounded by $\sqrt{2}$ as given in Eq. (15) in our proposal such that the minimum inner radius of the filter support of the ring-shaped filter is bounded as shown in Fig. 5(b). In fact, the diffusion filter becomes $F_{(1/\sqrt{2},3/\sqrt{2})}$ at the time when it is bounded.

4.2 *Case B: Dots of Color s or k Are Background Dots [i.e., (s ≠ k) and (k or s = β)]*

When either color s or color k is color β, the quantization error of $A_k(x_0, y_0)$ $[=A_k(x_0, y_0) - B_k(x_0, y_0) = A_k(x_0, y_0)]$ is diffused to the immediate neighborhood of pixel (x_0, y_0) with diffusion filter $F_{(1/\sqrt{2},3/\sqrt{2})}$. As mentioned earlier, $F_{(1/\sqrt{2},3/\sqrt{2})}$ is a discrete approximation of the ring-shaped filter with a minimum inner radius for not diffusing the error back to pixel (x_0, y_0) as shown in Fig. 5(b). Keeping the diffused intensity close to pixel (x_0, y_0) increases the potential of having a dot of color k next to pixel (x_0, y_0),

Table 1 Sparse feature fidelity (SFF) performance of various algorithms.

	SFF					
	Output for printers equipped with CMY cartridges			Output for printers equipped with CMYK cartridges		
Testing image	Proposed	HED (Ref. 28)	HCB-DBS (Ref. 29)	Proposed	HED (Ref. 28)	HCB-DBS (Ref. 29)
Parrot	0.9978	0.9969	0.9972	0.9982	0.9961	0.9932
Fruits	0.9860	0.9864	0.9868	0.9866	0.9877	0.9818
Mandrill	0.9928	0.9910	0.9912	0.9935	0.9891	0.9833
Boat	0.9900	0.9895	0.9873	0.9906	0.9855	0.9773
Tiffany	0.9899	0.9892	0.9891	0.9900	0.9874	0.9838
Girl	0.9965	0.9959	0.9953	0.9970	0.9938	0.9926
Average	0.9922	0.9915	0.9912	0.9926	0.9899	0.9853

HED, hierarchical error diffusion; HCB-DBS, hierarchical colorant based direct binary search.

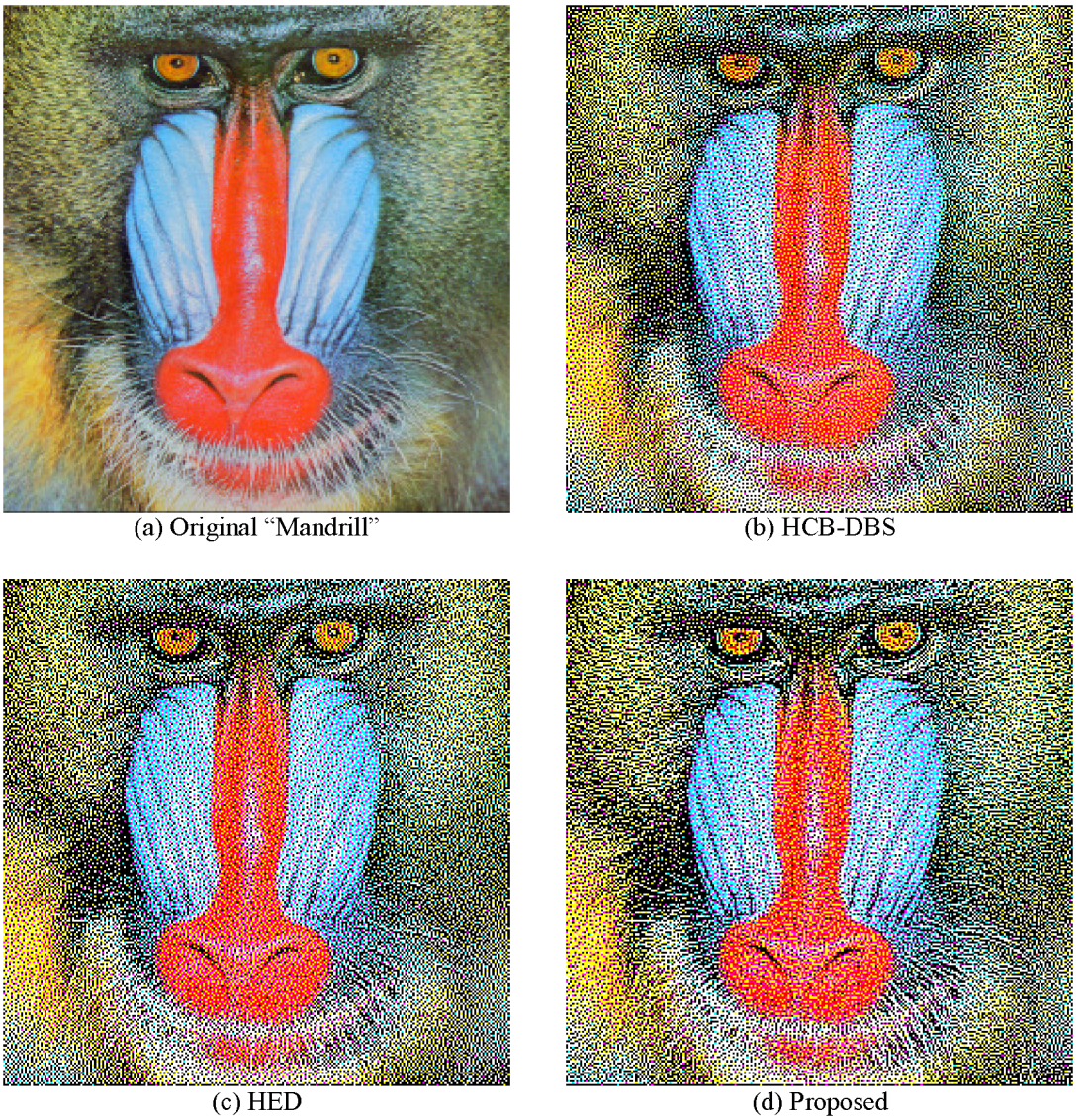


Fig. 7 Color halftones of testing image Mandrill for CMYK printers. (a) Original. (b) Hierarchical colorant based direct binary search (HCB-DBS). (c) HED. (d) Proposed.

which helps to reduce the color shift in the local region and preserve local spatial features.

5 Simulation Results

A simulation was carried out to study the performance of the proposed algorithm. In the simulation, various color error diffusion algorithms, including those of Refs. 28 and 29, and the proposed algorithm were compared. They are compared because all of them adopt a strategy that performs color overlapping control, dot positioning control, and dot coloring control separately and explicitly as suggested in HED.[28]

A set of 24-bit color testing images of size 256×256 as shown in Fig. 6 were used in the simulation. The resultant color halftones were descreened with an human visual system (HVS) filter derived based on Campbell's contrast sensitivity function model,[38] and the perceptual quality of the descreened color halftones was evaluated with the sparse feature fidelity (SFF) proposed in Ref. 39. The HVS filter was derived based on the condition that the printer resolution is 600 dpi and the viewing distance is 15 in. Table 1 shows the evaluation results. We note that a larger value indicates a better performance. One can see that the proposed algorithm is better than the others in terms of this measure.

For subjective evaluation, Figs. 7 and 8 show, respectively, the color halftoning outputs of testing images Mandrill and Boat. As shown in the figures, the color halftones produced by the proposed algorithm can preserve the spatial features of the original images very well. For example, it is able to show the eyes, the lower eyelids, and the whiskers of the mandrill more clearly in Fig. 7(d). It is also able to show the letters on the stern, the masts, and the poles more clearly in Fig. 8(d). The antenna attached to the main mast of the boat [in the middle top of Fig. 8(a)] is missing in Figs. 8(b) and 8(c), while it is clearly shown in Fig. 8(d).

Note that MBVC is exploited in all the evaluated algorithms to carry out color overlapping control, and hence, the mixture ratios of the color dots in their halftoning outputs are all the same. By placing the right color dots on the right

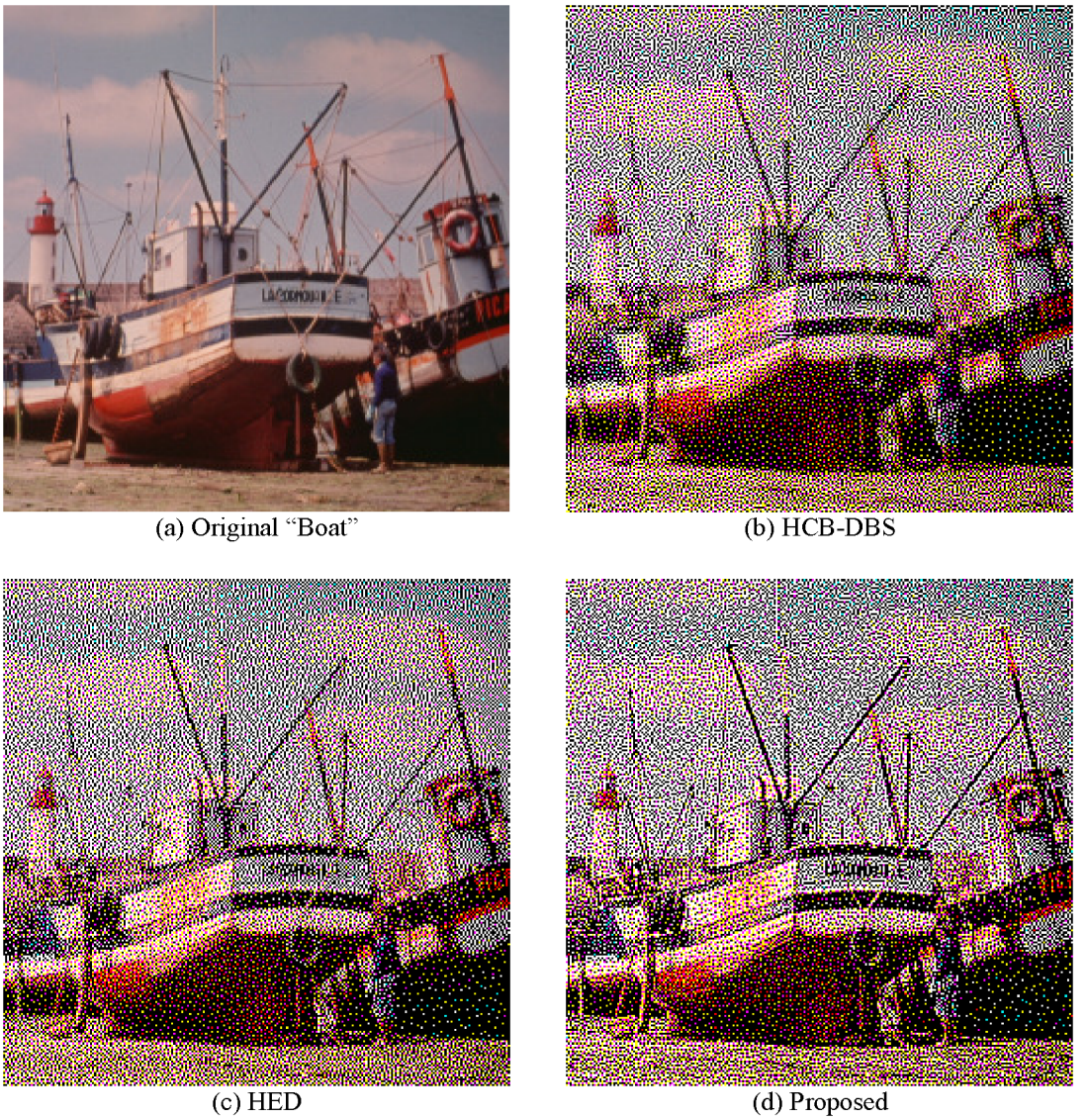


Fig. 8 Color halftones of testing image Boat for CMYK printers. (a) Original. (b) HCB-DBS. (c) HED. (d) Proposed.

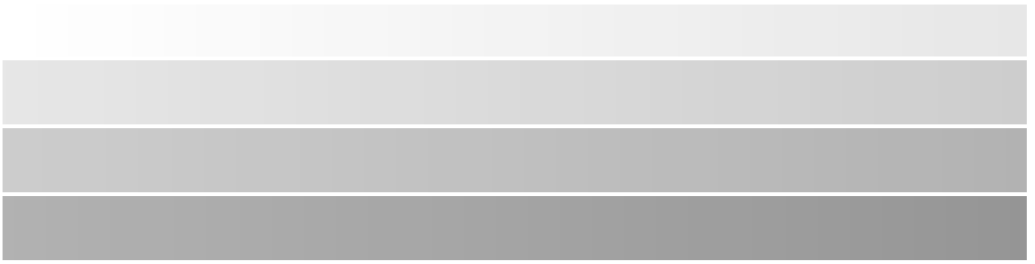
positions as shown in Figs. 7(d) and 8(d), detailed spatial features can be preserved.

Figure 9 shows the color halftoning results for a gray ramp image. Theoretically, DBS is optimized based on an HVS-based cost function. Hence, HCB-DBS should provide the smoothest rendering output. However, from Fig. 9(b), one can see that the yellow color is nonuniformly distributed patch-wise. This is due to the fact that, in HCB-DBS, color channels are grouped and processed in turns. As suggested by He in Ref. 29, in our simulation, color Y has the lowest priority and is processed last due to its low visibility in the highlight area. Yellow dots cannot be placed on positions that have been occupied and, hence, experience more constraint than dots of other colors. In practice, some other color grouping strategies can be exploited as mentioned in Ref. 29. However, the color of the lowest priority is always the one that suffers.

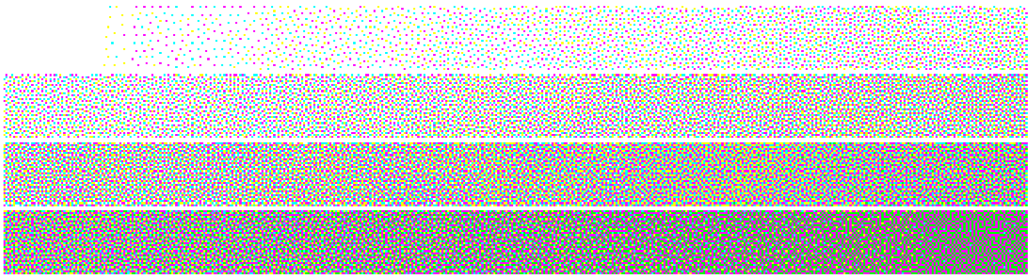
Though HED also introduces a bias in dot coloring control as mentioned in Sec. 2, from the smooth gradation shown in Fig. 9(c), it appears that no specific channel suffers from this disadvantage. This may be due to the fact that PDSV is pixel-dependent and, hence, the bias is adaptive to the local content and allows a good choice for the dot color of a pixel. However, the bias in the dot positioning control caused by the error diffusion framework used in HED introduces some directional and pattern artifacts to the color halftoning output. One can see the artifacts in a region on the right of the middle of the third row and the right-most region of the fourth row in Fig. 9(c).

As shown in Fig. 9(d), the proposed algorithm can also provide a smooth gradation in the rendered results of the ramp image. The rendered result is a bit grainier than that of HED, but it does not contain any directional and pattern artifacts as Fig. 9(c) does.

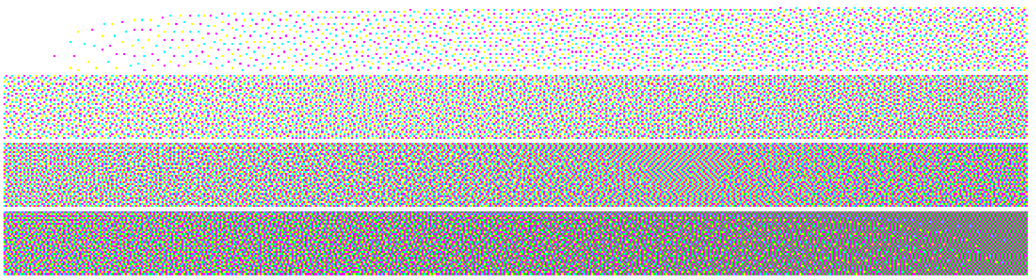
A graininess measure suggested in ISO15739 (Ref. 40) was used to evaluate the graininess of the outputs of different algorithms. The evaluation is based on the condition that the printer resolution is 600 dpi and the viewing distance is 15 in. Three constant gray color patches are halftoned with different algorithms to produce CMY outputs. The selected



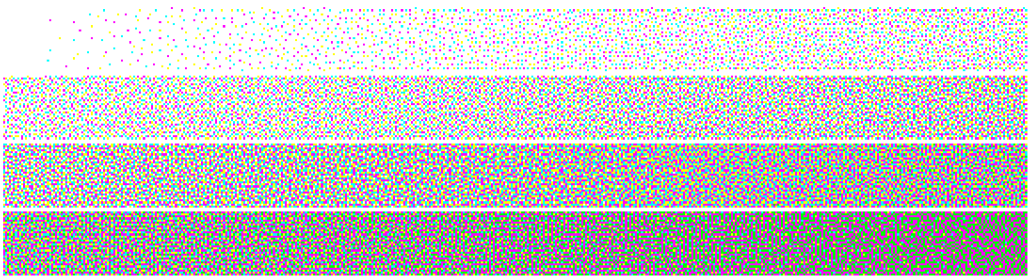
(a) Original "Ramp"



(b) HCB-DBS



(c) HED



(d) Proposed

Fig. 9 Color halftones of a gray ramp image for CMY printers. (a) Original. (b) HCB-DBS. (c) HED. (d) Proposed.

Table 2 Graininess measures (ISO15739) of the outputs produced by various algorithms.

	Gray level		
Algorithms	6/255	64/255	127/255
Proposed	13.4497	5.3966	2.8706
HED	14.0090	2.9593	1.8988
HCB-DBS	18.2073	4.4954	2.6531

gray levels of the patches are representatives of the low, the middle, and the high levels in the range of (0, 0.5). Table 2 shows the simulation results of the study. As shown in the table, the output of the proposed algorithm is a bit noisier in the middle range than the others.

It is well known in the field of error diffusion halftoning that some error diffusion filters can sharpen an image more than others.[41] As the outputs of the proposed algorithm appear to be sharper than the others, one would be interested in knowing whether the spatial features are actually sharpened by the diffusion filter exploited in the proposed algorithm. To address this issue, we carried out a study based on the linear gain model of the quantizer suggested in Ref. 41.

Table 3 The computed linear signal gain K_s for various testing images and color channels when different algorithms are used.

		The K_s of each color layer							
Image	Algorithm	K	B	R	G	M	C	Y	W
Parrot	HED	1.3183	—	1.0103	1.0202	1.1307	1.0352	1.1553	1.8189
	Proposed	0.9215	—	1.0542	1.0037	1.1166	1.0641	1.2970	0.9854
Fruits	HED	1.1877	—	1.0025	1.0007	1.1170	1.0621	1.1860	1.7982
	Proposed	0.9232	—	1.0189	1.0002	1.1410	1.0215	1.2786	0.9415
Mandrill	HED	1.2691	1.0046	1.0012	—	1.1085	1.0846	1.1039	1.8199
	Proposed	0.9210	1.0002	1.0202	—	1.1304	1.1239	1.1968	0.9766
Boat	HED	1.3407	—	1.0058	—	1.0770	1.1182	1.1144	1.9944
	Proposed	0.9589	—	1.0214	—	1.2657	1.0164	1.2047	0.9947
Tiffany	HED	1.3253	1.0343	1.0068	1.0001	1.1197	1.0753	1.0679	1.9827
	Proposed	0.9221	1.0102	1.0168	1.0002	1.3003	1.1550	1.0527	0.9374
Girl	HED	1.7630	—	1.0435	1.0027	1.0832	1.0640	1.1115	1.8985
	Proposed	0.9816	—	1.0562	1.0010	1.1384	1.0155	1.1854	1.0277
Average	**HED**	**1.3674**	**1.0195**	**1.0117**	**1.0059**	**1.1060**	**1.0732**	**1.1232**	**1.8854**
	Proposed	**0.9381**	**1.0052**	**1.0313**	**1.0013**	**1.1821**	**1.0661**	**1.2025**	**0.9772**

Table 4 The distribution of color dots in the outputs of different testing images.

	Number of dots in different channels							
Image	K	B	R	G	M	C	Y	W
Parrot	27,578	0	3621	73	3841	2502	12,692	15,229
Fruits	17,554	0	2060	1	4042	405	15,612	25,862
Mandrill	25,024	1	2680	0	4810	5297	8856	18,868
Boat	30,391	0	732	0	5879	276	6350	21,908
Tiffany	28,360	208	893	9	10,563	3749	4097	17,657
Girl	45,772	0	2952	9	3304	267	4902	8330

According to the model, the sharpening effect of an error diffusion based algorithm can be reflected by the linear signal gain of its quantizer. In equation form, the linear signal gain is defined as

$$K_s = \frac{\sum_{i,j} x'(i,j) y(i,j)}{\sum_{i,j} x'(i,j)^2}, \tag{17}$$

where $x'(i,j)$ is the input to the quantizer for pixel (i,j) and $y(i,j) \in \{-0.5, 0.5\}$ is its output. Note that in the model, the color components of all pixels of the original color image are normalized such that their intensity values are bounded in $[-0.5, 0.5]$. No sharpening is introduced by the error diffusion algorithm when $K_s = 1$.

Table 3 shows the K_s values of different color channels of the outputs obtained with different algorithms. As the

Fig. 10 A compensated HED output that suffers the same amount of sharpening as Fig. 8(d).

working principle of HCB-DBS is different from that of error diffusion based algorithms, the model does not apply and hence HCB-DBS is excluded in the study. When comparing HED and the proposed algorithm, one can see that HED is actually the one that sharpens the images more. Its K_s values are much further away from 1 for channels K and W. As shown in Table 4, channels K and W are the dominant colors in the outputs. Though the proposed algorithm sharpens the image a bit more in channels Y and M, the difference is not significant and they are not the dominant channels. As for the other channels, the performance of the two algorithms is more or less the same.

For visual comparison, based on the K_s values extracted from the outputs of Boat, we adjusted the extent of sharpening by prefiltering the input image as suggested in Ref. 41 to produce an HED output that suffers the same amount of sharpening as Fig. 8(d). The result is shown in Fig. 10. It is not as sharp as Fig. 8(d).

The simulation results of the study reveal that the feature preserving capability of the proposed algorithm does not rely on the sharpening effect of its error diffusion filters. Instead, it mainly relies on its accuracy and flexibility in dot positioning and coloring.

6 Conclusions

A recent trend for improving the output quality of color halftoning is to explicitly control color overlapping, dot positioning, and dot coloring in different stages of the halftoning process. In Ref. 28, an effective scheme for controlling color overlapping, called MBVC, is proposed, in which a color is decomposed into an appropriate mixture of limited Neugebauer primaries such that no overlap of these Neugebauer primaries is required when the color is rendered. As the mixture of these Neugebauer primaries can also provide minimum brightness variation color density in a smooth region, it naturally becomes a useful means of color overlapping control.

As FMED performs well in dot positioning control in both binary halftoning[30,32–37] and multilevel halftoning,[42,43] we would naturally like to explore whether or not it can be used for dot positioning and coloring control in color halftoning. This paper reports some results of our study on this issue. In particular, an FMED-based color halftoning algorithm is proposed in this paper. Compared with other algorithms that explicitly control color overlapping, dot positioning, and dot coloring, the proposed algorithm does not assign a fixed priority order to Neugebauer primaries in dot coloring and does not place dots along a fixed scanning path in dot positioning. Consequently, it provides more flexibility in dot positioning and coloring and, hence, is able to better preserve image features.

Simulation results show that the proposed color halftoning algorithm can produce a high-quality color halftone compared with the others in terms of SFF. The proposed algorithm is particularly good at preserving spatial features and is able to remove pattern artifacts and directional artifacts.

Acknowledgments

The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No: PolyU5120/13E).

References

1. C. Haines, S. Wang, and K. Knox, "Digital color halftones," Chapter 6 in *Digital Color Imaging Handbook*, G. Sharma, Ed., pp. 457–462, CRC, Boca Raton, FL (2003).
2. F. A. Baqai et al., "Digital color halftoning," *IEEE Signal Process. Mag.* **22**(1), 87–96 (2005).
3. N. Damera-Venkata, B. L. Evans, and V. Monga, "Color error-diffusion halftoning," *IEEE Signal Process. Mag.* **20**(4), 51–58 (2003).
4. B. E. Bayer, "An optimal method for two-level rendition of continuous-tone pictures," in *Proc. IEEE Int. Conf. on Communication*, Vol. 1, pp. 11–15 (1973).
5. T. Mitsa and K. Parker, "Digital halftoning using a blue noise mask," *Proc. SPIE* **1452**, 47–56 (1991).
6. J. Allebach and Q. Lin, "FM screen design using DBS algorithm," in *Proc. IEEE Int. Conf. on Image Processing*, Vol. 1, pp. 549–552 (1996).
7. R. A. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA (1987).
8. R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey-scale," *Proc. S.I.D.* **17**(2), 75–77 (1976).
9. R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE* **76**, 56–79 (1988).
10. B. Kolpatzik and C. A. Bouman, "Optimized error diffusion for image display," *J. Electron. Imaging* **1**(3), 277–292 (1992).
11. M. Analoui and J. P. Allebach, "Model-based halftoning using direct binary search," *Proc. SPIE* **1666**, 96–108 (1992).
12. T. N. Pappas and D. L. Neuhoff, "Least-squares model-based halftoning," *Proc. SPIE* **1666**, 165–176 (1992).
13. H. Haneishi et al., "Color digital halftoning taking colorimetric color reproduction into account," *J. Electron. Imaging* **5**(1), 97–106 (1996).
14. Z. Fan and S. Harrington, "Improved quantization methods in color error diffusion," *J. Electron. Imaging* **8**(4), 430–437 (1999).
15. L. Akarun, Y. Yardimci, and A. Cetin, "Adaptive methods for dithering color images," *IEEE Trans. Image Process.* **6**(7), 950–955 (1997).
16. N. Damera-Venkata and B. L. Evans, "Design and analysis of vector color error diffusion halftoning systems," *IEEE Trans. Image Process.* **10**(10), 1552–1565 (2001).
17. M. Monga, N. Damera-Venkata, and B. L. Evans, "Design of tone-dependent color error diffusion system," *IEEE Trans. Image Process.* **16**(1), 198–211 (2007).
18. Z. Fan, "Stability analysis for color error diffusion," *Proc. SPIE* **3963**, 483–488 (2000).
19. D. Shaked et al., "Color diffusion: error diffusion for color halftones," Tech. Rep., HPL-96-128R1, HP Labs Israel (1996).
20. R. V. Klassen and R. Eschbach, "Vector error diffusion in a distorted color space," in *Proc. IS&T Conf. 47th Annual Conf.*, pp. 489–491 (1994).
21. J. Shu and J. Boyce, "Adaptive color error diffusion to improve halftone smoothness," *Proc. SPIE* **3018**, 308–315 (1997).
22. R. V. Klassen and R. Eschbach, "Vector error diffusion in a distorted color space," in *Proc. IS&T 47th Annual Conf.*, pp. 489–491 (1994).
23. Z. Fan, "Error diffusion for CMYK color images," in *Image Processing, Image Quality, Image Capture, Systems Conf.*, pp. 324–326 (1999).

24. T. N. Pappas, "Model-based halftoning of color images," *IEEE Trans. Image Process.* **6**(7), 1014–1024 (1997).
25. T. J. Flohr et al., "Model-based color image quantization," *Proc. SPIE* **1913**, 270–281 (1993).
26. A. U. Agar and J. P. Allebach, "Model-based color halftoning using direct binary search color imaging: device-independent color, color hardcopy, and graphic arts V," *Proc. SPIE* **3963**, 521–535 (2000).
27. J. Lee and J. P. Allebach, "Colorant-based direct binary search halftoning," *J. Electron. Imaging* **11**, 517–527 (2002).
28. Z. He, "Hierarchical error diffusion," *IEEE Trans. Image Process.* **18**(7), 1524–1535 (2009).
29. Z. He, "Hierarchical colorant-based direct binary search halftoning," *IEEE Trans. Image Process.* **19**(7), 1824–1836 (2010).
30. Y. H. Chan and S. M. Cheung, "Feature-preserving multiscale error diffusion for digital halftoning," *J. Electron. Imaging* **13**(3), 639–645 (2004).
31. I. Katsavounidis and C. C. J. Kuo, "A multiscale error diffusion technique for digital halftoning," *IEEE Trans. Image Process.* **6**(3), 483–490 (1997).
32. Y. H. Fung, K. C. Lui, and Y. H. Chan, "Low-complexity high-performance multiscale error diffusion technique for digital halftoning," *J. Electron. Imaging* **16**(1), 1–12 (2007).
33. Y. H. Fung and Y. H. Chan, "Optimizing the error diffusion filter for blue noise halftoning with multiscale error diffusion," *IEEE Trans. Image Process.* **22**(1), 413–417 (2013).
34. Y. H. Chan, "A modified multiscale error diffusion technique for digital halftoning," *IEEE Signal Process. Lett.* **5**(11), 277–280 (1998).
35. Y. H. Fung and Y. H. Chan, "Embedding halftones of different resolutions in a full-scale halftone," *IEEE Signal Process. Lett.* **13**(3), 153–156 (2006).
36. Y. H. Fung and Y. H. Chan, "Green noise digital halftoning with multiscale error diffusion," *IEEE Trans. Image Process.* **19**(7), 1808–1823 (2010).
37. Y. H. Fung and Y. H. Chan, "Tone-dependent noise model for high-quality halftones," *J. Electron. Imaging* **22**(2), 023004 (2013).
38. F. W. Campbell, R. H. Carpenter, and J. Levinson, "Visibility of aperiodic patterns compared with that of sinusoidal gratings," *J. Physiol.* **204**(2), 283–298 (1969).
39. H.-w. Chang et al., "Sparse feature fidelity for perceptual image quality assessment," *IEEE Trans. Image Process.* **22**(10), 4007–4018 (2013).
40. ISO 15739:2013, Photography—Electronic still-picture imaging—Noise measurements, 2nd ed., ICS: 37.040.99, TC/SC: ISO/TC 42, p. 31.
41. T. D. Kite, B. L. Evans, and A. C. Bovik, "Modeling and quality assessment of halftoning by error diffusion," *IEEE Trans. Image Process.* **9**, 909–922 (2000).
42. Y. H. Fung and Y. H. Chan, "Multilevel halftoning using multiscale error diffusion," *J. Electron. Imaging* **19**, 030501 (2010).
43. L. Y. Wong and Y. H. Chan, "A feature preserving multilevel halftoning algorithm," *J. Electron. Imaging* **21**(4), 043016 (2012).

Yik-Hing Fung received the BEng (Hons.) degree and PhD degree from the Hong Kong Polytechnic University (HKPU), Kowloon, Hong Kong, in 2000 and 2006, respectively. From 2000 to 2001, he was an engineer with HKPU for a project entitled Web Based Portable Monitoring and Surveillance System. He was with ASM Pacific Technology Ltd. as an R&D engineer working for computer vision in 2006. Between 2007 and 2008, he was with Department of Computer Science, Hong Kong Baptist University as a postdoctoral teaching fellow. In 2008, he was with AppoTech Ltd. as an IC design engineer working for IC design in multimedia applications. He is currently a research fellow in the Department of Electronic and Information Engineering, HKPU. His research interests include digital halftoning, computer vision, color image restoration and compression.

Yuk-Hee Chan received his BSc degree with honors in electronics from Chinese University of Hong Kong in 1987, and his PhD degree in signal processing from The Hong Kong Polytechnic University in 1992. Between 1987 and 1989, he worked as an R&D engineer at Elec & Eltek Group, Hong Kong. He joined this University in 1992 and is now an associate professor in the Department of Electronic & Information Engineering. He has published over 145 research papers in various international journals and conferences. His research interests include image and video compression, image restoration, halftoning, demosaicking, and fast computational algorithms in digital signal processing. He was the chairman of the IEEE Hong Kong Joint Chapter of CAS and COM in 2003 to 2004 and is now the Chair-Elect of the IEEE Hong Kong Section.