Model-Based Color Halftoning Using Direct Binary Search

A. Ufuk Agar, *Member, IEEE,* and Jan P. Allebach, *Fellow, IEEE*

***Abstract*—In this paper, we develop a model-based color halftoning method using the direct binary search (DBS) algorithm. Our method strives to minimize the perceived error between the continuous tone original color image and the color halftone image. We exploit the differences in how the human viewers respond to luminance and chrominance information and use the total squared error in a luminance/chrominance based space as our metric. Starting with an initial halftone, we minimize this error metric using the DBS algorithm. Our method also incorporates a measurement based color printer dot interaction model to prevent the artifacts due to dot overlap and to improve color texture quality. We calibrate our halftoning algorithm to ensure accurate colorant distributions in resulting halftones. We present the color halftones which demonstrate the efficacy of our method.**

***Index Terms*—Color printer, color printer calibration, dot overlap, halftoning, human visual system (HVS) model, printer gamut, printer model, tetrahedral interpolation.**

I. Introduction

DIGITAL color halftoning is the process of transforming continuous-tone color images into images with a limited number of colors. The importance of this process arises from the fact that many color imaging systems contain output devices such as color printers and low-bit depth displays that are bilevel or multilevel with a few levels. The goal is to create the perception of a continuous-tone color image using the limited color discrimination capability and the low-pass characteristics of the spatial sensitivity of the human visual system (HVS).

The algorithms for digital monochrome halftoning can be categorized into three classes, in decreasing order of how myopically they transform a given image into a halftone and, thus, in increasing order of computational complexity and halftone quality. In the first class are the pointwise approaches like screening or dithering where the image is pixelwise compared to an array of thresholds and the pixels exceeding this spatially varying mask of thresholds are changed to black. The next class of approaches uses the information about a neighborhood of pixels to decide the halftone state of a given pixel. Error diffusion is the most important example of this category. The third class consists of iterative approaches where several passes over the image pixels are made to minimize an error metric or satisfy certain constraints before the halftoning process is completed. All three of these classes of algorithms can be generalized to digital color halftoning with some modifications, as color halftoning can be viewed as an L-D generalization of the monochrome process where L depends on the number of primaries that the rendering device possesses. For example, L is equal to 3 for cyan–magenta–yellow (CMY) printers and 4 for cyan–magenta–yellow-black (CMYK) printers.

The naive digital color halftoning approach is to apply these monochrome halftoning techniques scalarly and independently to these colors [red, green, blue (RGB)] or colorant planes (CMYK). As expected, this scalar approach leads to color artifacts and poor color rendition because it does not exploit the correlation between colors or colorant planes which is a key element in our color perception and appreciation of the halftone quality. In screening, approaches used to prevent these artifacts include changing the angle of rotation of the clustered dot screens [1]–[4], employing uncorrelated screens for different color planes [5], designing the AM-FM screens jointly by using a special case of linear filters [6], and jointly designing the dither matrices to minimize the visibility of moiré and rosette artifacts [7], to minimize color or brightness fluctuation [8], [9] or to create halftones with a desired frequency content such as blue noise [10] and green noise [11], [12]. Error diffusion has been more successfully applied to color halftoning using the generalized version, vector error diffusion where the error due to halftoning at a given pixel is diffused to color planes jointly or the error diffusion is carried out in a device independent space [12]–[21].

The iterative monochrome halftoning methods that minimize an error metric based on a human visual model and, hence, named *model-based halftoning methods* have also been generalized to color. Pappas [22] proposed a least squares model-based color halftoning algorithm which incorporates a printer dot model and employs a two-dimensional (2-D) filter to model the HVS. His algorithm uses the same filter and minimizes the least squares error metric independently for each color component. Flohr *et al.* [23] exploited the differences in how the viewer responds to luminance and chrominance information, and used the total squared error in a luminance/chrominance-based space as the metric for their model-based color halftoning algorithm. They minimized this error metric using the method of iterated conditional modes. In this paper, we also take a model-based approach to color halftoning. We employ the HVS model and the error metric chosen by Flohr *et al.* [23], but we apply the direct binary search algorithm [24] to minimize this metric.

Manuscript received August 20, 2001; revised January 14, 2004. This work was supported by the Hewlett-Packard Company. The associate editor coordinating the review of the manuscript and approving it for publication was Dr. Zhigang Fan.

A. U. Agar was with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA. She is now with Garanti Technology, Istanbul, Turkey (e-mail: ufuka@garanti.com.tr; ayse_ufuk_agar@alumni.purdue.edu).

J. P. Allebach is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2035 47907-1285 USA (e-mail: allebach@ecn.purdue.edu).

Digital Object Identifier 10.1109/TIP.2005.859380

We also incorporate a dot interaction model [25], [26] into the algorithm to minimize the artifacts due to dot overlap and to improve color texture quality. Finally, we propose a halftoning calibration method to ensure accurate colorant distributions in the resulting halftones. Our model-based halftoning algorithm provides a means of creating color halftones with visually pleasing color textures due to the merit of the incorporated HVS model and the dot interaction model. Since we minimize the total squared error in a device independent luminance/chrominance-based space, the aim of our method is colorimetric reproduction. The calibration enables the use of our algorithm in applications where preserving average colorant in a local area in the image is important, as well as colorimetric reproduction and texture quality. In Section II, we present the color space and the HVS model we use. In Section III, we define the error metric and discuss the iterative search-based algorithm called color direct binary search (CDBS) for minimizing this error metric. We explain the dot interaction model we use in Section IV and show how this model is incorporated into the CDBS algorithm. In Section V, we discuss the calibration of our halftoning algorithm. We present our results in Section VI. Finally, we summarize our conclusions in Section VII.

II. Human Visual System Model

The HVS model that Flohr *et al.* [23] employed, which we will refer to as the Flohr HVS model, is based upon the uniform color space CIE $L^*a^*b^*$, which is widely used in industry to measure color reproduction errors. L^* is a correlate of lightness. The radial distance and the angular position in the a^*b^* plane are correlates of chroma and hue, respectively. The CIE $L^*a^*b^*$ space is defined in terms of CIE XYZ tristimulus values as

$$\begin{aligned} L^* &= 116 f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 200\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \\ b^* &= 500\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \end{aligned} \tag{1}$$

where X_n, Y_n, and Z_n are the tristimulus values of the reference stimulus, and $f(\cdot)$ is defined by

$$f(x) = \begin{cases} 7.787x + \frac{16}{116}, & 0 \le x \le 0.008\,856 \\ x^{1/3}, & 0.008\,856 < x \le 1. \end{cases} \tag{2}$$

This nonlinear transformation from CIE XYZ tristimulus values to CIE $L^*a^*b^*$ values cascaded with the linear transformation from RGB image coordinates to CIE XYZ tristimulus values does not preserve the spatial averaged tones of images [27], which is undesirable in halftone color reproduction. Therefore, Flohr *et al.* [23] linearized the CIE $L^*a^*b^*$ color space about the reference stimulus (X_n, Y_n, Z_n).They denoted this linearized color space by $Y_yC_xC_z$, and defined it as follows:

$$\begin{aligned} Y_y &= 116\frac{Y}{Y_n} \\ C_x &= 200\left[\frac{X}{X_n} - \frac{Y}{Y_n}\right] \\ C_z &= 500\left[\frac{Y}{Y_n} - \frac{Z}{Z_n}\right]. \end{aligned} \tag{3}$$

The Y_y component serves as a correlate of luminance; and the C_x and C_z components are similar to the red–green (R–G) and blue–yellow (B–Y) opponent color chrominance components that Mullen studied [28]. Modeling the intermediate output of the HVS as outputs of three channels, one achromatic and two opponent color chromatic channels is common to many of the color appearance models and color vision models [29], [30]. In these models, the outputs of these channels can then be combined to determine perceptual attributes like hue, saturation, and brightness. Psychophysical studies prove that color discrimination and appearance closely depend on spatial pattern in addition to the global differences in perceptual attributes like hue, saturation, and brightness [31], [32]. This finding motivates the use of HVS models for design and evaluation of the performance of digital color halftoning algorithms. The HVS models are usually presented as three linear, separable and shift-invariant filters in a three-dimensional (3-D) color space: one filter for the achromatic channel (component/dimension) and two filters for the opponent color chromatic channels (components/dimensions). Monga *et al.* [20] evaluated $Y_yC_xC_z$ and three other 3-D color spaces (YIQ and YUV, commonly used in video processing as well as the opponent color space used by Poirson and Wandell [33]) and found $Y_yC_xC_z$ and the filters of the Flohr HVS model to give the best results for vector error diffusion.

In this work, we use the Flohr HVS model to design an efficient color halftoning algorithm. In this model, the luminance channel response is based on the contrast sensitivity function proposed by Nasanen [34]. Kim and Allebach [35] compared Nasanen's model for the luminance channel response with three other models and found that Nasanen's model yields the best overall halftone quality when used with the DBS algorithm. The luminance spatial frequency response based on Nasanen's model $H_{Y_y}(\bar{u}, \bar{v})$ is given by

$$H_{Y_y}(\bar{u}, \bar{v}) = aL^b \exp\left(-\frac{(180)\sqrt{\bar{u}^2 + \bar{v}^2}}{\pi[c\ln(L) + d]}\right) \tag{4}$$

where $a = 131.6$, $b = 0.3188$, $c = 0.525$, $d = 3.91$, L is the average luminance of the light reflected from the print in cd/m^2, and $(\bar{u}, \bar{v})$ are the spatial frequency coordinates in cycles/radian subtended at the retina.

The chrominance spatial frequency response of the Flohr HVS model is based on an approximation by Kolpatzik and Bouman [36] to experimental data collected by Mullen [28]. This chrominance frequency response denoted by $H_{C_xC_z}(\bar{u}, \bar{v})$ is common to both of the opponent color chrominance components and is given by

$$H_{C_xC_z}(\bar{u}, \bar{v}) = A \exp\left(-\alpha\sqrt{\bar{u}^2 + \bar{v}^2}\right) \tag{5}$$

where $\alpha = 0.419$, $A = 100$, and $(\bar{u}, \bar{v})$ are again the spatial frequency coordinates in cycles/radian subtended at the retina.

To enable the adjustment of the relative importance between the luminance and chrominance errors in the error metric, the luminance frequency response is multiplied by a weighting factor κ. As κ increases, more of the error is placed into the chrominance components. Flohr *et al.* [23] showed that as κ exceeds unity, the textures in the resulting halftones shift from being

contrasty and monochromatic to being smoother and multicolored and found that $\kappa = 4$ gives visually pleasing results. The same weighting factor was also employed by Balasubramanian *et al.* [37] to form a visually weighted error metric in quantization of color images in a luminance/chrominance-based color space. The Flohr HVS model was previously used by Baqai and Allebach [38] for designing clustered dot color screens and by Monga *et al.* [39], [40] for tone dependent color error diffusion.

Taking the 2-D inverse continuous space Fourier transforms of $H_{Y_y}(\bar{u},\bar{v})$ and $H_{C_xC_z}(\bar{u},\bar{v})$ gives the impulse response functions $h_{Y_y}(\bar{x},\bar{y})$ and $h_{C_xC_z}(\bar{x},\bar{y})$, where the spatial coordinates $(\bar{x},\bar{y})$ have units of radians subtended at the retina. For a printed page viewed from a distance D *in*, a length x *in* on the page corresponds to $\tan^{-1}(x/D)$ radians subtended at the retina; and $\tan^{-1}(x/D) \approx x/D$ radians, for $x \ll D$. Substituting this approximation into $h_{Y_y}(\bar{x},\bar{y})$ and $h_{C_xC_z}(\bar{x},\bar{y})$, we obtain the HVS point spread functions $\tilde{p}_i(\mathbf{x})$ for $i = Y_y, C_x, C_z$, where $\mathbf{x}$ now has units *in* measured on the print

$$\tilde{p}_{Y_y}(\mathbf{x}) = \frac{1}{D^2} h_{Y_y}\left(\frac{\mathbf{x}}{D}\right) = \mathcal{F}^{-1}\{H_{Y_y}(D\bar{u}, D\bar{v})\} \tag{6}$$

$$\begin{aligned}\tilde{p}_{C_x}(\mathbf{x}) = \tilde{p}_{C_z}(\mathbf{x}) &= \frac{1}{D^2} h_{C_xC_z}\left(\frac{\mathbf{x}}{D}\right) \\ &= \mathcal{F}^{-1}\{H_{C_xC_z}(D\bar{u}, D\bar{v})\}.\end{aligned} \tag{7}$$

As the viewing distance D increases, the supports of the HVS point spread functions increase, i.e., a larger surrounding area affects our perception of a given point on the printed page.

III. Color Direct Binary Search

A. Preliminaries

We use $[\mathbf{m}] = [m,n]^T$ and $(\mathbf{x}) = (x,y)^T$ to denote discrete and continuous spatial coordinates in an image, respectively. We denote a continuous-tone color image by a 3-D vector valued function $\mathbf{f}[\mathbf{m}] : \Re^2 \mapsto \Re^3$ where the input values are the discrete spatial coordinates in the image and the output values are the RGB values at a given location in the image, ranging between 0 and 1. This function can be decomposed into 3 scalar valued functions $f_i : \Re^2 \mapsto \Re$, $i =$ R,G,B. Similarly, we denote a color halftone bitmap by a vector valued L-D function $\mathbf{g}[\mathbf{m}] : \Re^2 \mapsto \Re^L$ where the input values are again the discrete spatial coordinates in the image and L is the number of colorants that the color halftone contains. This function can again be decomposed into L scalar valued functions. We restrict our analysis to bilevel three-colorant CMY printers and bilevel four-colorant CMYK printers where full undercolor removal is employed, i.e., at a given pixel (printer addressable point) a black dot replaces cyan, magenta, and yellow dots if, and only if, all three are present there. Therefore, we assume that $L = 3$.

For the analysis in this section, we also assume that the dot interactions between neighboring pixels in the printer grid are additive. This assumption is generally not true; the interaction of the colorants of the neighboring printer pixels with each other and the paper substrate is fairly complex. We will discuss the approaches to modeling these dot interactions and show how we incorporate one such model into our halftoning algorithm in Section IV.

B. Color Direct Binary Search

Our proposed CDBS halftoning method is an iterative search-based algorithm that minimizes a measure of perceived error by starting from an initial halftone and modifying the halftone locally. The initial halftone is obtained by halftoning the RGB components of the continuous-tone color image separately using the same screen which is obtained with the monochrome DBS halftoning method [24], [41] and then superimposing these three halftones under the assumption that $C = 1 - R$, $M = 1 - G$, $Y = 1 - B$. For the initial halftone, we compute the perceived error in the device independent opponent color space of $Y_yC_xC_z$ as suggested by Flohr *et al.* [23] and discussed below. Then we process the pixels of the halftone in a raster-scan order. At each pixel, we compute the effect on the error metric of toggling (changing the colorant combination) of the pixel and swapping it with its eight nearest neighbor (NN) pixels. We accept that trial change, if any, which mostly reduces the error metric, and we iterate until there are no more accepted trial changes, i.e., until the error metric reaches a local minimum.

To compute the perceived error, we first need to convert the original continuous-tone color image $\mathbf{f}[\mathbf{m}]$ and the color halftone bitmap $\mathbf{g}[\mathbf{m}]$ from the device dependent color spaces of RGB and CMY, respectively, to the device independent opponent color space of $Y_yC_xC_z$ using transformations calibrated for a given color imaging system. To transform $\mathbf{g}[\mathbf{m}]$ into $Y_yC_xC_z$, we use an eight-entry CMY$\mapsto$ $Y_yC_xC_z$ LUT. This LUT contains the $Y_yC_xC_z$ values of the eight possible colorant combinations known as the Neugebauer primaries (C, M, Y, R, G, B, K, W) measured with a spectrophotometer, for a given printer and paper substrate. We will discuss how we obtain the transformation for RGB$\mapsto$ $Y_yC_xC_z$ in Section V. We denote the $Y_yC_xC_z$ continuous-tone original color image and the $Y_yC_xC_z$ rendered color image by $\mathbf{f}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$, and their components by $f_i[\mathbf{m}]$, $i = Y_y, C_x, C_z$ and $g_i[\mathbf{m}]$, $i = Y_y, C_x, C_z$, respectively. Note that we also represent the RGB components of a continuous-tone color image by $f_i[\mathbf{m}]$. The quantity that $f_i[\mathbf{m}]$ stands for will be clear from the context. We define the error image in the $Y_yC_xC_z$ color space and its components as

$$\begin{aligned}\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] &\equiv \mathbf{f}_{Y_yC_xC_z}[\mathbf{m}] - \mathbf{g}_{Y_yC_xC_z}[\mathbf{m}] \\ e_i[\mathbf{m}] &\equiv f_i[\mathbf{m}] - g_i[\mathbf{m}], \quad i = Y_y, C_x, C_z.\end{aligned} \tag{8}$$

Using the linear, channel independent and shift-invariant HVS model introduced in Section II and assuming additive interaction between neighboring dots, we model the perceived error $\tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x})$ in the $Y_yC_xC_z$ opponent color space, which is defined on the continuous spatial coordinates $(\mathbf{x})$ as

$$\begin{aligned}\tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x}) = \sum_{\mathbf{m}} &\mathrm{diag}(\tilde{p}_{\mathrm{dot\_Y_y}}(\mathbf{x} - \mathbf{Xm}) \\ &\tilde{p}_{\mathrm{dot\_C_x}}(\mathbf{x} - \mathbf{Xm})\tilde{p}_{\mathrm{dot\_C_y}}, (\mathbf{x} - \mathbf{Xm}))\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]\end{aligned} \tag{9}$$

where $\tilde{p}_{\mathrm{dot\_i}}(\mathbf{x}) \equiv \tilde{p}_i(\mathbf{x}) * p_{\mathrm{dot}}(\mathbf{x})$ is the HVS point spread function for the ith component of the $Y_yC_xC_z$ opponent color space $\tilde{p}_i(\mathbf{x})$ convolved with the printer dot profile $p_{\mathrm{dot}}(\mathbf{x})$, $\mathbf{X}$ is a periodicity matrix whose columns form the basis for the lattice of printer addressable dots and $\mathrm{diag}(\cdot)$ is a diagonal matrix with the diagonal elements listed between the parentheses. Since the

printer dot profile has much more limited support than the HVS point spread function and we assume that the printer dot profile has unit volume, $\tilde{p}_{\mathrm{dot\_i}}(\mathbf{x}) \approx \tilde{p}_i(\mathbf{x})$. Therefore, we can rewrite (9) as

$$\tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x}) = \sum_{\mathbf{m}} \tilde{\mathbf{P}}(\mathbf{x}-\mathbf{X}\mathbf{m})\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] \tag{10}$$

where

$$\tilde{\mathbf{P}}(\mathbf{x}) \equiv \mathrm{diag}\left(\tilde{p}_{Y_y}(\mathbf{x}), \tilde{p}_{C_x}(\mathbf{x}), \tilde{p}_{C_y}(\mathbf{x})\right). \tag{11}$$

We define the error metric E to be the sum of the total squared perceived errors in all three components of the $Y_yC_xC_z$ color space.

$$E = \int \tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x})^T \tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x})\mathrm{d}\mathbf{x}. \tag{12}$$

Substituting (10) into (12), we get

$$E = \sum_{\mathbf{m}}\sum_{\mathbf{n}} \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]^T \left(\int \tilde{\mathbf{P}}(\mathbf{x}-\mathbf{X}\mathbf{m})\tilde{\mathbf{P}}(\mathbf{x}-\mathbf{X}\mathbf{n})\mathrm{d}\mathbf{x}\right) \times \mathbf{e}_{Y_yC_xC_z}[\mathbf{n}]. \tag{13}$$

We denote the matrix of autocorrelation functions of $\tilde{p}_i(\mathbf{x})$, for $i = Y_y, C_x, C_z$, by $\mathbf{c}_{\tilde{p}\tilde{p}}(\mathbf{x})$, and the matrix of cross correlation functions between $\tilde{p}_i(\mathbf{x})$ and $\tilde{e}_i(\mathbf{x})$, for $i = Y_y, C_x, C_z$ by $\mathbf{c}_{\tilde{p}\tilde{e}}(\mathbf{x})$

$$\mathbf{c}_{\tilde{p}\tilde{p}}(\mathbf{x}) \equiv \mathrm{diag}\left(\int \tilde{p}_{Y_y}(\mathbf{y})\tilde{p}_{Y_y}(\mathbf{y}+\mathbf{x})\mathrm{d}\mathbf{y} \int \tilde{p}_{C_x}(\mathbf{y})\tilde{p}_{C_x}(\mathbf{y}+\mathbf{x})\mathrm{d}\mathbf{y} \int \tilde{p}_{C_z}(\mathbf{y})\tilde{p}_{C_z}(\mathbf{y}+\mathbf{x})\mathrm{d}\mathbf{y}\right) \tag{14}$$

$$\mathbf{c}_{\tilde{p}\tilde{e}}(\mathbf{x}) \equiv \mathrm{diag}\left(\int \tilde{p}_{Y_y}(\mathbf{y})\tilde{e}_{Y_y}(\mathbf{y}+\mathbf{x})\mathrm{d}\mathbf{y} \int \tilde{p}_{C_x}(\mathbf{y})\tilde{e}_{C_x}(\mathbf{y}+\mathbf{x})\mathrm{d}\mathbf{y} \int \tilde{p}_{C_z}(\mathbf{y})\tilde{e}_{C_z}(\mathbf{y}+\mathbf{x})\mathrm{d}\mathbf{y}\right). \tag{15}$$

Evaluating these two matrices at the pixels in the printer grid $\mathbf{x} = \mathbf{X}\mathbf{m}$, gives us $\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$. Then, we can rewrite (13) as

$$\begin{aligned} E &= \sum_{\mathbf{m}}\sum_{\mathbf{n}} \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]^T \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{n}]\mathbf{e}_{Y_yC_xC_z}[\mathbf{n}] \\ &= \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]. \end{aligned} \tag{16}$$

As stated above, the goal of our proposed method is to minimize the perceived error E in the device-independent opponent color space $Y_yC_xC_z$ using the DBS algorithm. Lee and Allebach [42] also used the DBS algorithm for model-based color halftoning, but they operated directly in the CMY colorant space instead of $Y_yC_xC_z$. They employed only a luminance HVS model to enforce exclusion of visually nonhomogeneous patterns for the total dot distribution, as well as for each individual colorant texture. Their algorithm enables explicit control over the individual colorant textures and the way in which these textures interact to determine the overall visual appearance of the halftone and to minimize dot-on-dot printing.

In the next section, we explain an efficient technique for calculating the effect of trial halftone changes (toggles and swaps) on the error metric E (16), a generalization of a technique developed by Analoui and Allebach [41] and further refined by Lieberman and Allebach [24] for monochrome halftoning using DBS.

C. Efficient Computation of the Effect of Trial Halftone Changes

For a bilevel three-colorant CMY printer, each halftone pixel displays one of 8 (2^3) possible colorant combinations. These combinations correspond to one, two, and three color overprints of the colorants C, M, Y or to no colorant on paper (W). An alternate labeling of these combinations is, therefore, W, C, M, Y, MY, CY, CM, and CMY, respectively. Similarly, for a binary four-colorant printer with full undercolor removal, there are also eight possible combinations which are labeled W, C, M, Y, MY, CY, CM, and K, respectively. Therefore, a toggle for CDBS is defined as changing the value of a halftone pixel from a given colorant combination to a different one out of these eight.

Consider a trial halftone pixel toggle at index $\mathbf{m}_t$. We denote the change that this toggle will cause in the $Y_yC_xC_z$ color rendered image $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$ at index $\mathbf{m}_t$ by $\mathbf{a}[\mathbf{m}_t] = [a_{Y_y}[\mathbf{m}_t], a_{C_x}[\mathbf{m}_t], a_{C_z}[\mathbf{m}_t]]^T$. We then define $\mathbf{A}[\mathbf{m}_t] = \mathrm{diag}(\mathbf{a}[\mathbf{m}_t])$. With this toggle, $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$, $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$, and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ will change as follows:

$$\mathbf{g'}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{g}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m}-\mathbf{m}_t] \tag{17}$$

$$\mathbf{e'}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m}-\mathbf{m}_t] \tag{18}$$

$$\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{m}_t] \tag{19}$$

where the prime denotes the version after the toggle. The change $\triangle E_t$ in E due to the toggle can then be written as

$$\begin{aligned} \triangle E_t &= \sum_{\mathbf{m}} {\mathbf{e'}_{Y_yC_xC_z}}^T[\mathbf{m}]\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] - \sum_{\mathbf{m}} {\mathbf{e}_{Y_yC_xC_z}}^T[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\ &= \sum_{\mathbf{m}} \left(\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m}-\mathbf{m}_t]\right)^T \times \left(\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{m}_t]\right) - \sum_{\mathbf{m}} {\mathbf{e}_{Y_yC_xC_z}}^T[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\ &= \mathbf{a}[\mathbf{m}_t]^T\mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2\mathbf{a}[\mathbf{m}_t]^T\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_t] \end{aligned} \tag{20}$$

where we have used $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] = \sum_{\mathbf{n}} \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{n}]\mathbf{e}_{Y_yC_xC_z}[\mathbf{n}]$ and $\mathbf{c}_{\tilde{p}\tilde{p}}[-\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$. The computational cost of evaluating $\triangle E_t$ can be reduced significantly by storing $\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ in look-up-tables (LUTs). If the toggle under trial is accepted, only then we update the $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ LUTs using (17) and (19), respectively.

Another type of trial halftone change we will analyze in this section is swapping the colorant combination of a given pixel in the halftone with that of another pixel. A swap is a more subtle change to the halftone under test than a toggle. By definition, it conserves the average local "color" in the area containing the swapped pixels, and can be interpreted as two consecutive toggles. Consider a trial swap of the halftone pixels at indices $\mathbf{m}_t$

and $\mathbf{m}_s$. Let us denote the changes that the swap under trial will cause in the $Y_yC_xC_z$ rendered color image at indices $\mathbf{m}_t$ and $\mathbf{m}_s$ by $\mathbf{a}[\mathbf{m}_t]$ and $\mathbf{a}[\mathbf{m}_s]$, respectively. Similar to $\mathbf{A}[\mathbf{m}_t]$, we define $\mathbf{A}[\mathbf{m}_s] \equiv \mathrm{diag}(\mathbf{a}[\mathbf{m}_s])$ Then, the change in the error metric $\triangle E_s$ due to the swap can be written as

$$\triangle E_s = \triangle E_t + \mathbf{a}[\mathbf{m}_s]^T \mathbf{A}[\mathbf{m}_s] \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2\mathbf{a}[\mathbf{m}_s]^T \mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}_s] \quad (21)$$

where $\triangle E_t$ is the change in the error metric due to the toggle of the halftone pixel at $\mathbf{m}_t$ to the colorant combination at $\mathbf{m}_s$ and $\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}_s]$ is the cross correlation vector after this toggle, evaluated at $\mathbf{m} = \mathbf{m}_s$. Substituting (19) and (20) into (21), we obtain

$$\begin{aligned} \triangle E_s = {} & \left(\mathbf{a}[\mathbf{m}_t]^T \mathbf{A}[\mathbf{m}_t] + \mathbf{a}[\mathbf{m}_s]^T \mathbf{A}[\mathbf{m}_s]\right) \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{0}] \\ & + 2\mathbf{a}[\mathbf{m}_t]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_t] + 2\mathbf{a}[\mathbf{m}_s]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_s] \\ & + 2\mathbf{a}[\mathbf{m}_s]^T \mathbf{A}[\mathbf{m}_t] \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}_s - \mathbf{m}_t]. \end{aligned} \quad (22)$$

Similar to $\triangle E_t$, the computational cost of finding $\triangle E_s$ can also be reduced greatly by keeping LUTs for $\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$. If the swap under trial is accepted, the $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ LUTs are updated as follows:

$$\begin{aligned} \mathbf{g}'_{Y_yC_xC_z}[\mathbf{m}] = {} & \mathbf{g}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m} - \mathbf{m}_t] \\ & + \mathbf{a}[\mathbf{m}_s]\delta[\mathbf{m} - \mathbf{m}_s] \end{aligned} \quad (23)$$

$$\begin{aligned} \mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] = {} & \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_t] \\ & + \mathbf{A}[\mathbf{m}_s]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_s]. \end{aligned} \quad (24)$$

D. Efficient Search Strategy

Lieberman and Allebach [43] showed that the time complexity of monochrome DBS can be improved considerably by using two efficient search techniques, instead of computing the effect of every possible toggle and every possible NN swap in raster-scan order. We generalize and apply these two strategies to CDBS. First, we limit the search of possible swapping pixels to those in the anticausal neighborhood of a given pixel only, and, hence, decrease the number of trial swap computations by 50%. Second, we exploit the fact that the cost of a trial change computation is much lower than that of the actual update if the trial change is accepted. Pixels of the halftone image are partitioned into $C \times C$ cells. These cells are processed in raster-scan order. Within each cell, the effect of all possible toggles and swaps are computed; but only one halftone pixel change is allowed for the whole cell. Only the one toggle or swap within the cell that causes the largest decrease in the error metric is accepted. We found empirically that $C = 3$ gives the lowest time complexity for CDBS.

IV. Dot Interaction Models

It is common practice to conceptualize a printer as producing dots that are perfect $(1/R) \times (1/R)\text{in}^2$ squares where R is the printer resolution in dots per inch (dpi), as shown in Fig. 1(a). For our model, developed in Section III-B, we assumed that printer dots have a unit volume profile with much smaller support than the HVS point spread function, that the dot profiles are identical for all colorants, and that dot interactions are additive. In reality, most printers fail to satisfy even these less restrictive assumptions. They produce dots that may overlap as depicted in Fig. 1(b). There may be dot placement errors and the colorant interactions between neighboring halftone pixels are not additive. Also, the printer dot profiles for different colorants may be considerably different, as can be observed in a high-resolution scan of a color halftone.

Color halftoning algorithms that do not account for these dot interactions suffer from incorrect color reproduction. Neglecting these interactions also leads to poor color texture rendition. Researchers have taken several paths to counteract dot interactions in halftoning. The most common approach borrowed from monochrome halftoning is to use clustered dot textures instead of dispersed dot textures. This approach increases the visibility of halftone patterns and limits the spatial resolution of the halftone, resulting in loss of detail rendition [44]. It also distorts the colors in the image [22].

The second approach frequently used in monochrome halftoning [45], [46] is called *tone correction*. This approach precompensates for dot interactions by modifying the image data prior to halftoning and printing, using a LUT of input absorptance versus printed absorptance. In color halftoning, the method can theoretically be applied by halftoning and printing all possible colors, making colorimetric measurements of the resulting halftones and applying the inverse mapping of this transformation to color images prior to halftoning, to obtain halftones with desired colorimetric values. Since measuring halftone patches for a large set of colors, e.g., $256^3 \approx 16{,}750{,}000$ for an 8-bit color imaging system is infeasible, it is impossible to have a complete *color correction* LUT. Instead, a limited number of halftoned colors are stored in a LUT, and interpolation schemes are employed to obtain the inverse mapping for the rest of the colors.

The third approach to compensating dot interaction in halftoning is the model-based approach in which a printer model accounting for nonideal dot profiles and interactions between neighboring dots is incorporated into the halftoning algorithm. One of the most widely used dot models for monochrome printers is the hard circular dot model introduced by Roetling and Holladay [46]. This popular model is an idealization of printer behavior and assumes that every printer dot is a perfect circle. The circular dot modeling approach is easily generalizable to color halftones. In [22], Pappas generalized the circular dot model to three colorant printers where the color of a given pixel in the halftone is expressed as a sum of the overlapped colors of the segments, weighted by the areas of the segments. He used this model for color error diffusion and an iterative least squares model-based color halftoning technique. This parametric circular color dot overlap model was also used by Kim *et al.* [16]. They incorporated the model into color error diffusion in the device independent CIE $L^*a^*b^*$ space, and employed a Neugebauer color-mixing model to find the average color of a target region in a printed test pattern that is made up of segments displaying different overlapped colors. Lee *et al.* [47] used a similar area coverage LUT-based color dot interaction model to modify the dot diffusion halftoning algorithm introduced by Knuth [48]. Lai and Chen [49] exploited the circular dot model presented in [22] in vector color error diffusion in YC_rC_b space and also incorporated edge enhancement.

Another novel dot interaction model first introduced for monochrome printers and later extended to color printers is the 2×2 centering model which was developed by Wang *et al.* [25] and generalized to color by Wang [26]. In this model, the printer grid is assumed to be offset from the pixel grid for the continuous-tone image to be halftoned. This results in a structure where four (2×2) neighboring printer dots in the printer grid contribute to the rendered tone (color) of one image pixel. The parameters of the model for the monochrome (color) case are the reflectances of all possible 2×2 monochrome (color) overlap patterns. The Yule–Nielsen effect can also be incorporated into this model [50]. We will discuss this model in detail in Section IV-A.

A. Color 2×2 Centering Method

For a given halftone, there are two grids. The first one, the printer grid, is the physical grid on which the printer places the colorant dots. The second one, the image grid, showing the rendered colors is a conceptual one. Many dot interaction models such as the circular dot model choose to align these two grids and compensate for the dot interactions between neighboring printer dots (pixels in the printer grid) which also correspond to neighboring rendered image pixels as shown in Fig. 2(a). In these models, full accounting for the dot interactions necessitates measuring all possible $(8^9 \approx 134{,}200{,}000)$ colorant combinations in a 3×3 pixel neighborhood [51]. Since the number of measurements to be made is prohibitively large, researchers have tried to quantify dot overlap using other approaches.

The 2×2 centering model [25], [26] provides a means of quantifying dot overlap in color halftones with only 1072 measurements. The novelty of this method is the idea that the printer grid is assumed to be shifted by $(0.5, 0.5)$ pixels from the image grid as shown in Fig. 2(b). When the image grid is imposed on top of the printer grid as depicted in Fig. 2(b), every pixel in the image grid has the same rendered "color" that is created by the four (2×2) printer pixels and the overlaps in between these four printer pixels. Since there are eight possible colorant combinations per printer pixel for the three colorant bilevel printers or four colorant bilevel printers with full undercolor removal that we are modeling, this approach results in $8^4(4096)$ possible 2×2 colorant combinations, i.e., "colors" for each rendered image pixel instead of the eight that the conventional method provides [Fig. 2(a)]. The assumption that the dots are symmetric with respect to their vertical and horizontal axes decreases the number of unique 2×2 colorant combinations to 1072. Calibration of the model necessitates the colorimetric measurement of these 1072 overlapping patterns. Symmetry characteristics of the pattern allows us to tile the 2×2 colorant combinations vertically and horizontally to form a color patch with average XYZ tristimulus value equal to the XYZ tristimulus value of this combination as shown in Fig. 3. We then macroscopically measure the XYZ tristimulus value directly using a spectrophotometer.

Macroscopic measurement of the colorant combinations allows us to quantify the dot overlap in these patterns and to measure the overlapped rendered color of the four colorant combinations making up the pattern. These 1072 measurements depend on both the colorant set and the paper substrate. We convert these measured XYZ values to $Y_yC_xC_z$ and store them in a LUT to incorporate into our CDBS halftoning algorithm, as was done by Pappas *et al.* [51] with a LUT of reflectances of all unique 3×3 pixel neighborhoods for monochrome error diffusion. In Section IV-B, we will explain how we modify CDBS to incorporate the LUT summarizing the 2×2 centering-based dot interaction model.

B. The 2×2 Centering-Based CDBS

Incorporating the 2×2 centering method into CDBS requires changes in the definition of the $Y_yC_xC_z$ rendered color image and the computation of the effect of trial halftone changes on the error metric. In Section III-B, we stated that the $Y_yC_xC_z$ rendered color image $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$ is obtained from the CMY color halftone $\mathbf{g}[\mathbf{m}]$ by a pixelwise CMY$\mapsto Y_yC_xC_z$ transformation calibrated for a given color imaging system. In 2×2 centering-based CDBS, the $Y_yC_xC_z$ centering-based rendered color image at pixel location $\mathbf{m}$ which we will denote by $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$ is no longer a function of only $\mathbf{g}[\mathbf{m}]$; it is a function of the colorant combinations at the four printer pixels contributing to the rendered color at this pixel [Fig. 2(b)]. We denote the neighborhood of these four printer pixels that intersect with pixel $\mathbf{m}$ in the image pixel by $\mathcal{N}(\mathbf{m})$. Therefore, in centering-based CDBS

$$\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}] \equiv \mathcal{O}_{Y_yC_xC_z}(g[\mathbf{m}], \mathbf{m} \in \mathcal{N}(\mathbf{m})) \tag{25}$$

where $\mathcal{O}_{Y_yC_xC_z}$ is the function that generates the rendered $Y_yC_xC_z$ for a given input 2×2 overlapping colorant combination using the LUT of the measured $Y_yC_xC_z$ values for the 4096 such combinations. We define a dual neighborhood $\mathcal{N}^{-1}(\mathbf{m})$ in the image grid which denotes the neighborhood made of four image pixels that intersect with pixel $\mathbf{m}$ in the printer grid, i.e., the four image pixels that are affected by the choice of the colorant combination in the halftone at pixel $\mathbf{m}$ in the printer grid.

For the 2×2 centering-based CDBS, the error image in the $Y_yC_xC_z$ color space $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$ is given by

$$\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] \equiv \mathbf{f}_{Y_yC_xC_z}[\mathbf{m}] - \mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]. \tag{26}$$

The remainder of the equations defining the error metric for CDBS (9)–(16) also apply to the 2×2 centering-based CDBS. The modifications to the computation of the effect on the error metric of the trial toggles and swaps are described below.

In centering-based CDBS, a trial toggle at halftone pixel $\mathbf{m}_t$ from a given colorant combination to a different one out of the eight possible colorant combinations causes a change in the $Y_yC_xC_z$ rendered color image at the four image pixel locations in $\mathcal{N}^{-1}(\mathbf{m}_t)$, as shown in Fig. 4(a). This trial toggle will change the $Y_yC_xC_z$ rendered color image by $\mathbf{a}[\mathbf{n}]$ at the pixels $\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)$. Similar to $\mathbf{A}[\mathbf{m}_t]$, we define $\mathbf{A}[\mathbf{n}] \equiv \mathrm{diag}(\mathbf{a}[\mathbf{n}])$. This trial toggle will affect $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$, $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$, and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ as follows:

$$\mathbf{g}^{o\prime}{}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]\delta[\mathbf{m} - \mathbf{n}] \tag{27}$$

$$\mathbf{e'}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \sum_{\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]\delta[\mathbf{m}-\mathbf{n}] \quad (28)$$

$$\mathbf{c'}_{\tilde{p}\tilde{e}}[\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \sum_{\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{A}[\mathbf{n}]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{n}] \quad (29)$$

where the prime denotes the version after the toggle. We observe that, for a given trial toggle, centering-based CDBS changes a larger area in the $Y_yC_xC_z$ rendered color image than regular CDBS does, affecting 4 image pixels instead of 1. However, the changes $\triangle Y_yC_xC_z$ observed at these four locations $\{\mathbf{a}[\mathbf{n}]|\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)\}$ are generally smaller in magnitude than we see at the single pixel location for regular CDBS. The change in E due to this trial toggle can be written as

$$\begin{aligned}\triangle E_t &= \sum_{\mathbf{m}} \mathbf{e'}_{Y_yC_xC_z}{}^T[\mathbf{m}]\mathbf{c'}_{\tilde{p}\tilde{e}}[\mathbf{m}] - \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}{}^T[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\ &= \sum_{\mathbf{m}} \left(\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \sum_{\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]\delta[\mathbf{m}-\mathbf{n}]\right)^T \\ &\quad \times \left(\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] + \sum_{\mathbf{k}\in\mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{A}[\mathbf{k}]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{k}]\right) \\ &\quad - \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}{}^T[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\ &= \sum_{\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)} \sum_{\mathbf{k}\in\mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]^T\mathbf{A}[\mathbf{k}]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{n}-\mathbf{k}] \\ &\quad + \sum_{\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)} 2\mathbf{a}[\mathbf{n}]^T\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{n}]. \end{aligned} \quad (30)$$

If the toggle under trial is accepted, we modify the $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ LUTs using (27) and (29), respectively.

Now, consider a trial swap of the halftone pixels $\mathbf{m}_t$ and $\mathbf{m}_s$. This trial change to the halftone will modify the $Y_yC_xC_z$ rendered color image at the pixels in $\mathcal{N}^{-1}(\mathbf{m}_t)\bigcup\mathcal{N}^{-1}(\mathbf{m}_s)$. Assuming that $\mathbf{m}_t \neq \mathbf{m}_s, \mathcal{N}^{-1}(\mathbf{m}_t)\bigcup\mathcal{N}^{-1}(\mathbf{m}_s)$ may contain either six or seven pixels, depending on the relative locations of $\mathbf{m}_t$ and $\mathbf{m}_s$. Fig. 4(b) depicts the seven image pixels that will exhibit a change in $Y_yC_xC_z$ values when the colorant combinations at the two diagonally touching pixels $\mathbf{m}_t$ and $\mathbf{m}_s$ are swapped. Similar to the trial toggle analysis, we denote the change that this swap will create in the $Y_yC_xC_z$ rendered color image at the pixels $\mathbf{n}\in\mathcal{N}^{-1}(\mathbf{m}_t)\bigcup\mathcal{N}^{-1}(\mathbf{m}_s)$ by $\mathbf{a}[\mathbf{n}]$. We again define $\mathbf{A}[\mathbf{n}] \equiv \mathrm{diag}(\mathbf{a}[\mathbf{n}])$. The effect of this trial swap on $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$, $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$, and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ is given by the generalization of (27)–(29) with $\mathcal{N}^{-1}(\mathbf{m}_t)$ replaced by $\mathcal{N}^{-1}(\mathbf{m}_t)\bigcup\mathcal{N}^{-1}(\mathbf{m}_s)$. Similarly, $\triangle E_s$, the change in E due to this trial swap, is given by (30) with the same modification. If the swap under trial is accepted, we alter $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ accordingly. We note that as in the case of regular CDBS, in centering-based CDBS a swap is a more subtle change (of larger support and smaller effect per pixel) to the $Y_yC_xC_z$ rendered color image than a toggle, and it tends to conserve the average local "color" in the area containing the swapped pixels. We also note that due to the fact that a toggle in centering-based CDBS brings a much more subtle change to the halftone than one in regular CDBS, we obtain very high-quality halftones doing only toggles in centering-based CDBS. Unlike regular CDBS and monochrome DBS, swaps do not lead to a significant improvement in texture. As with regular CDBS, for efficient implementation of centering-based CDBS, we again exploit the modified search strategies discussed in Section III-D.

V. Halftoning Calibration

One of the major problems in achieving device-independent color imaging is color printer calibration. In simplified terms, color printer calibration is the process of computing what input to feed into a printer to obtain a desired "color." It is the evaluation of the inverse printer transfer function which maps points in a device independent printer output space, e.g., $Y_yC_xC_z$ to points in the device dependent printer input colorant space CMY or CMYK. In the literature on color printer calibration, the printer is viewed as a black box with a predetermined halftoning algorithm which creates a halftone with a desired input colorant combination. This definition is valid for the majority of color printers where the user has no control over the halftoning employed in the printer; and the default halftoning technique aims to preserve the average colorant level in the output halftone.

Unlike colorant space-based, colorant-level preserving halftoning algorithms, CDBS operates with an input in the device independent space of $Y_yC_xC_z$. CDBS minimizes a least squares error metric in the $Y_yC_xC_z$ space, and this error minimization favors color textures pleasing to the human viewer due to the HVS model incorporated. Our experimentation has shown that CDBS is biased to create visually pleasing halftones at the expense of leading to $Y_yC_xC_z$ values somewhat different than the desired input $Y_yC_xC_z$ values, especially in the very dark colors. Subject to this bias and limit in accuracy, CDBS is a colorimetric technique developed to minimize a measure of perceived error in $Y_yC_xC_z$. Therefore, it is not calibrated on a colorant basis. This is mainly due to the fact that different spatial arrangements of a single desired colorant combination in a region in the image can lead to different $Y_yC_xC_z$ values. To allow the CDBS algorithm to be used in applications where the calibration of the color printer on a colorant basis is important, such as business graphics, we need to devise a calibration scheme that can control the colorant combination in a CDBS halftone. As we will explain below, we can achieve this by calibrating the transformation RGB$\mapsto Y_yC_xC_z$ applied to the original continuous-tone color image $f[\mathbf{m}]$.

Since our task is to find the transformation from a colorant space to a device independent printer output color space, CDBS halftoning calibration can be viewed as the inverse of the problem of color printer calibration. Therefore, the literature on color printer calibration is relevant to our task. As stated above, a complete color correction or a complete printer calibration is impossible due to the large number of measurements to be made. Thus, the researchers who have investigated this problem

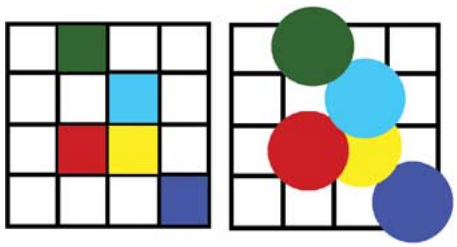

Fig. 1. (a) Ideal printer dots. (b) Actual printer dots.

Fig. 2. Alignment of the printer and image grids: (a) conventional method and (b) 2×2 centering method.

relied on interpolation techniques along with a limited set of measurements [52]–[55]. Below we present two methods to calibrate the CDBS halftoning algorithm. The first one, which we will discuss in Section V-A, is a simple undercolor removal-based mapping. The second method, which we will present in Section V-B, is motivated by the printer calibration methods listed above. It uses a cellular measurement-based grid structure which is iteratively updated, and then exploits tetrahedral interpolation. The second method is developed solely for centering-based CDBS.

A. Simple RGB$\mapsto Y_yC_xC_z$ Calibration

This method is a common undercolor removal (gray component replacement) strategy. It is based on the assumption that the printing geometry is dot-on-dot, i.e., the colorants are placed exactly on top of each other. In this method, the colorants C, M, and Y are referred to as subtractive primaries. The colors corresponding to two color overprints $R(M+Y)$, $G(C+Y)$, and $B(C+M)$ are referred to as subtractive secondaries. The three-color overprint is called K and unprinted substrate is defined as W. The method computes the proportions of the eight Neugebauer primaries, fi, for $i = W, C, M, Y, R, G, B, K$ for a given desired color in an image defined by a 3-tuple $(R_i, G_i, B_i) \in [0, 1]$ according to the following equations:

$$
\begin{aligned}
&\texttt{fW} = \min(\texttt{R\_i}, \texttt{G\_i}, \texttt{B\_i});\ \ \texttt{fK} = 1 - \max(\texttt{R\_i}, \texttt{G\_i}, \texttt{B\_i})\\
&\text{if } \texttt{R\_i} = \min(\texttt{R\_i}, \texttt{G\_i}, \texttt{B\_i})\\
&\ \ \texttt{fR} = 0;\ \ \texttt{fM} = 0;\ \ \texttt{fY} = 0;\ \ \texttt{fC} = \min(\texttt{R\_i}, \texttt{G\_i}) - \texttt{fW};\\
&\ \ \texttt{fG} = \texttt{G\_i} - \texttt{fC} - \texttt{fW};\ \ \texttt{fB} = \texttt{B\_i} - \texttt{fC} - \texttt{fW}\\
&\text{end}\\
&\text{if } \texttt{G\_i} = \min(\texttt{R\_i}, \texttt{G\_i}, \texttt{B\_i})\\
&\ \ \texttt{fG} = 0;\ \ \texttt{fC} = 0;\ \ \texttt{fY} = 0;\ \ \texttt{fM} = \min(\texttt{R\_i}, \texttt{B\_i}) - \texttt{fW};\\
&\ \ \texttt{fR} = \texttt{R\_i} - \texttt{fM} - \texttt{fW};\ \ \texttt{fB} = \texttt{B\_i} - \texttt{fM} - \texttt{fW}\\
&\text{end}\\
&\text{if } \texttt{B\_i} = \min(\texttt{R\_i}, \texttt{G\_i}, \texttt{B\_i})\\
&\ \ \texttt{fB} = 0;\ \ \texttt{fC} = 0;\ \ \texttt{fM} = 0;\ \ \texttt{fY} = \min(\texttt{R\_i}, \texttt{G\_i}) - \texttt{fW};\\
&\ \ \texttt{fR} = \texttt{R\_i} - \texttt{fY} - \texttt{fW};\ \ \texttt{fG} = \texttt{G\_i} - \texttt{fY} - \texttt{fW}\\
&\text{end}.
\end{aligned}
$$

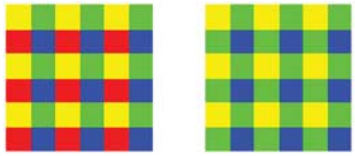

Fig. 3. Two examples of 2×2 colorant combination bit maps.

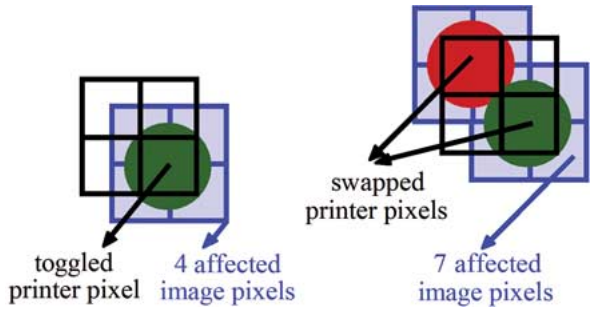

Fig. 4. Effect of a (a) trial toggle and (b) trial swap.

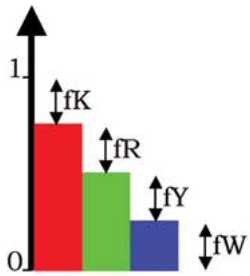

Fig. 5. Undercolor removal strategy.

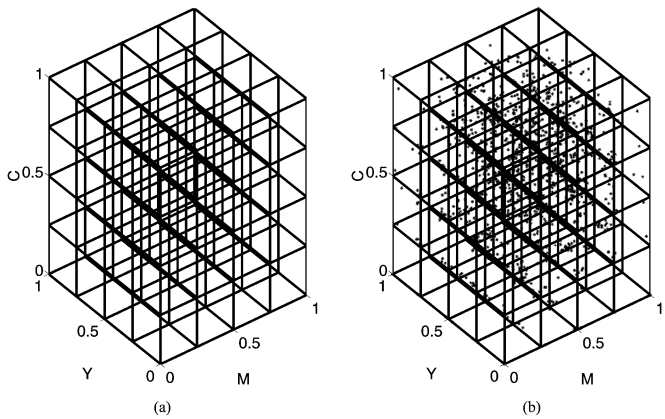

Fig. 6. (a) Theoretical $5 \times 5 \times 5$ $\widehat{\mathrm{CMY}}$ grid. (b) Actual $\widehat{\mathrm{CMY}}$ measurements superimposed on the $5 \times 5 \times 5$ grid.

Since R_i, G_i, and B_i are color coordinates in an additive color system, such as a display, fW is set equal to the minimum of R_i, G_i, and B_i, and fK is set equal to one minus their maximum. For example, if $B_i = \min(R_i, G_i, B_i)$ as shown in Fig. 5, then fW is set equal to B_i and the proportion of the corresponding subtractive secondary fB is set equal to zero. Since we assume dot-on-dot printing geometry, this also implies that fC and fM are equal to zero. In this method, the desired color can always be created using only one subtractive secondary (R for Fig. 5), one subtractive primary (Y for Fig. 5), black, and white. This method necessitates the measurement

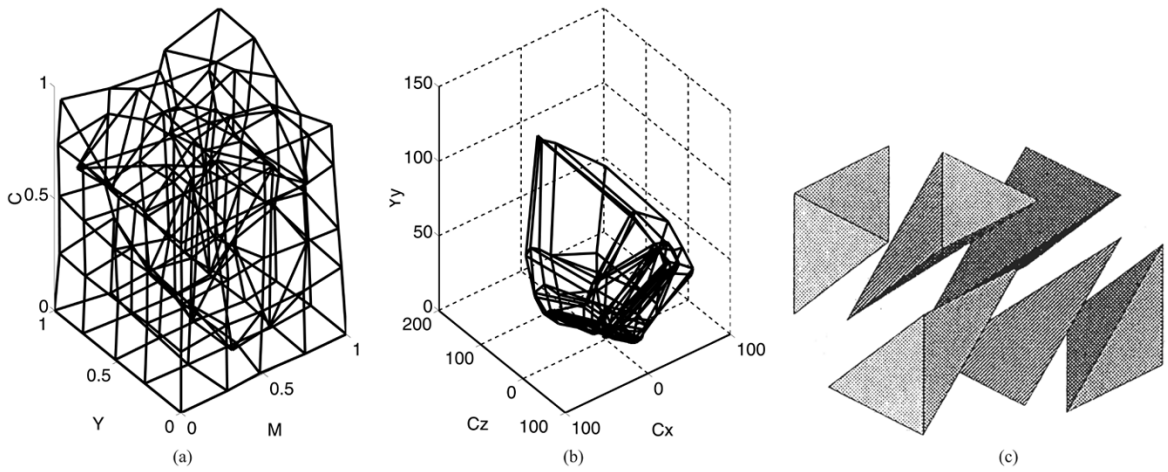

Fig. 7. (a) Actual $\widehat{\mathrm{CMY}}$ grid, (b) $Y_yC_xC_z$ grid, and (c) Tessellation of a subcube in $\widehat{\mathrm{CMY}}$ into tetrahedra.

of the $Y_yC_xC_z$ values of only the eight Neugebauer primaries. After fi, for $i = W, C, M, Y, R, G, B, K$ have been computed, the $Y_yC_xC_z$ coordinates of the desired color can be found as a weighted sum of the $Y_yC_xC_z$ values of the eight Neugebauer primaries, using the fi as weights.

Since this method assumes that the color printer has three colorants and that the color black is obtained by putting down C, M, and Y together, referred to as composite black, caution should be taken in generalizing this strategy to four colorant CMYK printers that use a black colorant to create the color black, referred to as process black. Generally, process black has a $Y_yC_xC_z$ value significantly closer to $(0, 0, 0)$ than composite black. For CMYK printers, we have observed, as have others [8], [13], that favoring composite black dots over process black dots improves the color texture of the halftones. Therefore, for CMYK printers with full undercolor removal, we modify this undercolor removal strategy, as follows. We decrease the proportion of the black color fK and linearly increase the proportions of the remaining Neugebauer primaries using (31) so that the sum of fi' equals 1, since the physical area coverage of the Neugebauer primaries should sum to 1. We find the $Y_yC_xC_z$ coordinates using the modified weights

$$fK' = fK^2, \quad fi' = \frac{1 - fK'}{1 - fK} fi, \;\; i = W, C, M, Y, R, G, B. \tag{31}$$

As we shall see in Section VI, CDBS with the simple RGB$\mapsto Y_yC_xC_z$ calibration approach gives us limited success in applications such as business graphics where the intent is to preserve the purity of colors, i.e., the desired colorant combinations. Therefore, we propose the following iterative halftoning calibration technique solely for centering-based CDBS.

B. Iterative RGB$\mapsto Y_yC_xC_z$ Calibration

To convert RGB $\mathbf{f}[\mathbf{m}]$ to $\mathbf{f}_{Y_yC_xC_z}[\mathbf{m}]$, we first change the RGB values to CMY using $C = 1 - R, M = 1 - G, Y = 1 - B$. We denote this continuous pseudocolorant space by $\widehat{\mathrm{CMY}}$. In this method, our goal is to increase the accuracy of the simple RGB$\mapsto Y_yC_xC_z$ calibration method using a cellular measurement-based grid structure and iterative update of the grid structure. This calibration method does not require extra colorimetric measurements. It uses the 1072 measurements made before. We first discuss how we construct the initial cellular grid structure. Then, we explain how we update this grid structure.

1) Initial $\widehat{\mathrm{CMY}}$ Grid Structure: We start by feeding the measured $Y_yC_xC_z$ values for the 1072 2×2 colorant combinations into CDBS to obtain halftone patches for them. We choose to start with these values, as they form a good estimate of what the printer gamut will be for this halftoning algorithm. Since we macroscopically measure the $Y_yC_xC_z$ values of the tiled version of these 2×2 colorant combinations (Fig. 3), theoretically these $Y_yC_xC_z$ values correspond to $\widehat{\mathrm{CMY}}$ values $\{(n_C/4), (n_M/4), (n_Y/4)\}$ where $n_C, n_M, n_Y \in \{0, 1, 2, 3, 4\}$. For a four-colorant printer with full undercolor removal, we assume that a K dot is equivalent to C, M, and Y dots put down together. These theoretical $\widehat{\mathrm{CMY}}$ values are the grid points of the uniform $5 \times 5 \times 5$ grid in $\widehat{\mathrm{CMY}}$ shown Fig. 6(a). We compute the actual $\widehat{\mathrm{CMY}}$ values for the 1072 CDBS halftone patches we created by counting the number of dots of each colorant present in the bit maps. If CDBS were an algorithm calibrated on a colorant basis, these values would fill the whole unit cube in $\widehat{\mathrm{CMY}}$ and would be exactly equal to the grid points of the $5 \times 5 \times 5$ $\widehat{\mathrm{CMY}}$ grid of Fig. 6(a). We typically obtain $\widehat{\mathrm{CMY}}$ values as shown in Fig. 6(b). Among these 1072 $\widehat{\mathrm{CMY}}$ values, we pick the 125 that are closest in Euclidean distance to the grid points of the $5 \times 5 \times 5$ grid in $\widehat{\mathrm{CMY}}$. This gives us a nonuniform grid structure that does not fill the unit cube in $\widehat{\mathrm{CMY}}$ as shown in Fig. 7(a) and a corresponding nonuniform grid structure in $Y_yC_xC_z$ as shown Fig. 7(b). There is a 1–1 and onto mapping between the grid points of the ideal $5 \times 5 \times 5$ $\widehat{\mathrm{CMY}}$ grid, the actual $\widehat{\mathrm{CMY}}$ grid, and the $Y_yC_xC_z$ grid.

We then divide each subcube of the ideal $5 \times 5 \times 5$ $\widehat{\mathrm{CMY}}$ grid into 6 tetrahedra as shown in Fig. 7(c). We use this specific tetrahedralization among all possible tetrahedralizations of a cube because deviations from the equi-faced tetrahedra are limited and all interfaces between tetrahedra are oppositely congruent,

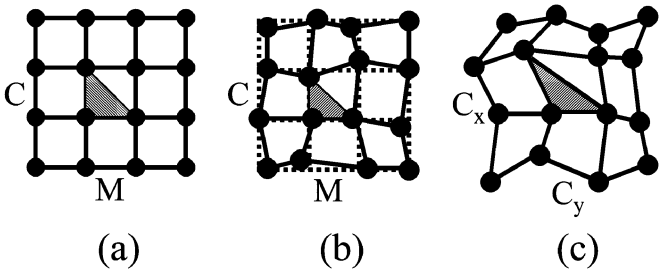

Fig. 8. Illustrating the relation between different grid structures for a 2-D example with a 3×3 grid: (a) ideal $\widehat{\mathrm{CMY}}$ grid, (b) actual $\widehat{\mathrm{CMY}}$ grid, and (c) $Y_y C_x C_z$ grid.

as stated necessary by Gennetten [56] for accurate interpolation. Furthermore, this tetrahedralization with the lower left vertex located at the smallest CMY value satisfies the minimum brightness variance criterion [57] which states that in selecting from possible colorant sets to render a desired color in a halftone, choosing the set with minimum brightness variation reduces the halftone noise and results in visually more pleasing halftones. We carry out the same tessellation in the actual $\widehat{\mathrm{CMY}}$ grid and the $Y_y C_x C_z$ grid. Fig. 8 depicts the 2-D versions of these three grid structures where each subsquare is divided into two triangles. For a desired $\widehat{\mathrm{CMY}}$, we locate the tetrahedron that the color belongs to in the ideal $\widehat{\mathrm{CMY}}$ grid and calculate interpolation weights. Then, we use the $Y_y C_x C_z$ values of the vertices of the corresponding tetrahedron in the $Y_y C_x C_z$ space to interpolate the $Y_y C_x C_z$ value for the desired $\widehat{\mathrm{CMY}}$. We present our results in Section VI.

Even though this calibration method gives better results than the simple Neugebauer model-based $\widehat{\mathrm{CMY}} \mapsto Y_y C_x C_z$ transformation discussed above, it is still incapable of producing CDBS halftones that span the whole unit cube in $\widehat{\mathrm{CMY}}$ since the rendered colors are restricted to the ones in the convex hull of the nonuniform grid structure shown in Fig. 7(a). We take the following iterative grid expansion and correction approach to fill a larger portion of the $\widehat{\mathrm{CMY}}$ unit cube, and to bring the grid points closer to the grid points in the ideal $\widehat{\mathrm{CMY}}$ grid, i.e., to render a larger set of colors and to attain more accurate colorant distributions using CDBS.

2) Iterative Update of the $\widehat{\mathrm{CMY}}$ Grid Structure: For a given grid point $\mathrm{CMY}_{\mathrm{target}}$ in the ideal $5 \times 5 \times 5$ $\widehat{\mathrm{CMY}}$ grid, we find the tetrahedra in the ideal $\widehat{\mathrm{CMY}}$ grid for which this point is a vertex. Depending on the location of $\mathrm{CMY}_{\mathrm{target}}$ in the unit cube, $\mathrm{CMY}_{\mathrm{target}}$ may be a vertex of up to 20 tetrahedra. For each such tetrahedron, we use the corresponding tetrahedron in the actual $\widehat{\mathrm{CMY}}$ grid to find the weights for tetrahedral extrapolation or interpolation of $\mathrm{CMY}_{\mathrm{target}}$. Fig. 9(a) depicts a grid point extrapolated from two tetrahedra. Using these weights and the $Y_y C_x C_z$ values of the vertices of the corresponding tetrahedra in the $Y_y C_x C_z$ grid, we find approximate $Y_y C_x C_z$ values for $\mathrm{CMY}_{\mathrm{target}}$. We feed these values into CDBS, obtain halftone patches, and compute the $\widehat{\mathrm{CMY}}$ for the resulting halftones. We pick the $\widehat{\mathrm{CMY}}$ value with minimum Euclidean distance $\triangle\mathrm{CMY}$ from $\mathrm{CMY}_{\mathrm{target}}$. If this new $\widehat{\mathrm{CMY}}$ value for $\mathrm{CMY}_{\mathrm{target}}$ is closer to $\mathrm{CMY}_{\mathrm{target}}$ then the already existing

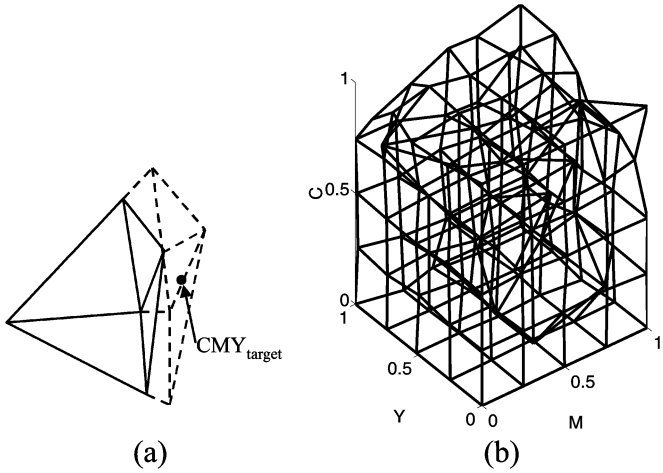

Fig. 9. (a) Extrapolation of $\mathrm{CMY}_{\mathrm{target}}$. (b) The $\widehat{\mathrm{CMY}}$ grid structure after the first update.

grid point in the actual $\widehat{\mathrm{CMY}}$ grid structure, we update our $\widehat{\mathrm{CMY}}$ and $Y_y C_x C_z$ grid structures. We repeat this process in parallel for all of the 125 grid points. This expansion and correction approach gives us a visible improvement in the resulting halftones as we will show in Section VI. In addition, the new $\widehat{\mathrm{CMY}}$ grid structure shown in Fig. 9(b) is noticeably closer to the ideal $5 \times 5 \times 5$ grid. This procedure may be iterated until the desired accuracy in the CMY rendition is obtained with the halftones. In the presentation of this halftoning calibration algorithm, we have assumed that the color printer has three colorants CMY. However, this algorithm can also be applied to four-colorant CMYK printers with full undercolor removal. For a four-colorant CMYK printer with full undercolor removal, each CMYK value can be mapped to a CMY value according to $\mathrm{CMYK} \mapsto (C + K)(M + K)(Y + K)$. Since favoring composite black dots over process black dots improves the color texture of the halftones, we modify the previously described method for choosing the points for our grid structures in $\widehat{\mathrm{CMY}}$ as follows. We add an empirically determined penalty term $(K - \min(CMY_{\mathrm{grid}})^2)^2$ if $K > \min(CMY_{\mathrm{grid}})^2$ to our distance metric $\triangle\mathrm{CMY}$. That is, given two candidate points for a given grid point with the same CMY values, we favor the one with less K coverage. This penalty serves to limit the process K in the halftones, like the modification to the interpolation weights discussed in Section V-A. Finally, we note that the two calibration algorithms we propose can be applied to any colorimetric halftoning algorithm and are not restricted to CDBS.

VI. Results

We present our halftone results showing the effect of the choice of color space, the dot interaction model, and the halftoning calibration algorithm on the quality of the color halftones. The halftones we present are printed with a 300-dpi Hewlett Packard 692C printer, which is a CMYK inkjet printer with full undercolor removal. The printouts are obtained by bypassing the printer driver and sending the CMY bit map from our algorithms directly to the printer. The printer is used in

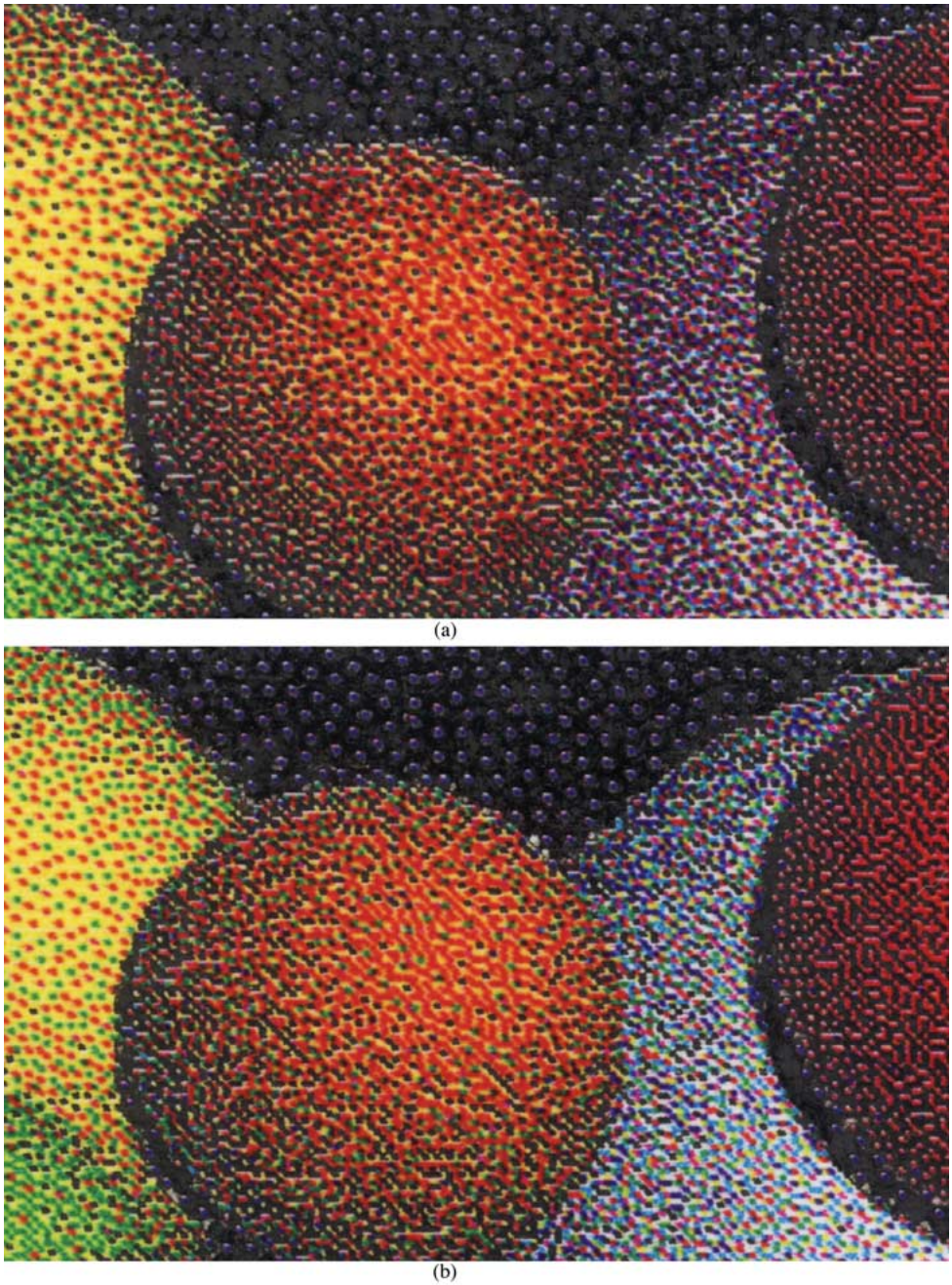

Fig. 10. The 2400-dpi scans of the 300-dpi halftones of the *faxballs* image, obtained using CDBS (a) in the RGB space (b) in the $Y_yC_xC_z$ space, with the simple RGB$\mapsto$ $Y_yC_xC_z$ calibration and printed at 150 dpi.

The printer is used in binary-only mode. The images are printed on specially coated inkjet paper (Hewlett-Packard Photo Paper).

We start with the effect of the choice of color space. To show this, we apply our CDBS halftoning algorithm in the RGB color space instead of $Y_yC_xC_z$. We use $\tilde{p}_{Y_y}(\mathbf{x})$, the HVS point spread function for the luminance channel, for all 3 color channels, we set $\kappa = 1$, and we choose our error metric to be the sum of total squared RGB errors after passing through this HVS filter. This approach is similar to the one presented by Pappas [22] for nonseparable printer models. Fig. 10 compares two halftones of the *faxballs* image, one obtained by applying CDBS in the RGB space and the other by applying CDBS in the $Y_yC_xC_z$ space. These two 300-dpi halftones are printed at 150 dpi by enlarging every halftone pixel to a 2×2 pixel area to display the difference in halftone texture. The comparison of these two low-resolution halftones shows that the use of a luminance/chrominance-based space like $Y_yC_xC_z$ gives us an overall finer, therefore more pleasing texture. The differences of the visual consequences of the chosen color space are especially noticeable in highlights like the white ball and the yellow ball.

Even though texture rendition has been improved by the choice of a luminance/chrominance-based space instead of a device dependent color space, there is room for further improve-

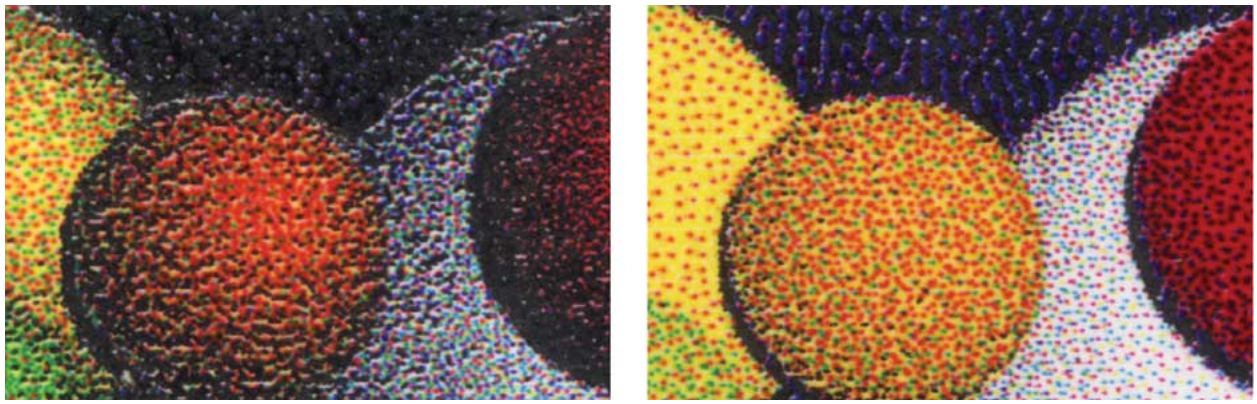

Fig. 11. The 2400-dpi scans of the halftones of the *faxballs* image printed at 300 dpi, obtained using (a) CDBS and (b) centering-based CDBS, with the simple RGB$\mapsto$ $Y_yC_xC_z$ calibration.

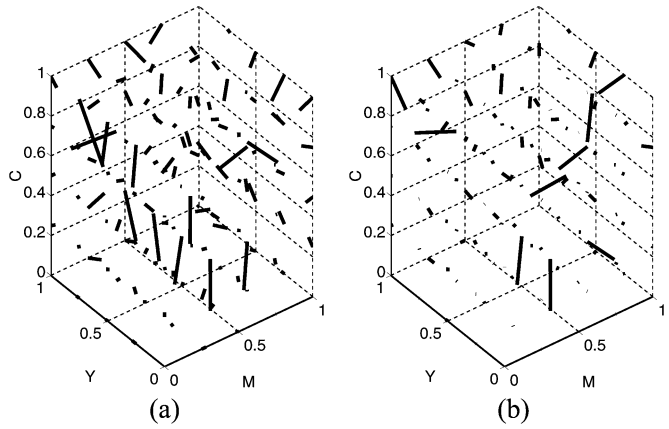

Fig. 12. $\triangle$CMY error vectors of (a) the initial $\widehat{\text{CMY}}$ grid structure and (b) the iterated $\widehat{\text{CMY}}$ grid structure.

ment, especially in the darker shadow areas of Fig. 10(b) where there are artifacts due to the interactions between dark dots neighboring light dots, e.g., in the shadow regions of the orange and red balls. This effect is more clearly visible in the same halftone printed at the native resolution of 300 dpi, as shown in Fig. 11(a).

Next, we discuss the effect of the 2×2 centering model. Comparison of Fig. 11(a) and (b) which were both created using the simple RGB $\mapsto$ $Y_yC_xC_z$ calibration shows that the 2×2 centering model has been very effective in reducing dot interaction artifacts especially in the shadow regions of the orange, red, and white balls. The dot interaction model also improves the color textures and gives us much smoother transitions between the light and dark areas. It aims to improve halftone texture and does not try to preserve lightness and saturation. Therefore, Fig. 11(b) is generally lighter and less saturated than Fig. 11(a). This side effect may be eliminated with a tone correction prior to halftoning. Both CDBS [Fig. 11(a)] and centering-based CDBS [Fig. 11(b)] with the simple RGB $\mapsto$ $Y_yC_xC_z$ calibration method give colorant-wise acceptable results for the *faxballs* image, but applying CDBS to the ramps of six Neugebauer primaries ($CMYRGB$) [Fig. 13(a)] reveals

TABLE I
COMPARISON OF INITIAL AND ITERATED $\widehat{\text{CMY}}$ GRID STRUCTURES

	Initial $\widehat{CMY}$ grid	Iterated $\widehat{CMY}$ grid
Max. $\triangle CMY$	0.3221	0.2607
Mean $\triangle CMY$	0.0703	0.0383

the need for the iterative RGB $\mapsto$ $Y_yC_xC_z$ calibration method for business graphics where the intent is to preserve the purity of colors. In Fig. 13(a), we notice *phantom* dots, i.e., dots with wrong colorant values such as magenta dots in the yellow ramp and cyan dots in the red ramp.

The $\triangle$CMY error vectors for the initial $\widehat{\text{CMY}}$ grid structure [Fig. 7(a)] and its iterated version [Fig. 9(b)] are depicted in Fig. 12, and the maximum and mean $\triangle$CMY values are summarized in Table I. The expansion and correction approach provides a significant decrease in the $\triangle$CMY errors. Comparison of the halftone of the ramps of the Neugebauer primaries obtained using centering-based CDBS with this iterated $\widehat{\text{CMY}}$ grid structure-based calibration [Fig. 13(b)] with that obtained using centering-based CDBS with the simple calibration [Fig. 13(a)] emphasizes what we have gained by incorporating the iterative calibration. The colorant rendition is almost perfect, i.e., we have considerably fewer *phantom* dots. The subtractive primary ramps (CMY) are almost solely made up of dots of the desired respective colorant. For example, there are almost no magenta dots in the cyan and the yellow ramps. Similarly, the subtractive secondary ramps (RGB) contain very few dots of the wrong colorant; e.g., the number of cyan dots in the red ramp and the number of magenta dots in the green ramp have decreased significantly. While attaining improved colorant rendition, we still maintain the visually pleasing smooth color textures of CDBS.

We conclude with a comment on one shortcoming of our halftoning method which necessitates improvement. Our error metric fails to predict the perception of a chromatic error when the luminance Y_y is very close to 0 and there is little chroma, and may lead to a few white dots in such dark areas. This can be seen on the upper portion of the perimeter of the orange ball in Fig. 11(a) and in the 150-dpi reproduction of the same image in Fig. 10(b).

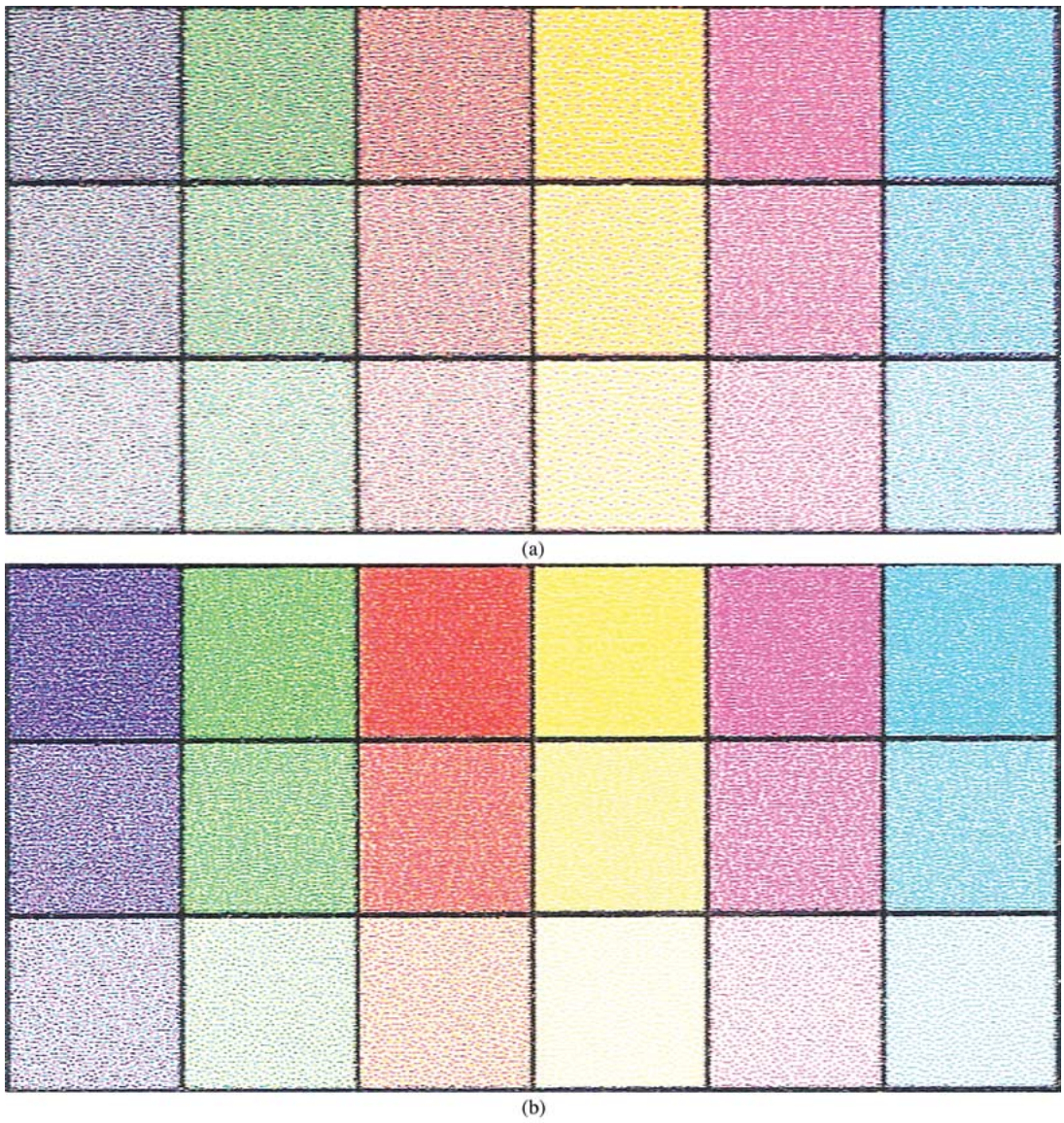

Fig. 13. The 1200-dpi scans of the halftones of the ramps of the six Neugebauer primaries printed at 300 dpi, obtained using (a) centering-based CDBS with the simple calibration and (b) centering-based CDBS with the iterated $\widehat{\mathrm{CMY}}$ grid structure-based calibration.

VII. Conclusion

We have developed a model-based color halftoning algorithm that minimizes a measure of perceived error in the $Y_yC_xC_z$ opponent color space. In this algorithm, using a luminance/chrominance-based space and incorporating a HVS model into the halftoning process gives us visually pleasing color textures. The 2×2 centering dot interaction model improves the color texture rendition especially in dark and light areas where accurate accounting of the dot interaction of minority colorants with the majority colorant is important. The DBS algorithm and the efficient search strategies provide faster implementation. Even though CDBS in the $Y_yC_xC_z$ space is a colorimetric halftoning technique that cannot guarantee a desired colorant combination, the calibration algorithm we have developed, cascaded with CDBS allows us to create color halftones with high texture quality and desired colorant composition. Our calibrated CDBS halftoning algorithm leads to color halftones with very good overall color texture quality and faithful colorant reproduction, except in very dark colors where the underlying HVS model fails to give us a good measure of perceived error.

References

[1] P. Delabastita, "Screening system and method for color reproduction in offset printing," U.S. Patent 5 155 599, 1992.

[2] I. Amidror, R. Hersch, and V. Ostromoukhov, "Spectral analysis and minimization of Moiré patterns in color separation," *J. Electron. Imag.*, vol. 3, no. 3, pp. 295–317, Jul. 1994.

[3] M. Kaji, Y. Satou, and J. Tajima, "A construction method of digital screen sets: Realization of moiré-free rational tangent screens by using the multi-unit area design method," in *Proc. IS&T PICS*, 1998, pp. 349–357.

[4] S. Wang, Z. Fan, and Z. Wen, "Non-orthogonal screens and its application in moiré free halftoning," in *Proc. SPIE, Color Imaging VIII: Device-Independent Color, Color Hardcopy, and Graphic Arts*, vol. 5008, 2003, pp. 399–408.

[5] M. Yao and K. Parker, "Color halftoning with blue noise mask," in *Proc. IS&T's 11th Int. Congr. Advances in Non-Impact Printing Technologies*, 1995, pp. 466–468.

[6] N. Damera-Venkata and Q. Lin, "AM-FM screen design using donut filters," in *Proc. SPIE, Color Imaging IX: Processing, Hardcopy and Applications*, vol. 5293, 2004, pp. 469–480.

[7] F. Baqai and J. Allebach, "Computer-aided design of clustered dot color screens based on a human visual system model," *Proc. IEEE*, vol. 90, no. 1, pp. 104–122, Jan. 2002.

[8] Q. Lin and J. Allebach, "Color FM screen design using DBS algorithm," in *Proc. IS&T/SPIE Int. Symp. Electronic Imaging Science and Technology*, San Jose, CA, 1998, pp. 353–361.

[9] D. Shaked, Z. Baharav, and Q. Lin, "G/M dither or color dither from monochrome dither matrices," *Proc. SPIE, Color Imaging IX: Processing, Hardcopy, and Applications*, vol. 5293, pp. 448–459, 2004.

[10] M. Wang and K. Parker, "Properties of jointly-blue noise masks and applications to color halftoning," *J. Imag. Sci. Technol.*, vol. 44, no. 4, pp. 360–370, Jul./Aug. 2000.

[11] D. Lau, A. Khan, and G. Arce, "Minimizing stochastic moiré by means of green noise masks," *J. Opt. Soc. Amer. A*, vol. 19, pp. 2203–2217, Nov. 2002.

[12] D. Lau, G. Arce, and N. Gallagher, "Digital color halftoning with generalized error diffusion and multichannel green-noise masks," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 923–935, May 2000.

[13] R. Klassen, R. Eschbach, and K. Bharat, "Vector diffusion in a distorted color space," in *Proc. IS&T's 47th Annu. Conf.*, 1994, pp. 489–491.

[14] G. Marcu and S. Abe, "An error diffusion algorithm for arbitrary set of output color with application to textile sewing," in *Proc. SPIE, Color Hard Copy and Graphic Arts IV*, vol. 2413, San Jose, CA, 1995, pp. 375–384.

[15] H. Haneishi, T. Suzuki, N. Shimoyama, and Y. Miyake, "Color digital halftoning taking colorimetric color reproduction into account," *J. Electron. Imag.*, vol. 5, no. 1, pp. 97–106, Jan. 1996.

[16] C. Y. Kim, I. Kweon, and Y. S. Seo, "Color and printer models for color halftoning," *J. Electron. Imag.*, vol. 6, no. 2, pp. 166–180, Apr. 1997.

[17] L. Akarun, Y. Yardimci, and A. Cetin, "Adaptive methods for dithering color images," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 950–956, Jul. 1997.

[18] G. Bozkurt, Y. Yardimci, O. Arikan, and E. Cetin, "QR-RLS algorithm for error diffusion of color images," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998, pp. 49–53.

[19] N. Damera-Venkata and B. Evans, "Design and analysis of vector color error diffusion halftoning systems," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1552–1565, Oct. 2001.

[20] V. Monga, W. Geisler, and B. Evans, "Linear color-separable human visual system models for vector error diffusion," *IEEE Signal Process. Lett.*, vol. 10, no. 5, pp. 93–97, Apr. 2003.

[21] N. Damera-Venkata, B. Evans, and V. Monga, "Color error diffusion halftoning," *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 51–58, Jul. 2003.

[22] T. Pappas, "Model-based halftoning of color images," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1014–1024, Jul. 1997.

[23] T. Flohr, B. Kolpatzik, R. Balasubramanian, D. Carrara, C. Bouman, and J. Allebach, "Model-based color image quantization," in *Proc. SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, 1993, pp. 270–281.

[24] D. Lieberman and J. Allebach, "Dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1950–1963, Nov. 2000.

[25] S. Wang, K. Knox, and N. George, "Novel centering method for overlapping correction in halftoning," in *Proc. ICPS: The Physics and Chemistry of Imaging Systems*, 1994, pp. 482–486.

[26] S. Wang, "Algorithm-independent color calibration for digital halftoning," in *Proc. 4th IS&T/SID Color Imaging Conf.*, 1996, pp. 75–79.

[27] R. S. Gentile, "Device independent color in postscript," *Proc. SPIE*, vol. 1913, pp. 419–432, Feb. 1993.

[28] K. Mullen, "The contrast sensitivity of human color vision to red-green and blue-yellow chromatic gratings," *Physiol.*, vol. 359, pp. 381–400, 1985.

[29] S. Daly, "The visible differences predictor: An algorithm for the assessment of image fidelity," *Proc. SPIE Human Vision, Visual Processing, and Digital Display III*, vol. 1966, pp. 2–15, 1992.

[30] X. Zhang, D. Silverstein, J. Farrell, and B. Wandell, "Color image fidelity metric s-CIELAB and its application on halftone texture visibility," in *IEEE Compcon Symp. Dig.*, 1997, pp. 44–48.

[31] C. Noorlander and J. Koenderink, "Spatial and temporal discrimination of ellipsoids in color space," *J. Opt. Soc. Amer.*, vol. 73, pp. 1533–1543, 1983.

[32] H. Bauml and B. Wandell, "The color appearance mixture of gratings," *Vis. Res.*, vol. 99, no. 7, pp. 1–100, 1996.

[33] A. Poirson and B. Wandell, "Appearance of colored patterns: Pattern-color separability," *J. Opt. Soc. Amer.*, vol. 10, no. 12, pp. 2458–2470, 1993.

[34] R. Näsänen, "Visibility of halftone dot textures," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-14, no. 6, pp. 920–924, Dec. 1984.

[35] S. Kim and J. Allebach, "Impact of HVS models on model-based halftoning," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 258–269, Mar. 2002.

[36] B. Kolpatzik and C. Bouman, "Optimized error diffusion for image display," *J. Electron. Imag.*, vol. 1, no. 3, pp. 277–292, Jul. 1992.

[37] R. Balasubramanian, C. A. Bouman, and J. P. Allebach, "Sequential scalar quantization of vectors: An analysis," *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1282–1295, Sep. 1995.

[38] F. Baqai and J. Allebach, "Optimal clustered-dot color screen design based on a human visual system model," *Proc. SPIE*, vol. 4300, pp. 404–415, 2001.

[39] V. Monga, N. Damera-Venkata, and B. Evans, "An input-level dependent approach to color error diffusion," in *Proc. SPIE, Color Imaging IX: Processing, Hardcopy and Applications*, vol. 5293, 2004, pp. 333–343.

[40] V. Monga and B. Evans, "Tone dependent color error diffusion," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 3, May 2004, pp. 101–104.

[41] M. Analoui and J. Allebach, "Model-based halftoning using direct binary search," in *Proc. SPIE/IS&T Symp. Electronic Imaging Science and Technology*, San Jose, CA, Feb. 1992, pp. 96–108.

[42] J. Lee and J. Allebach, "Colorant-based direct binary search halftoning," *J. Electron. Imag.*, vol. 11, no. 4, pp. 517–527, Oct. 2002.

[43] D. Lieberman and J. Allebach, "Efficient model based halftoning using direct binary search," presented at the IEEE Int. Conf. Image Processing, Santa Barbara, CA, Oct. 26–29, 1997.

[44] T. Pappas and D. Neuhoff, "Printer models and error diffusion," *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 66–80, Jan. 1995.

[45] G. Goertzel and G. Thompson, "Digital halftoning on the IBM 4250 printer," *IBM J. Res. Develop.*, vol. 31, no. 1, pp. 2–15, Jan. 1987.

[46] P. Roetling and T. Holladay, "Tone reproduction and screen design for pictorial electrographic printing," *J. Appl. Photograph. Eng.*, vol. 5, no. 4, pp. 179–182, 1979.

[47] C. L. Kim, K. M. Kim, E. Jee, and Y. H. Ha, "Color quantization and dithering method based on HVS characteristics," in *Proc. 3rd IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, 1995, pp. 90–92.

[48] D. Knuth, "Digital halftones by dot diffusion," *ACM Trans. Graphics*, vol. 6, pp. 245–273, Oct. 1987.

[49] J. Lai and C. Chen, "Color image halftoning with the dot overlap printer model," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Kobe, Japan, 1999, pp. 333–337.

[50] S. Wang, "Two-by-two centering printer model with Yule-Nielsen equation," in *IS&T's NIP 14: Int. Conf. Digital Printing Technologies*, 1998, pp. 302–305.

[51] T. Pappas, C. Dong, and D. Neuhoff, "Measurement of printer parameters for model-based halftoning," *J. Imag. Technol.*, vol. 2, no. 3, pp. 193–204, Jul. 1993.

[52] R. Rolleston and R. Balasubramanian, "Accuracy of various types of neugebauer model," in *Proc. 1st IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, Nov. 1993, pp. 32–37.

[53] H. Motomura, T. Fumoto, O. Yamada, K. Kanamori, and H. Kotera, "CIELab to CMYK color conversion by prism and slant prism interpolation method," in *Proc. 2nd IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, 1994, pp. 156–158.

[54] J. M. Kasson, W. Plouffe, and S. I. Nin, "A tetrahedral interpolation technique for color space conversion," *Proc. SPIE*, vol. 1909, pp. 127–138, 1993.

[55] J. Z. Chang, J. P. Allebach, and C. A. Bouman, "Sequential linear interpolation of multidimensional functions," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1231–1245, Sep. 1997.
[56] K. Gennetten, "RGB to CMYK conversion using 3-D barycentric interpolation," *Proc. SPIE*, vol. 1909, pp. 116–126, 1993.
[57] D. Shaked, N. Arad, A. Fitzhugh, and I. Sobel, "Ink Relocation for Color Halftones," HP Labs, Palo Alto, CA, Tech. Rep. HPL-96-127RL, 1996.

A. Ufuk Agar (M'99) received the B.S. degrees in electrical engineering and mathematics from Bogazici University, Istanbul, Turkey, in 1992, and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, in 1995 and 1999, respectively.

From 1999 to 2004, she was with the Color Imaging and Printing Technologies Department, Hewlett Packard Laboratories, Palo Alto, CA, first as a Research Scientist and then as a Senior Research Scientist. During the 2002–2003 academic year, she was an Assistant Professor with the Department of Electronics Engineering, Isik University, Istanbul. She is currently with Garanti Technology, Istanbul. Her research interests include color imaging, color halftoning, spectral modeling, colorimetric calibration, and interpolation of multidimensional functions.

Dr. Agar was a Fulbright Scholar from 1993 to 1995, and she received the Raymond Davis Scholarship from the Society for Imaging Science and Technology in 1998.

Jan P. Allebach (F'91) received the B.S.E.E. from the University of Delaware, Newark, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1972 and 1976, respectively.

He was on the faculty at the University of Delaware from 1976 to 1983. Since 1983, he has been with Purdue University West Lafayette, IN, where he is the Michael J. and Katherine R. Birck Professor of Electrical and Computer Engineering. His current research interests include image rendering, image quality, color imaging and color measurement, and document management.

Prof. Allebach is a member of the IEEE Signal Processing (SP) Society, the Society for Imaging Science and Technology (IS&T), and SPIE. He has been especially active with the IEEE SP Society and IS&T. He is a Fellow of both these societies, has served as Distinguished/Visiting Lecturer for both societies, and has served as an Officer and on the Board of Directors of both societies. He is a past Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is presently Editor for the IS&T/SPIE *Journal of Electronic Imaging*. He received the Senior (best paper) Award from the IEEE Signal Processing Society and the Bowman Award from IS&T. In 2004, he was named Electronic Imaging Scientist of the Year by IS&T and SPIE.