https://doi.org/10.2352/ISSN.2470-1173.2017.18.COLOR-042

Color Halftoning Based on Multi-Stage, Multi-Pass, Clustered-DBS*

***Weijuan Xi*[1]*, Tal Frank*[2]*, Yotam Ben-Shoshan*[2]*, Robert Ulichney*[3]*, Jan P. Allebach*[1]**

[1] ***Electronic Imaging Systems Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN-47906, U.S.A. xi9,allebach@purdue.edu;***

[2] ***Hewlett-Packard Indigo Division, Nes Ziona, Israel;***

[3] ***Hewlett-Packard Laboratories USA, Cambridge, MA 02142, U.S.A.***

Abstract

The conventional color halftoning method for laser printer is based on periodic clustered-dot halftoning screen, which visible moire and rosette artifacts may appear. In this paper, the first two color aperiodic, clustered-dot halftoning algorithms, NPAC-MS-MP-CLU-DBS and PARAWACS-MS-MP-CLU-DBS, are introduced, which have the inherent stability during the printing process to resist misregistration and the potential to avoid moire and rosette for the aperiodic property. Besides, the color management pipeline for halftoning algorithms based on NPAC (Neugebauer Primaries area coverages) also be introduced in this paper.

Introduction

Digital color halftoning is the process of generating a pattern of pixels with a limited number of colors that creates the illusion of a continuous-tone image. Color halftoning presents many problems that we encounter in monochrome halftoning. However, it also presents many problems that are unique to color, mainly due to the interactions between color planes. All color printers use a limited number of colorants, typically three colorants. Each colorant strongly absorbs in either the long-, medium-, or short-wavelength region of the visible spectrum, resulting in the perceived colors cyan, magenta, or yellow (CMY). Methods for digital color halftoning may be categorized into three groups according to the computational complexity required to render the continuous-tone image in halftone form, independent of the computation required to design the halftoning algorithm[1]: screening, error diffusion, and iterative processes.

In screening, threshold arrays are applied to each of the CMY colorant planes independently, and then the resulting halftones with different colorants are superimposed to generate a final color halftone. It is known that screening may cause artifacts such as undesirable color texture called moire pattern, due to the low frequency components of the interference between color planes. Error diffusion lends itself naturally to a vector treatment where the color components are halftoned jointly rather than independently, and hence often yields a higher level of image quality than screening does. However, straightforward vector error diffusion in the colorant space results in artifacts due to error diffusion in the colorant space results in artifacts due to error accumulation. Iterative techniques such as least squares[2] and direct binary search (DBS)[3][4] have also been applied to color halftoning. A major advantage of these approaches is that they can support a relatively complex HVS model.

Comparing with the conventional periodic clustered-dot halftone method[12], the new aperiodic clustered-dot algorithms have the advantage of resisting visible morie and rosette. So far, the color aperiodic clustered-dot halftoning methods are based on the screen for monochrome aperiodic clustered-dot halftoning, and there are three aperiodic clustered-dot halftone methods are presented: In 1990, Daneil Lau generated the ”Green-Noise Digital Halftoning” method[16][18]; In 2004, Damera Venkata presented the ”AM-FM Screen Design” method[17]; In 2013, Goyal presented the CLU-DBS (clustered-dot Direct Binary Search) algorithm[6]. As much as we are aware, there exist no method for evaluating and comparing the quality of aperiodic clustered-dot halftone, so that Gupta generated a psychophysical experiment in his unpublished dissertation in 2010, which expressed the CLU-DBS algorithm always generated the most smooth and homogeneous clustered halftone.

The new color halftoning algorithms that will be introduced in this paper are based on CLU-DBS. Firstly, the background knowledge about MS-MP-CLU-DBS (Multi-Stage, Multi-Pass, clustered-dot Direct Binary Search) will be introduced; After that, I will present the color management pipeline for halftoning algorithms based on NPAC briefly; Thirdly, the new color halftoning algorithm, NPAC-MS-MP-CLU-DBS, is presented; Lastly, during the introduction of PARAWACS-MS-MP-CLU-DBS algorithm, a new screen design method are introduced. The comparison between the conventional culstered-dot halftoning method and the two new algorithms will be shown in the conclusion.

PRELIMINARIES: MS-MP-CLU-DBS Algorithm

DBS (Direct binary search) is an iterative halftone manipulation-based algorithm that achieves a homogeneous, dispersed-dot texture by minimizing a perceptual-error-based cost metric based on two kinds of operation: toggle and swap. CLU-DBS is a variant of DBS, which uses a dual-filter-based cost metric to generate clustered-dot texture. Throughout this paper, we use $[m] = [m,n]^T$ to represent the discrete spatial coordinates and $f[m]$ to denote the original discrete-space continuous-tone image. Each pixel of $f[m]$ takes on values of absorptance between 0 and 1, whereas each pixel of the halftone $g[m]$ takes on values 0 (white) or 1 (black) only. The error in the halftone is given by $e[m] = g[m] - f[m]$. The cost metric is given by

$$\theta = \theta_{homog} - \theta_{clust}, \tag{1}$$

[0]*Research sponsored by HP Inc., Palo Alto, CA

where,

$$\theta = \sum_m e[m] c^u_{\tilde{p}\tilde{p}}[m] \quad (2)$$

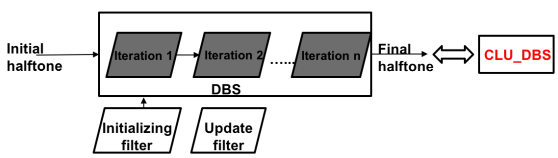
$$\theta_{clust} = 2\sum_m e[m] \Delta c_{\tilde{p}\tilde{e}_0}[m], \quad (3)$$

$$c^u_{\tilde{p}\tilde{e}}[m] = \sum_n e[n] c^u_{\tilde{p}\tilde{p}}[m-n], \quad (4)$$

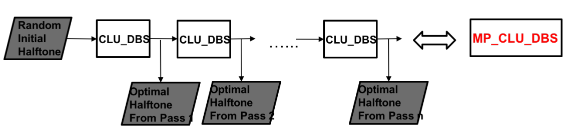
$$\Delta c_{\tilde{p}\tilde{e}_0}[m] = \sum_n e[n] c^u_{\tilde{p}\tilde{p}}[m-n], \quad (5)$$

In the above equations, $e_0[m]$ is the initial halftone error $c^i_{\tilde{p}\tilde{p}}[m]$ is the difference between the initialization-filter $c^i_{\tilde{p}\tilde{p}}[m]$ and update-filter $c^u_{\tilde{p}\tilde{p}}[m]$. Prior to optimizing the halftone given as input to the algorithm, the $c^u_{\tilde{p}\tilde{p}}[m]$ and $\Delta c_{\tilde{p}\tilde{e}_0}[m]$ LUTs are both initialized using $e_0[m]$. In equation (1), θ_{homog} encourages the formation of homogeneous texture and θ_{clust} encourages the formation of dot-clusters. The term θ_{homog} is computed by filtering the error image $e[m]$ with the update filter $c^u_{\tilde{p}\tilde{p}}[m]$ (Eq. (4)), and then forming the inner product of this filter output with $e[m]$ (Eq. (2)). The term θ_{clust} is computed similarly (Eqs. (3) and (5)), except that the filter $\Delta c_{\tilde{p}\tilde{p}}[m]$ is the difference between the initialization and update filters, and the filtering is performed on the initial error $e_0[m]$.

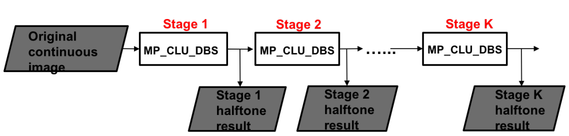
There are two refinement operations: Multi-Stage process and Multi-Pass process[5]. We have observed that the structure of the initial halftone strongly affects the cluster distribution in the output halftone. For the Multi-Pass process, a pass refers to a complete optimization of the halftone image using multiple iterations with the basic CLU-DBS algorithm; For Multi-Stage process, the optimization is performed in several stages, and at each stage the halftone is optimized with respect to an attenuated version of the original input image. The black diagram for MS-MP-CLU-DBS is shown in Fig. 1.



(a) block diagram for CLU-DBS



(b) block diagram for MP-CLU-DBS



(c) block diagram for MS-MP-CLU-DBS

Figure 1. block diagram for MS-MP-CLU-DBS: (a) is the block diagram for CLU-DBS; (b) is the block diagram for MP-CLU-DBS; (c) is the block diagram for MS-MP-CLU-DBS

During optimization, the halftone image is iteratively scanned in raster order, from left to right and top to bottom. At each pixel, we evaluate the effect of toggling its state or swapping its value with other pixels in a 3×3 neighborhood centered at that pixel location. If any toggle or swap decreases the cost metric, that change which decreases the cost metric the most is accepted. An iteration is complete when every pixel in the image has been visited. When no change is accepted during an iteration, the algorithm is said to have converged to a local minimum of the cost metric.

Color Management Pipeline for NPAC

The color management pipeline for NPAC contains color management and halftoning two parts. Traditionally, color separation consisted in answering the question of how much of each available colorant to use for matching each color within a printing system's color gamut and halftoning was then responsible for making spatial choices of where to apply each colorant in turn, given the choices of colorant amounts made during halftoning. In this section, a color separation method based on NPAC (Neugebauer Primaries area coverages) is introduced. The pipeline is shown in Fig. 2. Three steps are needed in order to get the NPAC for each color: Color space transformation; Color gamut mapping; Color Separation Based on Neugebauer Model and Tetrahedral Interpolation. The fourth step linearization is needed to correct the dot-gain whenever print color images. Let's focus on the first three steps, and follow them step by step.

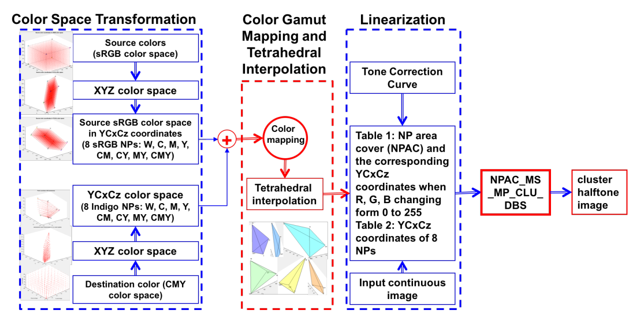


Figure 2. The pipline of color management for NPAC

Color space transformation: To get NPAC for each sRGB color, firstly, we define the sRGB color space as source color space, and Indigo press color space as destination color space. $Y_yC_xC_z$ color space[7] is used because it is a linearized version of CIE Lab, and its linear structure is suitable for halftoning, because $Y_yC_xC_z$ color space preserves local averages. Y channel is the correlate of luminance, C_x channel is the Red-Green opponent chrominance channel and C_z channel is the Yellow-Blue opponent color chrominance channel. So, before doing the Color Gamut Mapping, color space transformation is done. Therefore, the transformation to source and destination $Y_yC_xC_z$ color space is shown below.

Our source colors are sRGB colors in sRGB color space, which need to be transfered to XYZ color space, then to $Y_yC_xC_z$ color space. Our data about the destination colors are getting from the reflectance file based on Indigo press, which can let us get the XYZ coordinates of NPs directly. Similarly, we can also based on the XYZ coordinates of NPs to get $Y_yC_xC_z$ coordinates of NPs.

The formula of transform linear *sRGB* to XYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (6)$$

where,

$$[M]^{-1} = \begin{bmatrix} 3.1338561 & 1.6168667 & 0.4906146 \\ 0.9787684 & 1.9161415 & 0.0334540 \\ 0.0719453 & 0.2289914 & 1.4052427 \end{bmatrix}$$

And the formula of transferring from XYZ to $Y_yC_xC_z$ color space:

$$\begin{bmatrix} Y_y \\ C_x \\ C_z \end{bmatrix} = \begin{bmatrix} \frac{1}{X_w} & 0 & 0 \\ 0 & \frac{1}{Y_w} & 0 \\ 0 & 0 & \frac{1}{Z_w} \end{bmatrix} \begin{bmatrix} 0 & 116 & 0 \\ 500 & -500 & 0 \\ 0 & 200 & -200 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (7)$$

where, X_w, Y_w, Z_w is the D_{50} white point.

Here, NPs are referred to as device states. In this paper, we only consider Cyan, Magenta, Yellow toner of HP Indigo division and their combination as the destination primaries. So that, we have 8 NPs in total: white, yellow, cyan, cyan and yellow (green), magenta, magenta and yellow (red), cyan and magenta (blue) and cyan and magenta and yellow (black), we label the NPs as $W, Y, C, CY, M, MY, CM, CMY$. Based on the reflectance spectra getting from HP Indigo division, and the color matching function (CMF), we can get the XYZ coordinates of 8 destination NPs. Then, following the same process from XYZ to $Y_yC_xC_z$ color space. The $Y_yC_xC_z$ coordinates of source and destination NPs are shown in Table 1 and Table 2.

Since, $Y_yC_xC_z$ is a linearized color space, the $Y_yC_xC_z$ coordinates of any sRGB colors are in the range or on the surface of polyhedrons, which means that all the sRGB can be reproduced by 8 source NPs. All the colors in destination color space are the combination of 8 destination NPs, so that the $Y_yC_xC_z$ coordinates of destination colors should also in the range or on the surface of polyhedrons composed by 8 destination NPs in theory. However, from Fig. 3, we can see the mismatch in destination $Y_yC_xC_z$ color space, which is caused by printer. We just ignore this at present.

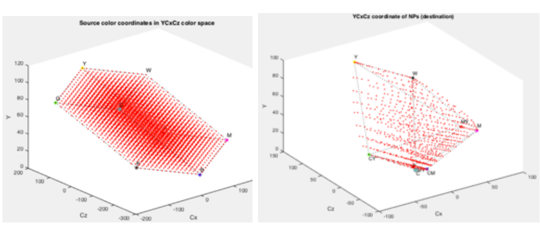


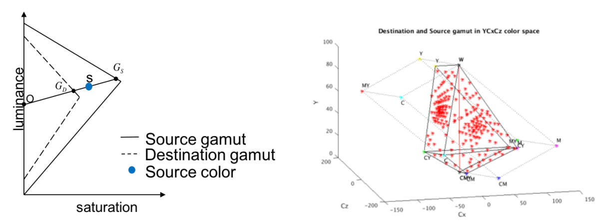
Figure 3. *Color gamut of source and destination $Y_yC_xC_z$ color space*

Color gamut mapping: We have get the $Y_yC_xC_z$ coordinates of all the sRGB colors in source $Y_yC_xC_z$ color space, the next step is to do the color gamut mapping. As we all know that a color can be expressed in polar coordinates as lightness, hue and saturation. The basic idea during our color gamut mapping is remaining the hue unchanged, and scale the lightness and saturation equally. To be reasonable, firstly, we should mapping the black and white NPs between the source and destination $Y_yC_xC_y$ color space by rotate and translate the source $Y_yC_xC_z$ color space.

The most popular building blocks of Gamut Mapping Algorithms (GMAs) are mapping along a path[14]. As for our mapping purpose, we can choose the linear mapping method along a path. Let O indicate the origin of the path, we choose O as the midpoint of W and CMY, $\overrightarrow{OS}$ the distance of the source color from the origin along the path, $\overrightarrow{OG_S}$ the distance of a source gamut boundary intersection with the path, $\overrightarrow{OG_D}$ the distance of an intersection of the destination gamut boundary, $\overrightarrow{OD}$ the distance computed for the destination color. So that, the source distances are linearly scaled to fit into the destination range as follows:

$$\overrightarrow{OD} = \frac{\overrightarrow{OS}}{\overrightarrow{OG_S}} \cdot \overrightarrow{OG_D} \quad (8)$$

Based on this GMA, we can find the corresponding $Y_yC_xC_z$ for all the sRGB colors in $Y_yC_xC_z$ destination color space. Let red dots represent the mapping color in destination $Y_yC_xC_y$ color space. Fig. 4 shown the color gamut mapping result, in which we can see all the mapped colors are in the gamut of destination $Y_yC_xC_z$ color space.



(a) 2-D schematic diagram (b) Color gamut mapping result

Figure 4. *Color gamut mapping result. (a) is the 2-D schematic diagram to show the process of color gamut mapping; (b) show the mapping result in $Y_yC_xC_z$ color space*

Color Separation Based on Neugebauer Model and Tetrahedral Interpolation: The color of a halftone pattern is the convex combination of the colors of the Neugebauer primaries (NPs) used in it. Here, an NP is one of the possible ink overprints, with its convex weight being the relative area covered by it[8]. In other words

$$T_c = \sum_{i=1}^{k^n} T_i a_i \quad (9)$$

where, $T \in Y_y, C_x, C_z$ is a tristimulus value which expresses the $Y_yC_xC_z$ coordinates of combination color, $a_i \in [0,1]$ is a relative area coverage (i.e.,$\sum a_i = 1$), i indicates the ith NP, c denotes the resulting halftone, n is the number of inks, and k is the number of levels per ink (e.g., 2 for a bilevel system).

We have got the $Y_yC_xC_z$ coordinates of 8 destination NPs, since the linear property of $Y_yC_xC_z$ color space, the *Tetrahedral Interpolation*[13] method can be used to determine the NP percentages of each color in halftone. The tetrahedral interpolation slices the destination color gamut into several tetrahedrons, and each tetrahedron has four flat triangle surfaces. Actually there are many ways to divide a color gamut into tetrahedrons, one of the way used in this paper is shown in Fig. 5. We divided the destination color gamut into 7 tetrahedrons, which are labeled by vertices as: W,C,CY,CMY; W,Y,CY,CMY; W,Y,MY,CMY; W,M,MY,CMY; W,C,CM,CMY; W,M,CM,CMY; W,C,M,CM.

Table 1: $Y_yC_xC_z$ **coordinates of source NPs**

NPs	W	C	M	Y	CM	CY	MY	CMY
Y_y	116	90.190	32.842	108.968	7.032	83.158	25.811	0
C_x	0	-114.876	158.762	-43.886	43.886	-158.762	114.876	0
C_z	0	-41.124	-119.841	160.965	-160.965	119.841	41.124	0

Table 2: $Y_yC_xC_z$ **coordinates of destination NPs**

NPs	W	C	M	Y	CM	CY	MY	CMY
Y_y	98.480	26.524	20.353	84.922	2.851	19.784	19.458	2.176
C_x	0.0	-36.830	90.901	-12.296	7.809	-49.832	82.719	1.229
C_z	0	-77.928	3.294	130.052	-22.565	21.001	29.505	-0.183

For a given color S, whose gamut mapping coordinates is $(S_{Y_y}, S_{C_x}, S_{C_z})$, compute its barycentric coordinates with respect to each of the tetrahedron. Since each tetrahedral has 4 vertices ($V_i (i \in [1,4])$), it is done as follows:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}^T = \begin{pmatrix} S_{Yy} \\ S_{Cx} \\ S_{Cz} \\ 1 \end{pmatrix}^T \times \left(\begin{pmatrix} V_{1Yy} & V_{2Yy} & V_{3Yy} & V_{4Yy} \\ V_{1Cx} & V_{2Cx} & V_{3Cx} & V_{4Cx} \\ V_{1Cz} & V_{2Cz} & V_{3Cz} & V_{4Cz} \\ 1 & 1 & 1 & 1 \end{pmatrix}^T \right)^{-1} \quad (10)$$

Equation (10) is the inverse of the system formed by three equations expressing how S is the convex combination of $V_1 - V_4$ and the equation of $b_1 - b_4$ summing to one. If all members of b are from $[0,1]$, S is inside the tetrahedron. Furthermore, since b represents the normalized volumes of subtetrahedra formed by sets of three original vertices and S (for the original vertex that S substitutes), they are also the relative area coverages with which tetrahedron vertex NPs need to be combined. Repeat this process for each sRGB colors, we can get the look up table (LUT) which save the NPAC for all the sRGB colors.

NPAC-MS-MP-CLU-DBS

MS-MP-CLU-DBS, as a monochrome halftoning algorithm, can only process one primary at a time. However, as a color halftoning algorithm, the purpose of NPAC-MS-MP-CLU-DBS is halftoning the color image and express the halftone result in 8 NPs. In order to solve this contradiction, the basic principle of NPAC-MS-MP-CLU-DBS is processing NPs one-by-one by iteration in a default order. Based on our intuitive experience, for a color cluster halftone, we prefer the cluster center is darker, and the cluster edge is lighter, and out of the cluster should be paper white. So that, we default the NP order as: CMY, CM, MY, M, CY, C, Y, W, and they are labeled as 1, 2, 3, 4, 5, 6, 7, 8, based on the luminance values from lowest to highest almostly, which means the CMY primary will be generated in the cluster center firstly, and Y primary will be generated lastly. We don't need to do anything for W, since W corresponding to the blank substrate. The block diagram of NPAC-MS-MP-CLU-DBS algorithm is shown in Fig. 6. The process of the new algorithm can be divided into 3 parts: Seed-Halftone Generation, Accumulation NPAC Image Generation, and NP Processing Separately.

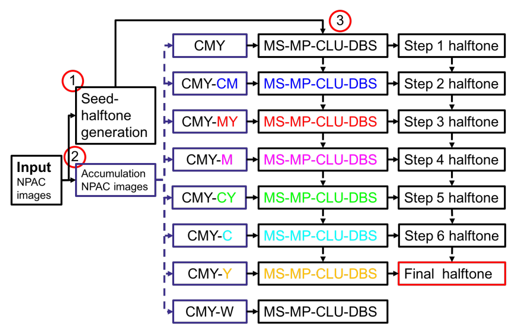


Figure 6. block diagram of NPAC-MS-MP-CLU-DBS

A. Seed-Halftone Generation

The spatial frequency of clusters in a clustered-dot halftone is a very important metric because it relates to the average cluster size, and hence influences texture visibility and print stability. Seed-halftone required for CLU-DBS halftoning of any input image is a homogeneous dispersed-dot halftone of the same size as the input image and with average dot-density determined by the intended degree of coarseness. If the desired lpi of a halftone is ρ, the printer resolution is R, then the desired density of clusters in that halftone is

$$\delta = (\rho / R)^2 \quad (11)$$

which is the same as the density of dots required in that halftone because each dot is supposed to be a cluster center. The seed halftone can be designed using conventional DBS.

However, as a color halftone image, it is not reasonable to default the dots in seed halftone as any constant NP, so that the NP generation for seed halftone is needed. Since, the NPs will be processed in the default order during the DBS algorithm, we can

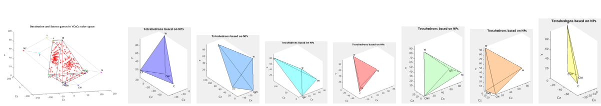


Figure 5. The destination color gamut is divided into 7 tetrahedrons

also update NPs for seed halftone in the same order from CMY to W (from 1 to 8).

For the green pattern shown in Fig. 7, the NPAC of green ($sRGB(0,255,0)$) contains $54.63\% CY$, $7.8\% Y$ and $37.57\% W$, and 0% other NPs. Following the default NPs order, we should check the percentage of CMY firstly and percentage of W lastly. Then, the first NP which percentage larger than 0 should be CY, so that all the dots in seed halftone should be updated as CY.

Special emphasis is needed at here. Our example is based on a green pattern, which only contain one sRGB color, so that we can define the corresponding NP for all dots in seed-halftone in one time. Actually, we can following this method to define the NPs for each dot in any color images. The only difference is that we should generate the NPs for dots in seed-halftone one-by-one. And the NPs of dots in seed-halftone may be same or different.

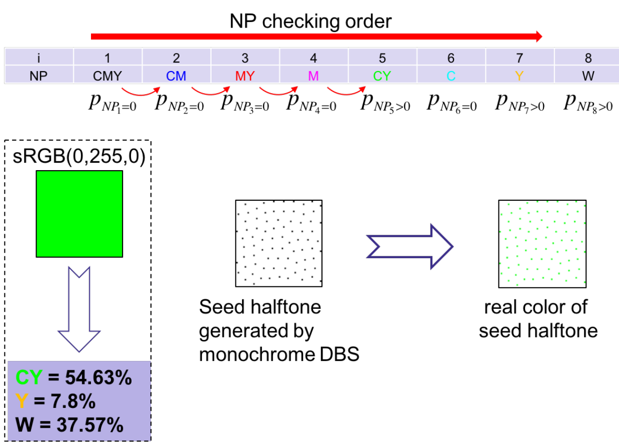


Figure 7. *seed halftone generation*

B. Accumulation NPAC images Generation

As the input of NPAC-CLU-DBS algorithm, NPAC images are the 8 images corresponding to the percentage of 8 NPs. Using the saturation of NPAC images represent the percentage of each NP. Here, a new concept **"Accumulation NPAC image"** is introduced in order to generate color halftone NP-by-NP in iteration. Using the luminance of "Accumulation NPAC images" to express the percentage of NPs, then, the luminance of i^{th} "Accumulation NPAC Image" is the percentage accumulation of NPAC images from the first NP to i^{th} NP according to the default NPs order. It means that we have 8 accumulation NPAC images in all, denote them as CMY, $CMY-CM$, $CMY-MY$, $CMY-M$, $CMY-CY$, $CMY-C$, $CMY-Y$ and $CMY-W$ in order.

For the same green pattern example, which contains 54.63%CY, 7.8%Y, 37.57%W, the NPAC images and accumulation NPAC images are shown in Fig. 8. The first row patterns are the 8 NPAC images, in which the shading patterns represent 0 percent corresponding NP, and the saturation of CY, C and W NPAC images express the percentage of CY, Y and W. The second row patterns are the 8 accumulation NPAC images. The luminance of accumulation NPAC images represent the accumulation percentage of NPs. Since, the percentage of CMY, CM, MY and M is 0, the luminance of first four accumulation NPAC images: CMY, $CMY-CM$, $CMY-MY$ and $CMY-M$ should be 0 and we express them using black patterns. Since, the percentage of CY is 54.63%, the luminance of fifth accumulation NPAC image, $CMY-CY$, should be 0.5463. Following this method, we can get all the 8 accumulation NPAC images. The luminance of last accumulation NPAC image, $CMY-W$, should always be 1, since the percentage sum of all NPs should always be 100%.

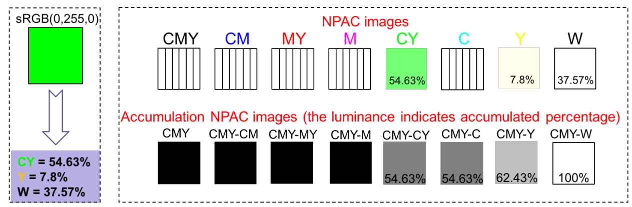


Figure 8. *NPAC images and accumulation NPAC images for green pattern*

Accordingly, the $Y_yC_xC_z$ value during the MS-MP-CLU-DBS algorithm should only be expressed by $Y_yC_xC_z$ coordinates of W and CMY. The update for $Y_yC_xC_z$ coordinates of NPs is needed. The updated $Y_yC_xC_z$ coordinates is shown in Table 3. The $Y_yC_xC_z$ coordinates of each accumulation NPAC images should be updated as:

$$Y_yC_xC_{zi} = \sum_{j=1}^{i} p_{NP_j} \cdot CMY_{Y_yC_xC_z} + \left(1 - \sum_{j=1}^{i} p_{NP_j}\right) \cdot W_{Y_yC_xC_z} \tag{12}$$

For the same example of green pattern, the $Y_yC_xC_y$ coordinate corresponding to i^{th} NP is shown in Fig. 9.

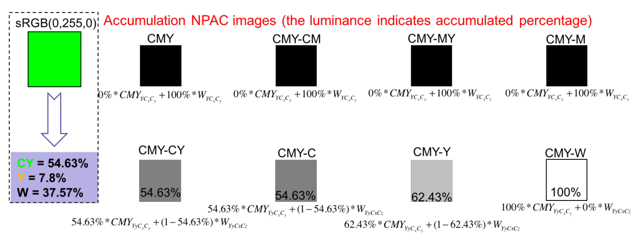


Figure 9. *update the $Y_yC_xC_z$*

C. NPs Processing Separately

We have got the accumulation NPAC images and the color seed halftone until now, the next part is using the MS-MP-CLU-DBS sequence to process NPs in default order separately. During this process, 7 MS-MP-CLU-DBS step is needed. Let's define a single MS-MP-CLU-DBS process is a single NP step.

As we have known, the NPAC according to each sRGB color is generated based on tetrahedral interpolation, so that the mapping NPAC of sRGB color can only contain at most 4 NPs. It means that, as for any sRGB color, 8 NPs can not be generated with equal probability. So that a constrain for NP generation is needed.

For each pixel in image, we process $[m,n]$ pixel during i^{th} NP step if and only if:

$$p(NP_i)_{[m,n]} > 0 \tag{13}$$

Table 3: updated $Y_yC_xC_z$ coordinates of destination NPs

NPs	CMY	CM	MY	M	CY	C	Y	W
Y_y	2.176	2.176	2.176	2.176	2.176	2.176	2.176	98.48
C_x	1.229	1.229	1.229	1.229	1.229	1.229	1.229	0.0
C_z	-0.183	-0.183	-0.183	-0.183	-0.183	-0.183	-0.183	0.0

where, NP_i is the i^{th} NP in default order: CMY, MY, CY, CM, Y, M, C, W; p is the percentage of NP_i.

Similar to CLU-DBS, the NPAC-MS-MP-CLU-DBS also work by evaluating the error metric after each swap or toggle, but process each NP separately in 7 different NP steps. We evaluate the effect of toggle and swap on error metric in the nearest 5×5 neighbor pixels. If the trial toggle or swap reduce the error metric, we accept it. This process is repeated for each pixel and each NP until there's no more changes accepted.

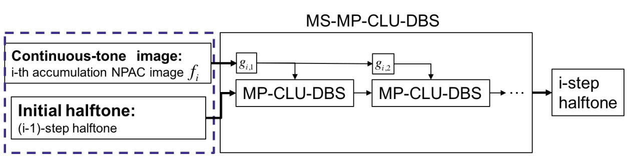


Figure 10. Block diagram for a single NP step of NPAC-CLU-DBS

Let's focus on a single NP step process, the block diagram of i^{th} NP step is shown in Fig. 10. Let f_i be the i^{th} accumulation image and $g_{i,j}$ be the input continuous-tone image corresponding to i^{th} NP step and j^{th} stage in $Y_yC_xC_z$ color space, α_j is the attenuation factor corresponding to j stage, then we have the relationship:

$$g_{i,j} = (1-\alpha_j)\cdot f_{i-1} + \alpha_j \cdot f_i \tag{14}$$

the i^{th} NP step, j^{th} stage error image between $g_{i,j}$ and the corresponding halftone image $h_{i,j}$ is:

$$e_{i,j}[m] = f_{i,j}[m] - g_{i,j}[m] \tag{15}$$

We denote the two filters as $c^{i}_{\tilde p\tilde p}[m]$ and $c^{u}_{\tilde p\tilde p}[m]$, corresponding to the initialization step and the update step, respectively.Given this notation, the i^{th} iteration cost matric of NPAC-MS-MP-CLU-DBS is given by:

$$\theta_{i,j} = \theta_{i,j_{homog}} - \theta_{i,j_{clust}}, \tag{16}$$

$$\theta_{i,j_{homog}} = \sum_m e_{i,j}[m]\cdot c^u_{\tilde p\tilde e}[m], \tag{17}$$

$$\theta_{i,j_{clust}} = 2\sum_m e_{i,j}[m]\Delta c_{\tilde p\tilde e0}[m], \tag{18}$$

$$c^u_{\tilde p\tilde e}[m] = \sum_n e_{i,j}[n]c^u_{\tilde p\tilde p}[m-n] \tag{19}$$

$$\Delta c_{\tilde p\tilde e0}[m] = \sum_n e_{i,j_0}[n]\Delta c_{\tilde p\tilde p}[m-n] \tag{20}$$

where, $\Delta c_{\tilde p\tilde p}[m] = c^i_{\tilde p\tilde p}[m] - c^u_{\tilde p\tilde p}[m]$.

The effect of a trial toggle operation at m_0 and a trial swap between m_0 and m_1 operation is given by

$$\Delta\phi_{tog} = a_0^2\cdot\Delta C_{pp}[0] - 2a_0\cdot C^u_{\tilde p\tilde e}[m_0] - 2a_0\cdot\Delta C_{\tilde p\tilde e0}[m_0] \tag{21}$$

$$\begin{aligned}\Delta\phi_{swap} = {} & \Delta\phi_{tog} + b_0^2\cdot\Delta C_{pp}[0] - 2b_0^2\cdot\Delta C_{\tilde p\tilde p}[m_1-m_0] \\ & + 2b_0\cdot C^u_{\tilde p\tilde e}[m_1] + 2b_0\Delta C_{\tilde p\tilde e0}[m_1]\end{aligned} \tag{22}$$

where,$a_0 = CMY_{Y_yC_xC_z} - W_{Y_yC_xC_z}$ when $g_{i,j} = W$ before the toggle, and $a_0 = W_{Y_yC_xC_z} - CMY_{Y_yC_xC_z}$ otherwise; And $b_0 = Y_yC_xC_{z_{i,j}}[m_1] - Y_yC_xC_{z_{i,j}}[m_0]$.

Each NP CLU-DBS step will output a monochrome halftone image corresponding to this NP step. If the percentage of i^{th} NP is not 0, there should be some dots added in the halftone from $(i-1)^{th}$ to i^{th} halftone. The color of added dots in i^{th} NP step should be defined as the color of i^{th} NP.

Summarily, Let's use a special example to show the process of NPAC-CLU-DBS. Assume, the NP separation of one color contains 0% CMY, 10% CM, 20% MY, 10% M, 20% CY, 10% C and 20%Y, 10% W. The process using NPAC-CLU-DBS to halftone the pattern of this color is shown in Fig. 11.

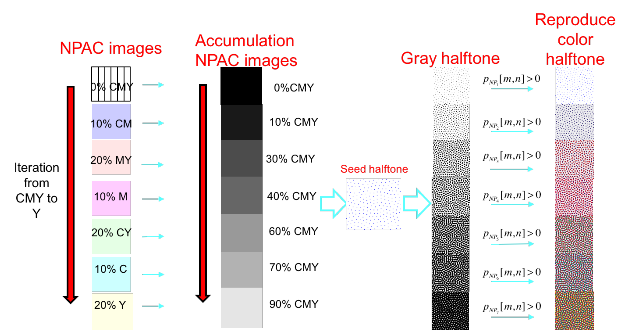


Figure 11. Special example for the process of NPAC-MS-MP-CLU-DBS

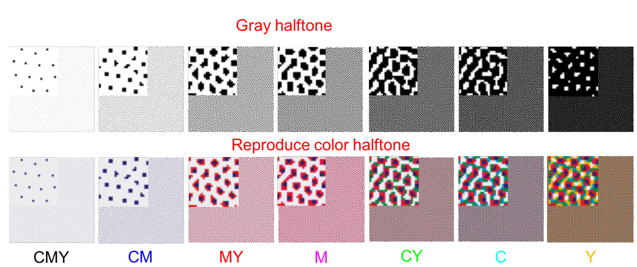


Figure 12. Halftone detail about special example

In Fig. 11, firstly, we should generate the 8 NPAC images (the first column patterns) according to the NP percentage, then based on the method that we introduced previous, we can get the 8 accumulation NPAC images and only the first 7 accumulation NPAC images (the second column patterns) will be used, since the last NP is white and the last accumulation NPAC image always be 100% white. Thirdly, process the CLU-DBS sequence based on 7 accumulation NPAC images. After each single NP step, we can get a halftone pattern corresponding to the input accumulation NPAC image, and the added dots should be colored as the NP color of current step. In order to show the color process clearly, the zoom in images of the third and fourth column patterns are shown in Fig. 12. In Fig. 12, we can see that the halftone corresponding to the first NP step is same with the seed halftone and the dots in this halftone is colored blue, it's because we have 0%CMY so that nothing in the first NP step can be changed; From CMY step to CM step, we have 10%CM so that some dots are added in the second halftone and the added dots are colored blue (CM). Same with this process, we will also generate dots corresponding to MY, M, CY, C and Y. The halftone result that we get from the Y step CLU-DBS should be our final halftone.

Let's based on this new algorithm to halftone a color image. In Fig. 13, (a) is the original continuous-tone image; (b) shown the 8 NPAC images based on the color management NP percentage; (c) is the generated 8 accumulation NPAC images based on the 8 NPAC images, and (d) is the 7 halftone images corresponding to each NP step based on NPAC-CLU-DBS algorithm, and comparing with the original continuous-tone image.

PARAWACS-MS-MP-CLU-DBS

A. PARAWACS (Parallel Random Weighted Area Coverage Selection)

PeterMorovič[11] generated the PARAWACS halftone method, which do the color halftoning with a single selector matrix based on the formula:

$$NP[m,n] = \begin{cases} NP_1 & p_{NP_1}[m,n] <= t[m,n] \\ NP_i & \sum_{i=1}^{i} p_{NP_{i-1}}[m,n] > t[m,n] \\ & and\, p_{NP_i}[m,n] <= t[m,n] \end{cases} \quad (23)$$

where, $p_{NP_i}[m,n]$ is the percentage of NP_i at $[m,n]$ pixel; $t[m,n]$ is the value of selector matrix value at $[m,n]$.

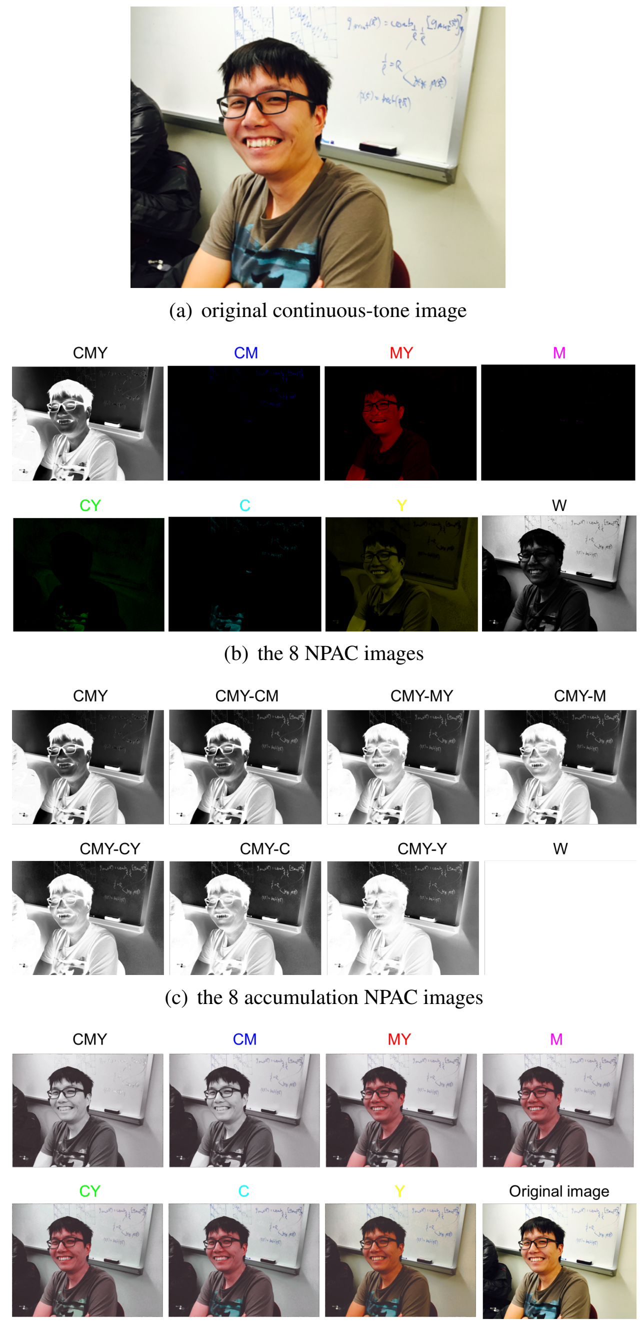
From the formula, we know that the result of PARAWACS can be affected a lot by the order of NPs. We define the default NPs order for PARAWACS-MS-MP-CLU-DBS as W, Y, C, CY, M, MY, CM, CMY, almost be the increasing order of luminance. For example, if We have a 2×2 PARAWACS screen shown in Fig. 4.1, and a uniform pattern which each pixel contains 60% C, 40% M. Then, the PARAWACS matrix is the threshold matrix for accumulative percentage of NPs, which shown in previous formula. Based on the PARAWACS halftone method, the halftone result should be the right figure in Fig. 14.

Now, we have had the NPAC of all the sRGB colors, and we also have the PARAWACS algorithm, which can halftone based on a single matrix. Our next step should be generating a reasonable screen for color aperiodic cluster-dot halftone, which is also the most important part of PARAWACS-MS-MP-CLU-DBS.

B. Screen Design for PARAWACS-MS-MP-CLU-DBS

Here, we design a 256×256 screen, which contains the integer from 1 to 255 corresponding to 255 gray levels from highlight to black. The level-by-level screen design method is using at here to generate screen based on MS-MP-CLU-DBS. The basic idea of level-by-level screen design is developing a screen by sequential design of dot profiles for each level, while obeying stacking constraints imposed by the previously designed level. Different sequences have been proposed in previous literature for best results.

Puneet Goyal generate screen for clustered-dot halftone according to the sequence 127, 126, 125, ..., 1; 128, 129, ..., 254. Here, I want to introduce a novel screen design method with double seed halftone according to a zigzag gray level sequence. You can see the block diagram of this method in Fig. 15. Before the introduction of new method, let's have a brief review about



(a) original continuous-tone image

(b) the 8 NPAC images

(c) the 8 accumulation NPAC images

(d) the 7 halftone images corresponding to each NP step

Figure 13. *Halftone a color image based on NPAC-MS-MP-CLU-DBS algorithm*

how to default the gray level of seed halftone for MS-MP-CLU-DBS algorithm. There is a relationship between seed absorptance δ, desired halftone frequency $\rho(lpi)$ and target printer resolution $R(dpi)$: $\delta = (\rho/R)^2$.

Generate double seed halftone: The double seed halftones are generated by monochrome DBS. Firstly, let the absorptance of the input image of monochrome DBS be δ, generate a dispersed-dot halftone pattern which can be regard as the lower seed halftone, it's actually the cluster center of our screen; Secondly, double the absorptance of input image to 2δ and remain the dots in lower seed halftone unchanged, we can get a new halftone in which the added dots form the higher seed halftone, and the dots in higher seed halftone is the hole center of our screen. An example of getting lower and higher seed halftone from double seed halftone is shown in Fig 16. Next, Based on the MS-MP-CLU-DBS algorithm, we should improve the gray level of haltone to $32/255$. For the higher seed halftone, the real gray level that it express is $1-32/255$.

Designing dot profiles for all gray levels: Once the dot profile for absorptance level $32/255$ and $1-32/255$ is optimally designed, the design of dot profiles for all subsequent levels has to obey the stacking constraint. It looks like that we are generating two halftones level-by-level but there is no overlap between

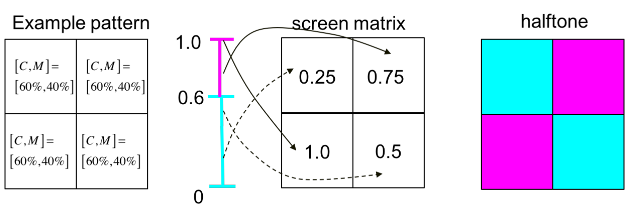


Figure 14. *An halftone example of PARAWACS algorithm. The left uniform pattern has 2×2 pixels. Based on the screen matrix shown in the middle, we can get the halftone result based on PARAWACS screen corresponding to a color with 60%C and 40%M*

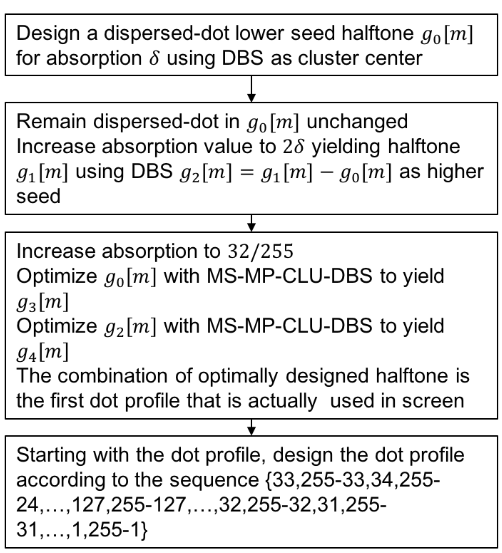
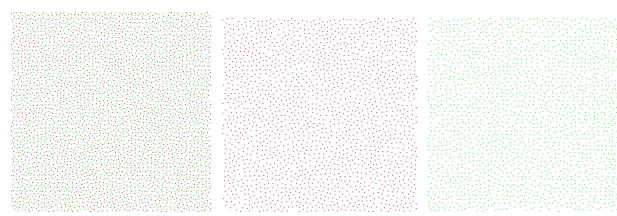


Figure 15. *Block diagram of the PARAWACS-MS-MP-CLU-DBS screen design algorithm*

dots in the two halftone. Combine the two halftones together, we can get the final screen. The screen value between 1 and 127 is generated according to the dots in lower seed halftone, and the screen value between 128 and 255 is generated according to the dots in higher seed halftone. Since the CLU-DBS cost metric preserves the distribution and shape of clusters with respect to the initial halftone, we use the dot profile for a previously designed level as the initial halftone to design the dot profile for a newer level. The sequence order of generate screen based on level-by-level shown in Fig. 15. Following this order, we can get more symmetry screen. Comparing with the standard low to high level single-seed screen, the two PARAWACS halftone results based on the two screens are shown in Fig. 17, we can see that the halftone based on standard order screen has more ”chain” texture, but the zigzag order double-seed screen can generate a more cluster texture near the middle gray levels.

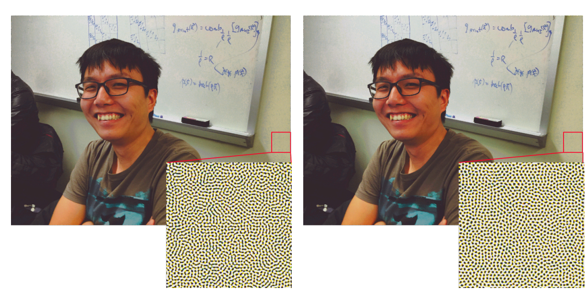
Comparison between Halftoning Algorithms

Now, let's do the comparison between the conventional periodic clustered-dot halftoning method, PARAWACS-MS-MP-CLU-DBS algorithm based on double-seed halftone and the NPAC-MS-MP-CLU-DBS algorithm. Fig. 18 shown the first comparison. Based on the zoom in image, it's very easy to recognize the periodic property of halftone based on conventional halftoning method. Both of the zoom in parts of halftone based on PARAWACS-MS-MP-CLU-DBS and NPAC-MS-MP-CLU-DBS are stochastic clustered-dot halftone. Actually, both of



(a) double seed halftone (b) lower seed halftone (c) upper seed halftone

Figure 16. *Double seed halftone pattern for PARAWACS-MS-MP-CLU-DBS. The red dots in (a) are the dots in lower seed halftone, and the green dots in (b) are the dots in higher seed halftone; (b) shown the lower seed halftone which is the dot center of final screen and (c) shown the higher seed halftone which is the hole center of final screen.*



(a) PARAWACS single-seed halftone (b) PARAWACS double-seed halftone

Figure 17. *Comparison between single-seed and double-seed halftone patterns for PARAWACS-MS-MP-CLU-DBS. (a) is the halftone image based on single-seed matrix for PARAWACS; (b) is the halftone image based on double-seed matrix for PARAWACS.*

the halftone looks nice, and it's difficult to say which algorithm is better only based on the halftone results of this image. The second comparison based on the test pattern, the "bulls-eye" pattern, also be done. The comparison is shown in Fig. 19. There are four patterns shown in this figure, from left to right, the fist pattern is the original continuous-tone image, the second one is the halftone based on conventional clustered-dot halftoning method, the third one is the halftone based on PARARACS-MS-MP-CLU-DBS, the fourth one is the halftone based on NPAC-MS-MP-CLU-DBS. From this figure, we can identify lots of moire in conventional halftone result; there is no moire in PARAWACS-MS-MP-CLU-DBS halftone result which improve a lot than the conventional method, but it's a little noisy; the NPAC-MS-MP-CLU-DBS halftone result is the best one, which is smooth and less noisy though there still be some tiny moire.

Conclusions

The pipeline for halftoning method based on NPAC, which can express the color of halftone in NP area coverages instead of pixel level, provide the input NPAC images for halftoning algorithm. Besides, the two new color aperiodic clustered-dot halftoning algorithms really reduce the visible moire and rosette and improve the quality of halftone image comparing with the conventional periodic clustered-dot halftoning method.

References

[1] Jan P. Allebach, "Selected Papers on Digital Halftoning", *SPIE Milestone Series*, SPIE Bellingham, WA, 1999.

[2] T.N. Pappas, "Model-based halftoning of color images", *IEEE Transactions on, Image Processing*, 6(7), 1014-1024 (1997)

[3] T. J. Flohr, B. W. Kolpatzik, R. Balasubramanian, D. A. Carrara, C. A. Bouman and J. p. Allebach, "Model based color image quantization, Human Vision, Visual Processing, and Digital Display IV", SPIE 1913, 270-281 (1993)

[4] 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)

[5] Madhur Gupta, Puneet Goyal, ManiFischer, Carl Staelin, Tamar Kashti, Omri Shacham and Jan Allebach, "Clustered-Dot Halftoning with Direct Binary Search", *IEEE Transactions on Image Processing*, 473-487 (2011)

[6] Puneet Goyal, Madhur Gupta, Carl Staelin, Mani Fischer, Omri Shacham, Jan Allebach, "Clustered-Dot Halftoning with Direct Binary Search", *IEEE Transactions on Image Processing* (2013)

[7] B. Kolpatzik and C. Bouman, "Qptimized Error Diffusion for Image Display", *Journal of Electronic Imaging*, vol. 1, no. 3, pp. 277-292. July 1992

[8] J*á*n Morovič, Peter Morovič and Jordi Arnabat, "HANS: Controlling Ink-Jet Print Attributes Via Neugebauer Primary Area Coverages", *IEEE Transactions on Image Processing* (2012)

[9] Je-Ho Lee and Jan Allebach, "Colorant-based direct binary search halftoning", *Electronic Imaging* 11(4), 517-527 (2002)

[10] Madhur Gupta, Puneet Goyal, Carl Staelin, Mani Fischer, Omri Shacham and Jan Allebach, "Improved Clustered-Dot Direct Binary Search Halftoning", *IEEE Transactions on Image Processing*, 473-487 (2011)

[11] Peter Morovič, Jan Morovič, Jay Gondek, Matthew Gaubatz and Robert Ulichney, "PARAWACS: color halftoning with a single selector matrix", *IS&T's 24th Color and Imaging Conference* (2016)

[12] Farhan A. Baqai and Jan P. Allebach, "Computer-Aided Design of Clustered-Dot Color Screens Based on a Human Visual System Model", *Proceedings of the IEEE* (2002)

[13] Bangyong Sun and Shisheng Zhou, "Study on the 3D Interpolation Models Used in Color Conversion", *IACSIT International Journal of Engineering and Technology* (2012)

[14] J*á*n Morovič, "COLOR GAMUT MAPPING", emphWiley-IS&T Series in Imaging Science and Technology (2008)

[15] F. E. Bockman and G. Li, "Device state error diffusion technique for halftoning", emphU.S. Patent 6081344 (2000)

[16] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Digital halftoning by means of green-noise masks", *J. Optical Society of America A*, vol. 16, no. 7, pp. 1575-1586, 1999.

[17] N. Damera-Venkata and Q. Lin, "AM-FM screen design using donut filters", in *Proc. SPIE, The International Society for Optical Engineering*, vol. 5293, pp. 469-480, Jan. 2004.

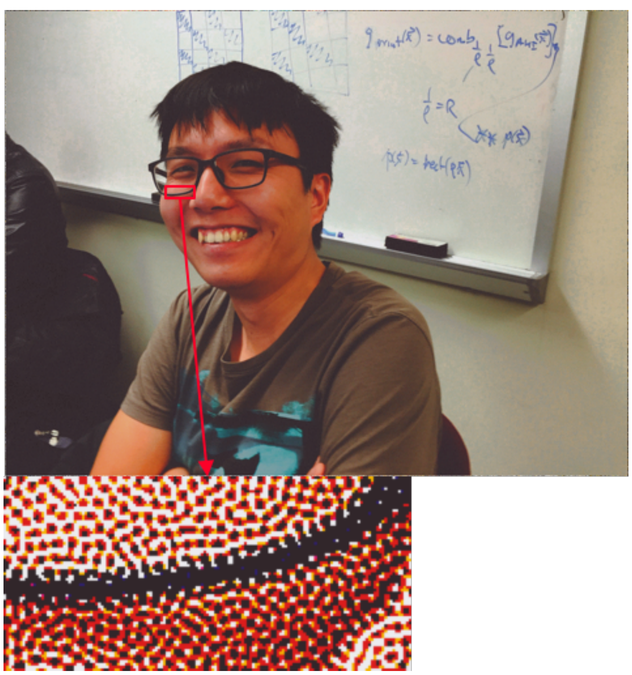
[18] Bong-Sun Lee, "Characterization of Opponent Channels, Automated Detection of PQ Defects, and Green-Noise Mask Design by Dual-Metric",*PhD dissertation*, 2006

Author Biography

Weijuan Xi received her BS in Electrical Engineering from Northwestern Polytechnical University, China (2013). Now, she is a direct PhD student working on image processing at Electrical and Computer Engineering, Purdue University. Her work is focus on image quality and digital halftoning.

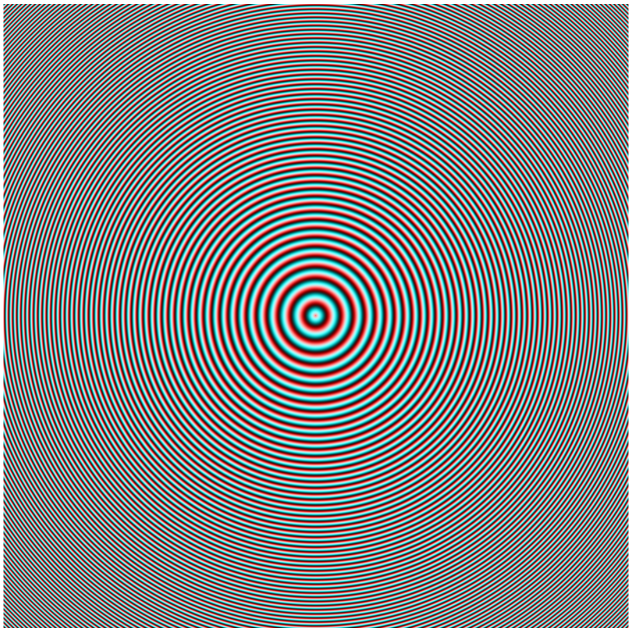
(a) conventional clustered-dot halftone image (130 lpi)

(b) PARAWACS-CLU-DBS halftone image (130 lpi)

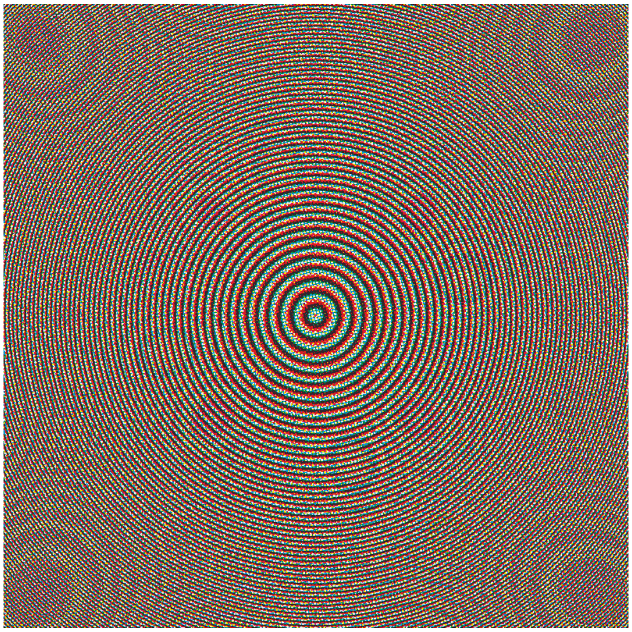


(c) NPAC-CLU-DBS halftone image (130 lpi)

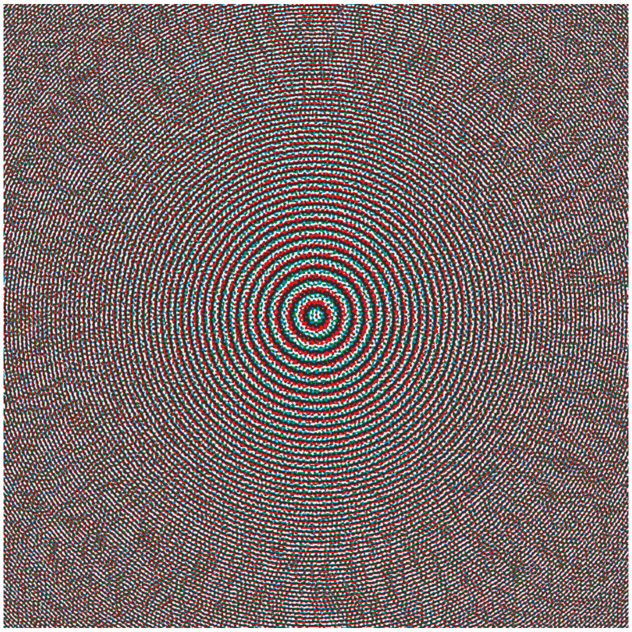
Figure 18. *Comparison between original continuous-tone image, halftone images of conventional CLU-DBS algorithm, PARAWACS-MS-MP-CLU-DBS and NPAC-MS-MP-CLU-DBS algorithm. (a) is the halftone image based on conventional clustered-dot halftoning method; (b) is the halftone image based on PARAWACS-MS-MP-CLU-DBS; (c) is the halftone image based on NPAC-MS-MP-CLU-DBS.*



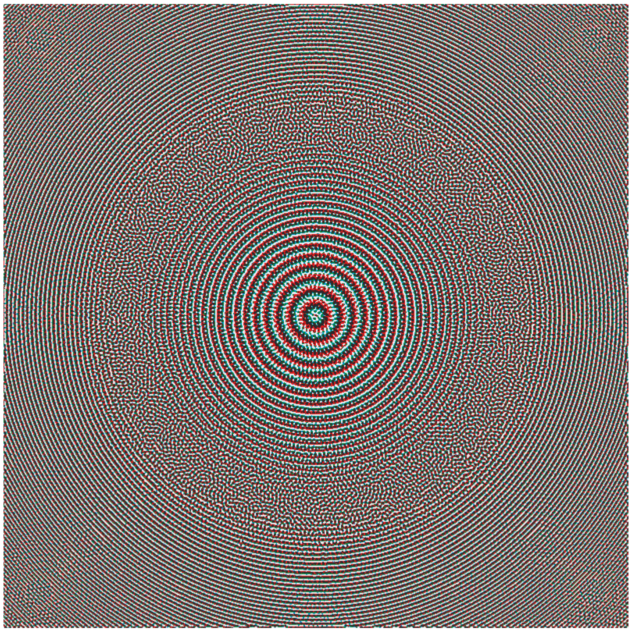
(a) original continuous-tone image



(b) conventional clustered-dot halftone image



(c) PARAWACS-CLU-DBS halftone image



(d) NPAC-CLU-DBS halftone image

Figure 19. *Comparison between original continuous-tone image, halftone images of conventional CLU-DBS algorithm, PARAWACS-MS-MP-CLU-DBS and NPAC-MS-MP-CLU-DBS algorithm. The cluster frequence of all the three halftone images is about 130lpi*