COLOR HALFTONING BASED ON NEUGEBAUER PRIMARY AREA COVERAGE AND NOVEL COLOR HALFTONING ALGORITHM FOR INK SAVINGS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Wanling Jiang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Allebach, Chair

School of Electrical and Computer Engineering

Dr. GEORGE T. CHIU

School of Mechanical Engineering

Dr. MARY L. COMER

School of Electrical and Computer Engineering

Dr. MICHAEL D. ZOLTOWSKI

School of Electrical and Computer Engineering

Approved by:

Dr. Pedro Irazoqui

Head of the School Graduate Program

To my family and friends.

ACKNOWLEDGMENTS

First of all, I would like to thank my major advisor Prof. Jan P. Allebach, who has given me the opportunity to work with him, provided me with various research opportunities, and to learn how to do great research under his guidance, and most of all, supported me and believed in me during the most difficult times when there's no one that could have given me such a tremendous support for my study at Purdue. Before joining EISL group, I have already heard that Prof. Allebach is a great advisor from my great friend Jing Li. After joining the group, I found the reason why. Prof. Allebach is very patient with his students, I was deeply influenced by his teaching style and the instructions in guiding me through difficult problems and making open-ended research problems approachable to me. I enjoyed working with him during my five years of PhD life not only because he is an excellent mentor in my research, but also because he is a great friend who always supported and believed in me.

Secondly, I'd like to thank Dr. Robert Ulichney who provided me with valuable inputs and guidance on my research.

I would also like to sincerely thank all other committee members: Prof. Mary Comer, Prof. Michael Zoltowski, and Prof. George Chiu. They ware all very encouraging and nice to me and gave me helpful explanations for my questions.

Also, I would like to thank my research sponsors, the Hewlett-Packard Barcelona division and Scitex division, who provided support and good feedback for the research. And the fellow research students who worked with me throughout the years.

Lastly, I would like to thank my family, especially my father Yusheng Jiang and my mother Xiuping Gu, for their continuous support in my studies. Their support is my fuel in perfecting myself and becoming a better researcher.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

HVS	Human Visual system
DBS	Direct binary search
CDBS	Color direct binary search
BSB	Basic screen block
NP-DBS	Neugebauer Primary
NPAC	Neugebauer Primary area coverage
PARAWACS	Parallel Random Weighted Area Coverage Selection
IS-SF-SA-MD	Ink-Saving, Single-Frequency, Single-Angle, Multi-Drop
CMYKW	Cyan, Magenta, Yellow, Black and White
LUT	Look-up table
GCR	Gray component removal
CPHC	Continuous-parameter halftone cell
DPHC	Discrete-parameter halftone cell
RAPS	Radially averaged power spectrum

ABSTRACT

Jiang, Wanling Ph.D., Purdue University, May 2019. Color Halftoning based on Neugebauer Primary Area Coverage and Novel Color Halftoning Algorithm for Ink Savings. Major Professor: Jan P. Allebach.

This dissertation primarily studies the problem related to image quality and ink savings in printing devices. It consists of three parts: color halftoning based on Neugebauer Primary area coverage, PARAWACS screen design and optimization and a novel color halftoning algorithm for ink savings.

We will first illustrate a halftoning method with Neugebauer Primary Area Coverage (NPAC) direct binary search (DBS). To represent a continuous-tone color image in limited amount of ink, we need the assistance of the human visual system (HVS) model, by comparing the filtered original and halftoned images, we are able to obtain a homogeneous and smooth halftone colored image. The halftoning is based on separating the colored image represented in Neugebauer Primary space in three channels based on human visual system. The optimization is obtained by swapping the halftoned image pixels and bringing the error metric to its minimum (swap only DBS). The separation of chrominance HVS filters between red-green and blue-yellow channels allows us to represent the HVS more accurately. Color halftone images generated using this method and the method of using traditional screening methods are compared, and our method significantly reduces moire effect.

Then, we discuss how to achieve faster halftoning speed using PARAWACS screen. PARAWACS screen in this document refer to a screen designed using DBS to generate halftone images mimicking the effect of DBS monochrome as well as colored halftone, this screen is created level by level.With PARAWACS screen, we can create halftone using simple pixel by pixel comparison to create halftone with the merit of DBS,

an iterative method to create homogeneous, visually pleasing halftone. Although PARAWACS screen is a monochrome screen, we can use it to create color halftone with good results.

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

1. INTRODUCTION

This dissertation studies image quality and ink saving problems for inkjet and laserjet printers.

First of all, we study the iteratively based halftoning method that can achieve optimum image quality: color halftoning problem based on Neugebauer Primaries. The challenge is to create a digital representation close to the original image with a small subset of colorants that expand the color space of the user's choice. We selected several Neugebauer Primaries to test our method and it yeilded an excellent result. In our experiments, the chosen colorants are white (W), the primaries (C, M, Y), secondaries (CM, MY, CM) and one tertiary (CMY or black). To represent a continuous-tone image with these colorants and create a smooth image, we developed a method to generate the initial halftone image that describes the original image in the chosen colorant set. This is done by color management based on tetrahedral interpolation. Then, we designed a DBS based method (NPAC-DBS) to optimize the initial halftone image to make it smooth and homogeneous. We also studied multiple human visual systems and selected the ones to be utilized in our method for luminance channel, such as Nasanen's filter, Wandell's filter and Daly's filter. We also compared using single and separate filters for blue-yellow and red-green channels. We choose the configuration that yields the best result for our purpose. The color space used in the method is $Y_yC_xC_z$ because it perserves local average. One major benefit of our NP-DBS is that this method does not have an restriction on the number of colorants to be used on the halftone. However, this method has a limitation on the speed of computation due to its iterative nature.

Then, we focus on improving the speed of color halftoning with minimum compromise of the halftone quality. We already know that iteratively optimizing the halftone using DBS give us a very visually pleasing halftone image, and we also know that a

screening based method is very fast in halftoning. So we come up with the design of PARAWACS screen which enables faster color halftoning compared with previous method. With this screening method, only one pass is needed to produce the halfoned image and the quality is similar to that produced by DBS based result. To generate PARAWACS screen with the merit of DBS generated halftone results, we design the screen level by level with stacking constraint. For each level, we use DBS to choose the dot location. We carefully select the design parameters, such as swap neighborhood, block size, swap only vs. swap toggle, etc., to improve the textures, especially in highlights and shadows, since the textures in these regions are more visible. We observed a defect of low frequency energy in the generated screen. But with improved design parameters, we are able to remove this defect and obtain a desirable halftoned image using the resulting screen.

In the novel halftoning method for ink savings, the challenge is to create a periodic clustered-dot color halftone pattern with a limited amount of ink that reduces the overlap between black and ink of other colorants. So the question is how to overlap different colorants that satisfy the above requirements. In the design, we developed detailed overlapping rules for different colorant amounts with ink limitation to create the desired halftone results. We place the C, M and Y colorants on half a grid shift in both directions, thus reducing the overlapping between CMY and K. We sequentially perform the halftoning, starting with a special growing sequence of black, followed by cyan, magenta and yellow. The placement of cyan dots depends on the placement of black dots, and the magenta dots depends on black and cyan dot placement. Similarly, yellow dots depends on all previous colorant placement. We observed the color space for the Scitex inkJet printers, and found that the colors are very close to each other in L*a*b* color space when there are three drops of K printed together with various other colorants. So we specifically do not allow any C, M, or Y colorants to print on 3 drops of K. We choose the L*a*b* color space because it is perceptually uniform color space. Similarly, we developed other rules for more desirable results. We also exploit multilevel halftoning in the novel halftoning method to create more gray levels.

In order to create a more compact growing sequence, we designed the dot growing sequence or microcell for C, M, Y plane and K plane, with each microcell the same angle. The design of the microcell needs to maintin existing dots or hole cluster, and new dot pixel are added to the boundary of existing dot or hole cluster. We also need to maintain symmetry of dot and hole clusters as much as possible, and consider the periodicity of the halftone screen. The ink saving is significant with satisfying halftoning texture as a result.

To sum up, this dissertation is intended to study image quality and ink savings for inkjet and laserjet printers. The dissertation is structured as follows:

• Chapter 2 describes how we designed the color halftoning based on Neugebauer Primary area coverage that can achieve desirable halftone image quality and how this method can potentially be used on printers with an arbitrary number of Neugebauer Primaries or ink combinations.

• Chapter 3 illustrate the PARAWACS screen design with faster computation speed with little compromise on the halftone quality.

• Chapter 4 studies the novel halftoning for ink savings problem.

• Chapter 5 summarises the contributions of this dissertation.

2. COLOR HALFTONING BASED ON NEUGEBAUER PRIMARY AREA COVERAGE

2.1 Introduction

Halftoning is used to represent different tones by placing black dots on a white surface. Due to the low pass nature of the human eye, the dots become invisible when viewed from a distance. Similarly, digital color halftoning converts continuous tone images to images comprising pixels with limited amount of colorants, ideally with minimal compromise of the image quality. Unlike traditional halftoning, digital color halftoning does not only represent the original image as traditional black and white halftoning does, but the color representation needs to be as close to the original image as possible.

There are several major color halftoning methods [1], the method that require the least computation is memoryless operation, processing each pixel at a time and do not revisit pixel, an example of this method is using selection matrix, in this method, each pixel is represented by Neugebauer Primary area coverage (NPAC), which will be introduced in section IV. Using the value given by the selection matrix, we can select the Neugebauer Primary (NP) for the pixel. A halftoned image represented by NPs is generated if we repeat this process for all the pixels. Since each NP corresponds to a unique sRGB value, we can represent the halftoned image in sRGB colorspace. This method is similar to screening method [2–5], which is also based on pixel by pixel comparison. Another method that requires only the minimal computation among all the methods of halftoning is look-up-table based algorithm [6].

Another halftoning method require the information of neighborhood pixels and thus has more computation but has better halftoning quality. A example of this type of halftoning is error diffusion [7]. In this method, each pixel diffuses the error to

its nearby pixels, or each pixel collects the error information from its neighboring pixels. The difference from previous methods is that it each pixel has information of its neighborhood, thus yielding halftone images with better quality. This method only has a single pass through the entire image, no iteration is needed.

The most computationally expensive ones are iterative methods. The image is visited multiple times. An example of this method is DBS [8]. There are two categories of DBS in terms of the graininess of the dots, dispersed-dot and clustered-dot DBS. Both methods render good halftone image, the difference between them is that clustered-dot DBS generates coarser grained halftone images, which is suitable for printers that can not print stable isolated pixels, while dispersed-dot DBS renders finer gained halftone images, and is suitable for printers that can print stable isolated pixels. The method used in this paper is dispersed-dot DBS. The DBS algorithm takes into account the human visual system (HVS), which is a low pass filter. The DBS algorithm optimizes the halftone by bringing the gap between the filtered error between continuous and halftoned images down by performing swap and toggle of the pixels in the halftone image. If both swap and toggle are permitted, the tone of the initial halftone image is not preserved, however. If only swap is permitted, the tone of the initial halftone image is preserved. In this paper, swap-only DBS is adopted. In DBS, we have observed that the initial halftone has impact on the final halftone image quality. With a DBS generated initial halftone, the final halftoned image has better quality than using a random initial halftone.

There are four major categories of halftones in terms of dot cluster size and periodicity. They are clustered dot periodic halftone, clustered dot aperiodic halftone, dispersed dot periodic halftone and dispersed dot aperiodic halftone.

Our work, NPAC-DBS halftoning, is based on the work of Je-Ho Lee [9] and A. Ufuk Agar [10], where color DBS is based on Neugebauer Primaries, and the algorithm operates in $Y_yC_xC_z$ color space. It belongs to the category of aperiodic dispersed dot halftone. We first take an image in sRGB color space, then transfer it to an NPAC

image set, which represent the continuous tone image in Neugebauer Primary color space, and finally use swap only NPAC DBS to obtain the halftoned image.

2.2 Human Visual System Model

We adopt the HVS model in $Y_yC_xC_z$ color space is based on CIE L*a*b* uniform color space which is used by Flohr et al. [11], this color space is defined using CIE XYZ as:

$$\begin{aligned} L^* &= 116 f(\frac{Y}{Y_n}) - 16 \\ a^* &= 200[f(\frac{X}{X_n}) - (\frac{Y}{Y_n})] \\ b^* &= 500[f(\frac{Y}{Y_n}) - f(\frac{Z}{Z_n})] \end{aligned} \tag{2.1}$$

Where X_n, Y_n, Z_n represent CIE XYZ tristimulus values , and $f(x)$ is defined by:

$$f(x) = \begin{cases} 7.787x + \frac{16}{116}, & 0 \leq x \leq 0.008856 \\ x^{1/3}, & 0.008856 < x \leq 1. \end{cases} \tag{2.2}$$

In the above equation, L^* corresponds to luminance and a^* and b^* correspond to the red-green and blue-yellow opponent channels respectively. The problem with CIE L*a*b* is it does not preserve the spatial averaged tone, so a linearized CIE L*a*b* color space is used by Flohr et all. [11] and they denote it by $Y_yC_xC_z$:

$$\begin{aligned} Y_y &= 116\frac{Y}{Y_n} \\ C_x &= 200[\frac{X}{X_n} - \frac{Y}{Y_n}] \\ C_z &= 500[\frac{Y}{Y_n} - \frac{Z}{Z_n}] \end{aligned} \tag{2.3}$$

Where Y_y represent luminance, and C_x and C_z represent red-green and blue-yellow channels respectively.

2.2.1 Nasanan's HVS model

For the luminance channel, we can use Nasanan's model [12]. The spatial frequency domain response is given by:

$$H_{Y_y}(u,v) = a\Gamma^b exp(-\frac{\sqrt{u^2+v^2}}{\lfloor c\ln\Gamma + d\rfloor}) \tag{2.4}$$

Were $a = 131.6$, $b = 0.3188$, $c = 0.525$, $d = 3.91$, and Γ is the average luminance of the light reflected from the print in cd/m^2, usually set to 11, and u and v are the spatial frequency coordinates in cycles/degree subtended at the retina.

2.2.2 Chromiance channel models based on data collected by Mullen

For the chrominance channels, the blue-yellow channel spatial frequency domain response is based on approximation by Kolpatzik and Bouman [13] to experimental data collected by Mullen [14]:

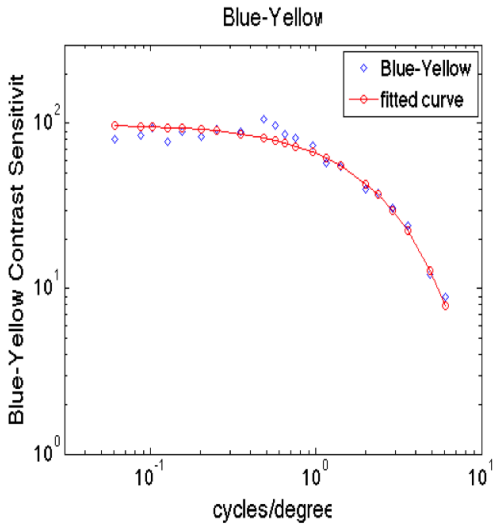


Fig. 2.1.: The blue-yellow channel CSF

The diamond marks in Fig. 2.1 represents the sampled data from Mullen's blue-yellow CSF figure, and the red curve is the fitted curve according to the samples. The contrast sensitivity is given as:

$$H_{C_z}(u,v) = Aexp(-\alpha\sqrt{u^2+v^2}) \tag{2.5}$$

Where $\alpha = 0.419$ and $A = 100$, and u and v are the spatial frequency coordinates in cycles/degree subtended at the retina.

Different from using blue-yellow channel filter for both red-green and blue-yellow channels, we use separate red-green channel filter based on the experimental data collected by Mullen [14]:

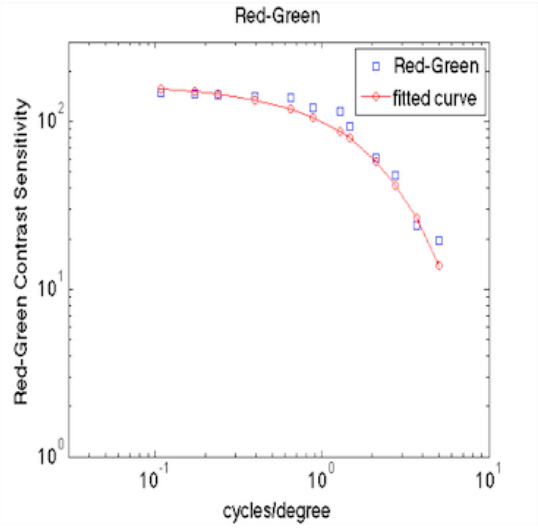


Fig. 2.2.: The red-green channel CSF

The square marks in Fig. 2.2 represents the sampled data from Mullen's red-green CSF figure, and the red curve is the fitted curve according to the samples. This spatial frequency domain model is approximated by:

$$H_{C_x}(u,v) = Aexp(-\alpha\sqrt{u^2+v^2}) \tag{2.6}$$

Where $\alpha = 0.497$ and $A = 165$, and u and v are the spatial frequency coordinates in

cycles/degree subtended at the retina.

2.2.3 Wandell's HVS model

In addition to Nasanen's luminance channel filter and Mullen's chrominance channel filters, we also examined Wandell's filters [15]. Wandell has done a lot of experiments on color and how human eyes perceive color. [16] Wandell's $O_1O_2O_3$ color space in terms of CIE XYZ is [15]:

$$\begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix} = \begin{bmatrix} 0.279 & 0.72 & -0.107 \\ 0.449 & -0.290 & 0.077 \\ 0.086 & -0.590 & 0.501 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.7}$$

Wandell's spatial kernel is given by:

$$f = k \sum_i \omega_i E_i \tag{2.8}$$

Where $E_i = k_i exp[-\frac{x^2+y^2}{\sigma_i^2}]$. The scale factor k_i is chosen so that E_i sums to 1. the scale factor k is chosen so that for each color plane, its two-dimensional kernel f sums to one. The parameters (ω_i, σ_i) for the three color planes are:

Table 2.1.: Wandell filter parameters

Plane	**Weights** ω_i	**Spreads** σ_i
Luminance	0.921	0.105
	0.105	0.133
	-0.108	4.336
Red-green	0.531	0.0392
	0.330	0.494
Blue-yellow	0.488	0.0536
	0.371	0.386

Where spread is in degrees of visual angle [15].

Wandell's filter with respect to angular degree could not be applied to digital image, we need to generate discrete Wandell filter. Let x_d, y_d be the coordinates in angular degrees subtended at the retina, the point spread function in terms of the angular degrees is:

$$p_r(x_d, y_d) = exp(-\frac{x_d^2 + y_d^2}{\sigma^2}) \tag{2.9}$$

Assume that the viewing distance is D (inches), and x_p,y_p (in inches) are the coordinates on a paper at viewing distance D:

$$x_p = D \cdot x_d \cdot \frac{2\pi}{360} \tag{2.10}$$

The HVS with respect to x_p, y_p is:

$$p_p(x_p, y_p) = \exp\left(-\frac{x_p^2 + y_p^2}{\left(\frac{2\pi\sigma D}{360}\right)^2}\right) \tag{2.11}$$

Let Δ be the sampling interval in plane of paper, and $R = \frac{1}{\Delta}$, then

$$p_d[m,n] = \exp\left(-\frac{m^2+n^2}{\left(\frac{2\pi\sigma S}{360}\right)^2}\right) \tag{2.12}$$

Where S is the scaling factor, and $S = R \cdot D$, $m, n = 0, 1, ..., 13$.

2.2.4 Daly's luminance channel model

Sang Ho Kim and Allebach [17] compared Daly's contrast sensitivity function with Nasanen's model. Here we also examined Daly's luminance channel filter. Its spatial frequency domain model is given by:

$$H(\bar{\rho}) = \begin{cases} a(b + c\bar{\rho})\exp(-(c\bar{\rho})^d), & \rho > \bar{\rho} \\ 1, & else. \end{cases} \tag{2.13}$$

Where spatial frequency $\bar{\rho}$ has unit of cycles/degree, $a = 2.2$, $b = 0.192$, $c = 0.114$, $d = 1.1$, $\bar{\rho}_{max} = 6.6$. In order to get a 2-D frequency domain model, we replace $\bar{\rho}$ by $\sqrt{\bar{u}^2 + \bar{v}^2}$, where $\bar{u}, \bar{v}$ are in cycles/degree. We cannot directly use Daly's spatial frequency domain filter, but after 2-D inverse Fourier transform, we are able to apply it to our purpose as the spatial domain luminance filter $h[x_d, y_d]$, where x_d, y_d are in angular degrees subtended at the retina. Following a derivation similar to that for (9-10), we get the HVS point spread functions $\tilde{p}_i(x_p, y_p)$, $i = Y_y, C_x, C_z$, where x_p, y_p have units of *in* viewed at a distance D (inches).

$$\tilde{p}_i(x_p, y_p) = \frac{1}{D^2} h_i\left(\frac{180}{\pi D x_p}, \frac{180}{\pi D y_p}\right) \tag{2.14}$$

Note that for all the spatial domain HVS filters, we use a circular area of support.

2.2.5 Filter comparison

Here, we compare the luminance channel frequency domain filters, namely Nasanen's filter, Wandell's O_1 channel filter and Daly's HVS filter in 1D.

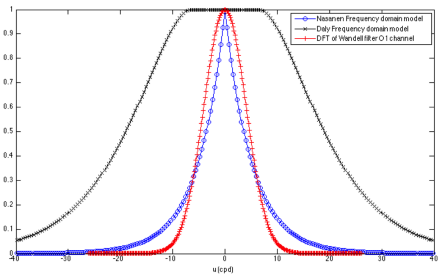


Fig. 2.3.: Comparison between Wandell, Daly and Nasanen frequency domain filters

Daly's frequency domain HVS model has a wide width compared to Nasanen's and Wandell's. If we take the inverse Fourier transform of Daly's luminance channel filter, the width of which is smaller compared with that of Nasanen's and Wandell's:

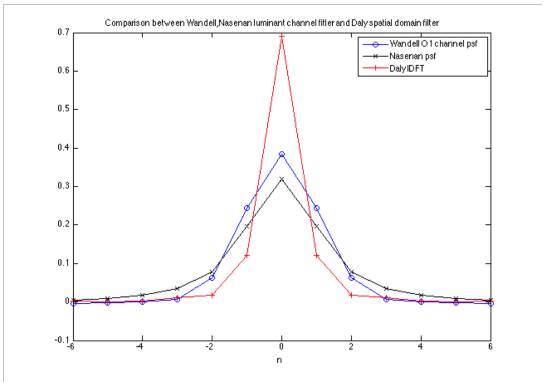


Fig. 2.4.: Comparison between Wandell, Daly and Nasanen spatial domain filters

First, we compare luminance channel filters by examine the filtered result of a constant tone halftone patch of size 512×512 that contains 50% cyan, 20% magenta, and 40% white, which is calculated in $Y_y C_x C_z$ color space. We set up the filter configurations as follows:

• Configuration 1: For luminance channel filter, we use Nasanen's model, for chrominance channel, we use the same filter for both red-green and blue-yellow channels as specified in (5).

• Configuration 2: Use Daly's model for luminance channel, chrominance channel

filter is the same as Configuration 1.

• Configuration 3: Wandell's HVS filter models in $O_1O_2O_3$ color space.

• Configuration 4: Use Nasanen's model for luminance channel, for chrominance channel, we use separate filters for red-green (6) and blue-yellow channel (5).

Here, a scale factor $S = 3000$ is used. scale factor is defined as $S \equiv RD$, it is the product of resolution and viewing distance, the product of these two factors affects the halftone result.

First we compare the images halftoned using three different luminance channel filters: (a) Nasanen's luminance filter (Configuration 1), (b) Wandell's filter (Configuration 3), (c) Daly's luminance filter (Configuration 2).

In order to compare the halftoning result using filters $(a) - (c)$, we need to first look at the error metric for color halftoning shown in the next section.

2.3 Color Direct Binary Search

2.3.1 color management

First, we need to convert an sRGB image to NPAC image set, which consists of the percentages of NPs. However, there is a mismatch between the gamut of sRGB (souce color space) and the gamut of NP of the target printer (destination color space). In order to represent an sRGB image in terms of percentages of NPs, we need to map the color space of sRGB to the color space of the NPs. The pipeline of the halftoning process is shown below:

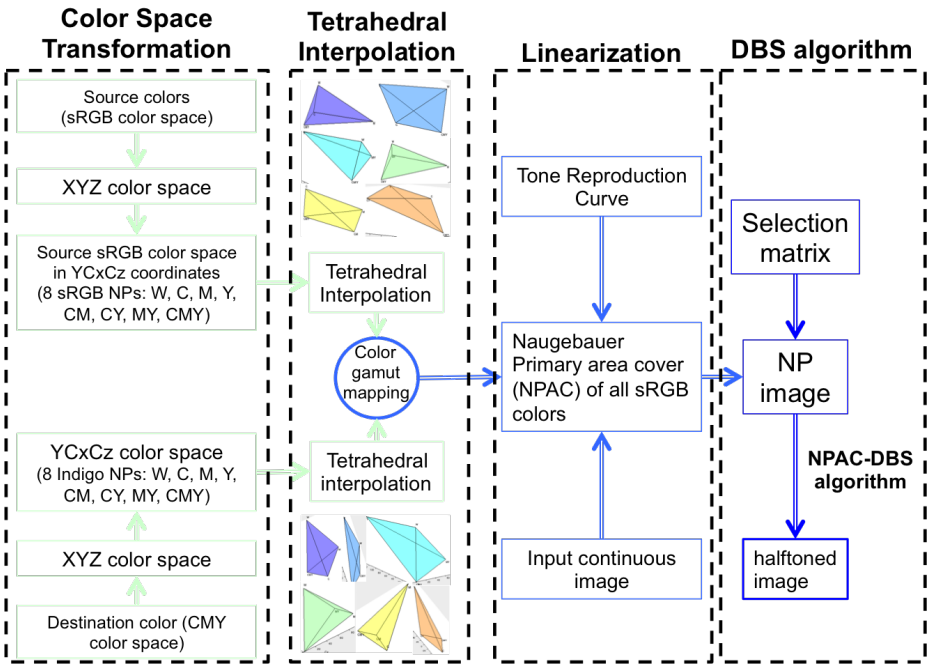


Fig. 2.5.: The pipline of getting halftoned image [18]

The figure above shows how the color management is performed. We first convert the the source (sRGB) to destination ($Y_yC_xC_z$) color space, because it is a linearized version of $CIEL*a*b*$ color space, and it preserves local average. The transformation is done by converting from sRGB to XYZ color space with D50 luminance, and then to $Y_yC_xC_z$ color space, the equations for the transformations are shown below:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.15}$$

Where

$$[M]^{-1} = \begin{bmatrix} 3.1338561 & -1.6168667 & -0.4906146 \\ -0.9787684 & 1.9161415 & 0.0334540 \\ 0.0719453 & -0.2289914 & 1.4052427 \end{bmatrix} \tag{2.16}$$

Then, we transform from XYZ to $Y_yC_xC_z$ color space:

$$\begin{bmatrix} Y_y \\ C_x \\ C_z \end{bmatrix} = \begin{bmatrix} \frac{1}{X_w} & 0 & 0 \\ 0 & \frac{1}{Y_w} & 0 \\ 0 & 0 & \frac{1}{Z_w} \end{bmatrix} \begin{bmatrix} 0 & 116 & 0 \\ 500 & -500 & 0 \\ 0 & 200 & -200 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.17}$$

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

Each sRGB value also corresponds to unique $Y_yC_xC_z$ values according to the conversion shown above. Then Tetrahedral interpolation [18] [19] is performed to get the NP separations, which is the representation of each sRGB value in terms of percentages of NPs in $Y_yC_xC_z$ color space. The color mapping method we adopt here is mapping the 8 NPs in the source and destination color space. By uniquely representing each sRGB colorant in the source color space, all corresponding colorants in the destination color space is also determined. The mapping from source to destination color space is shown below:

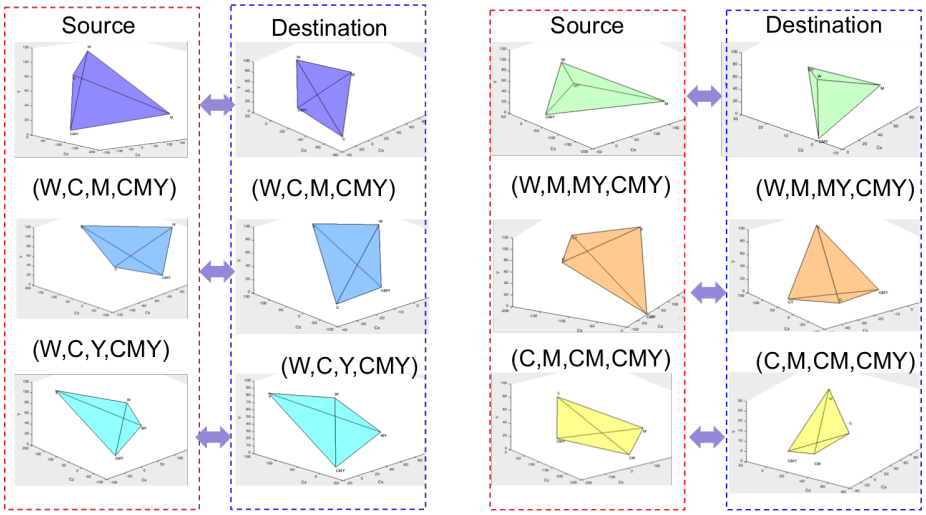


Fig. 2.6.: Mapping between source and destination color spaces [18]

This representation described above is called NPAC image set or NPAC separation. For each pixel in the image, the color of a halftone pattern represented by Neugebauer Primaries is:

$$T_c = \sum_{i=1}^{k^n} T_i a_i \tag{2.18}$$

Where $T_i \in Y_y C_x C_z$, is the color value in $Y_y C_x C_z$ color space for the i^{th} Neugebauer Primary, and $a_i \in [0, 1]$, is area coverage of the i^{th} Neugebauer Primary, $\sum a_i = 1$, c denotes the resulting halftone, n is the number of inks, and k is the number of levels per ink. For bilevel system with three types of inks, there are $2^3 = 8$ Neugebauer Primaries, which is used in this paper.

2.3.2 Initial NPAC halftone image

Given the continuous-tone NPAC image set, which has as many planes as NPs, we can get an initial NPAC halftone image using a random selection matrix, where each pixel is a random number between 0 and 1.

Let $Ac[m, n, k]$ be the area coverage of the NPs in an NPAC image set, where k denotes NP, and $k \in [0, NP_{max} - 1]$, where NP_{max} is the total number of NPs, and $[m, n]$ is the location of pixel on NPAC image, where $m \in [0, M-1]$ and $n \in [0, N-1]$, and M is the height, N is the width of the image. Let $r\ in[0, 1]$ be a random number, and let p be the NP being selected for pixel $[m, n]$, where $p \in [0, NP_{max} - 1]$, and p satisfy the following condition:

$$h[m, n] = \begin{cases} 0, & r \leq AC[m, n, 0] \\ p, & lb < r \leq ub, p \geq 1. \end{cases} \tag{2.19}$$

where $lb = \sum_{k=0}^{p-1} Ac[m, n, k]$, and $ub = \sum_{k=0}^{p} Ac[m, n, k]$.

Note that the NPs are arranged in an order such that 0 represent the lightest, and $NP_{max} - 1$ the darkest in visual weight. After repeating this process for all the pixels

in an NPAC image, we get the initial NP halftone image.

Using an existing selection matrix and NPAC image set to generate halftone is similar to how the initial Neugebauer Primary image is generated.

2.3.3 Color Direct Binary Search Error Metric

Similar to direct binary search (DBS) [20], NPAC-DBS also works by evaluating the error after each swap or toggle, but in order to preserve the average tone or color of the initial NP halftone image, we only allow swaps here. Our algorithm operates in opponent color space $Y_yC_xC_z$ as in A. Ufuk Agar's work [10]. We evaluate the effect of a swap on the error metric in the neighborhood of 25 nearest pixels, if the trial swap reduce the error metric, we accept it. This process is repeated for each pixel until there's no more changes accepted.

We use $[\mathbf{m}] = [m, n]^T$ to denote discrete coordinates and use $(\mathbf{x}) = (x, y)^T$ to denote continuous spatial coordinates. And we use a 3-D vector valued function to represent a continuous tone image: $\mathbf{f}[\mathbf{m}] : \mathbb{R}^2 \rightarrow \mathbb{R}^3$.

Let $f_{Y_yC_xC_z}[\mathbf{m}]$ and $g_{Y_yC_xC_z}[\mathbf{m}]$ be the continuous-tone color image and halftone color image in $Y_yC_xC_z$ color space,respectively. We represent each plane by $f_i[\mathbf{m}]$ and $g_i[\mathbf{m}]$, where $i = Y_y$,C_x,C_z. The error image between the original continuous-tone image and the halftone image is:

$$\begin{aligned} e_{Y_yC_xC_z}[\mathbf{m}] &\equiv f_{Y_yC_xC_z}[\mathbf{m}] - g_{Y_yC_xC_z}[\mathbf{m}] \\ e_i[\mathbf{m}] &\equiv f_i[\mathbf{m}] - g_i[\mathbf{m}], i = Y_y, C_x, C_z \end{aligned} \tag{2.20}$$

Assuming neighboring printer dot interaction is additive, with HVS filters, the perceived error in $Y_yC_xC_z$ color space $\tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x})$ with respect to continuous spatial coordinate $\mathbf{x}$ is:

$$\begin{aligned} \tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x}) = \sum_m diag\,(&\tilde{p}_{spot\_Yy}(\mathbf{x} - \mathbf{Xm}), \\ &\tilde{p}_{spot\_Cx}(\mathbf{x} - \mathbf{Xm}), \tilde{p}_{spot\_Cz}(\mathbf{x} - \mathbf{Xm}) e_{Y_yC_xC_z}[\mathbf{m}] \end{aligned} \tag{2.21}$$

Where $\tilde{p}_{spot\_i}(\mathbf{x}) \equiv \tilde{p}_i(\mathbf{x}) * p_{spot}(\mathbf{x})$ is the i^{th} component of the HVS filter point spread function convolved with the spot function of the printer, $\mathbf{X}$ is the periodicity matrix whose columns comprise the basis for the lattice of printer addressable dots. $diag(\cdot)$ is a diagonal matrix with elements listed between the parentheses. But because the spot function of the printer has much smaller support than HVS point spread function, we assume $\tilde{p}_{spot\_i}(\mathbf{x}) \approx \tilde{p}_i(\mathbf{x})$. Therefore, eq. (2.22) can be approximated by:

$$\tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x}) = \sum_m \tilde{\mathbf{P}}(\mathbf{x} - \mathbf{Xm}) e_{Y_yC_xC_z}[\mathbf{m}] \tag{2.22}$$

Where $\tilde{\mathbf{P}}(\mathbf{x}) \equiv diag(\tilde{p}_{Yy}(\mathbf{x}), \tilde{p}_{Cx}(\mathbf{x}), \tilde{p}_{Cz}(\mathbf{x}))$. The error metric E is defined to be the total squared perceived error of all three components of the $Y_yC_xC_z$ color space.

$$E = \int \tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x})^T \tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x}) d\mathbf{x} \tag{2.23}$$

Substitute eq.(2.23) into eq.(2.24):

$$\begin{aligned} E = \sum_m \sum_n \tilde{\mathbf{e}}_{Y_yC_xC_z}[\mathbf{m}]^T \left(\int \tilde{\mathbf{P}}(\mathbf{x} - \mathbf{Xm}) \tilde{\mathbf{P}}(\mathbf{x} - \mathbf{Xn}) d\mathbf{x} \right) \\ \times \tilde{\mathbf{e}}_{Y_yC_xC_z}[\mathbf{n}]. \end{aligned} \tag{2.24}$$

The auto-correlation function of $\tilde{p}_i(\mathbf{x})$ is denoted $\mathbf{c}_{\tilde{p}\tilde{p}}(\mathbf{x})$, and the cross correlation function between $\tilde{p}_i(\mathbf{x})$ and $\tilde{e}_i(\mathbf{x})$ is denoted as $\mathbf{c}_{\tilde{p}\tilde{e}}(\mathbf{x})$. Then we can rewrite eq.(2.25) as

$$\begin{aligned} E &= \sum_m \sum_n \tilde{\mathbf{e}}_{Y_yC_xC_z}[\mathbf{m}]^T \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{n}] \tilde{\mathbf{e}}_{Y_yC_xC_z}[\mathbf{n}]. \\ &= \sum_m \tilde{\mathbf{e}}_{Y_yC_xC_z}[\mathbf{m}]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \end{aligned} \tag{2.25}$$

Our goal is the minimize the perceived error E above, A. Ufuk Agar and Allebach [10] performed DBS algorithm in the $Y_yC_xC_z$ color space. They employed separate HVS models for luminance and chrominance channels, but they only used Nasanen's HVS model and a single HVS filter for both red-green and blue-yellow channel. Here, we are going to compare several different models for each channel.

Next, we will talk about efficient evaluation of trial swaps on the error metric E eq.(2.26).

2.4 Efficient Evaluation of Trial Swap

Assuming that we have three-colorants cyan, magenta and yellow, there are 8 possible Neugebauer primaries for a bilevel printer. We order all the combinations from lightest to darkest in visual weight: W, Y, C, CY, M, MY, CM, CMY, where W stands for white when there are no dots printed on the paper. We only perform swaps because it preserves the average local color of the image. Let $\mathbf{m}_1$ and $\mathbf{m}_2$ be the trial swap halftone pixel locations, the changes at the swap locations in $Y_yC_xC_z$ color space at locations $\mathbf{m}_1$ and $\mathbf{m}_2$ are denoted by $\mathbf{a}[\mathbf{m}_1]$ $\mathbf{a}[\mathbf{m}_2]$. We define $\mathbf{A}[\mathbf{m}_1] \equiv diag(\mathbf{a}[\mathbf{m}_1])$ and $\mathbf{A}[\mathbf{m}_2] \equiv diag(\mathbf{a}[\mathbf{m}_2])$. Then the change in the error metric is:

$$\begin{aligned}\Delta E_s =&(\mathbf{a}[\mathbf{m}_1]^T\mathbf{A}[\mathbf{m}_1] + \mathbf{a}[\mathbf{m}_2]^T\mathbf{A}[\mathbf{m}_2])\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{0}] \\ &+ 2\mathbf{a}[\mathbf{m}_1]^T\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_1] + 2\mathbf{a}[\mathbf{m}_2]^T\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_2] \\ &+ \mathbf{a}[\mathbf{m}_2]^T\mathbf{A}[\mathbf{m}_1]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}_2 - \mathbf{m}_1]\end{aligned} \tag{2.26}$$

Here we keep $\mathbf{c}_{\tilde{p}\tilde{p}}$ and $\mathbf{c}_{\tilde{p}\tilde{e}}$ as lookup tables, and if the trial swap is accepted, we update the $\mathbf{g}'_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}]$ as:

$$\begin{aligned}\mathbf{g}'_{Y_yC_xC_z}[\mathbf{m}] =&\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_1]\delta[\mathbf{m} - \mathbf{m}_1] \\ &+ \mathbf{a}[\mathbf{m}_2]\delta[\mathbf{m} - \mathbf{m}_2]\end{aligned} \tag{2.27}$$

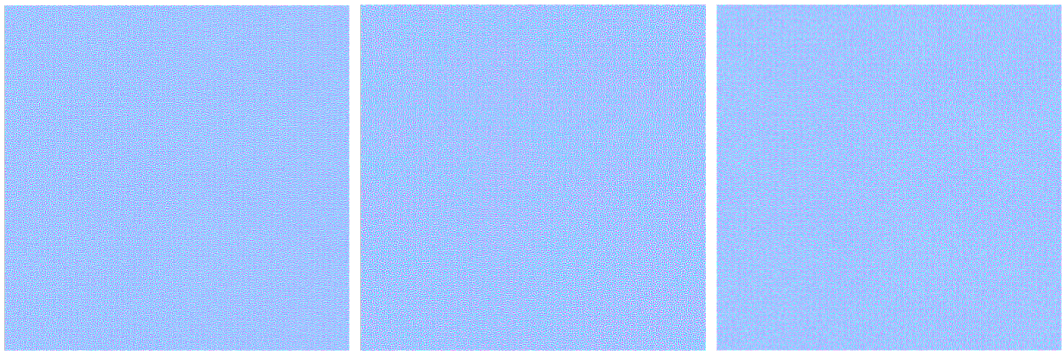
$$\begin{aligned}\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] =&\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_1]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_1] \\ &+ \mathbf{A}[\mathbf{m}_2]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_2]\end{aligned} \tag{2.28}$$

2.5 Experimental results

2.5.1 Comparison of halftoned images using different filters

Let the normalized error be $E_{norm} = \sqrt{\frac{E}{height \times width}}$, where *height* and *width* are the height and width of the halftone image, and E is the error metric. The halftoned images using the three different filters is shown below:

E_{norm} of halftone using (a) filter is 1.4263. For (b), it has $E_{norm} = 5.2454$ and (c) has $E_{norm} = 1.9160$. We cannot use E_{norm} to determine halftone quality, since



(a) Nasanen's HVS filter (b) Wandell's HVS filter (c) Daly's HVS filter

Fig. 2.7.: Comparison of halftoned images using different filters.

different HVS filters are applied in the halftoning process. The only reliable way to compare the halftoned images is by looking at them and make the final judgement. Based on this, using Nasanen's luminance channel filter in $Y_yC_xC_z$ color space is better. We can see that E_{norm} in this case is also the smallest.

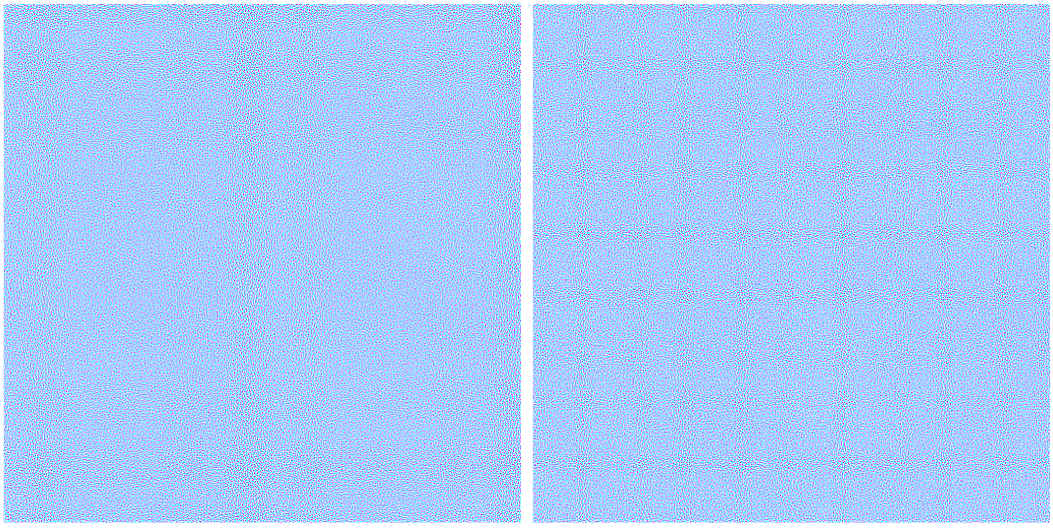
Next, we compare halftone results in $Y_yC_xC_z$ and $CIEL^*a^*b^*$ color spaces because these two color spaces are similar. The halftoned images in $Y_yC_xC_z$ and $CIEL^*a^*b^*$ color spaces are shown below:

The halftone image in Fig. 2.6 (a) and (b) has $E_{norm} = 1.4263$ and $E_{norm} = 1.4433$ respectively. They are very similar.

We also changed the gain factor of luminance channel filter on the overall error metric, we increased the gain factor g to four and compared with the original factor $g = 1$. Here, we still use Nasanen's lumiance channel filter (Configuration 1). The halftoned images with different gain factors are shown below:

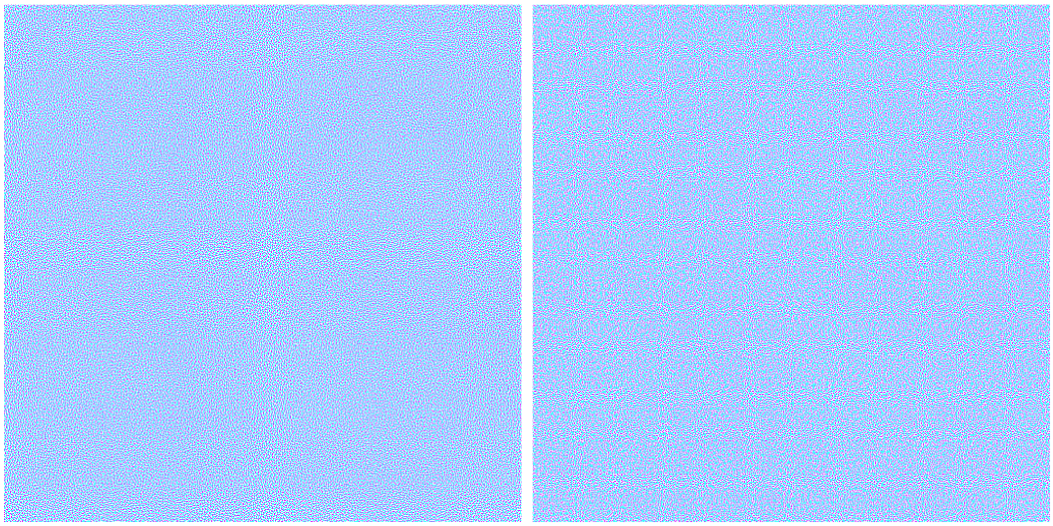
The halftone image of Fig 2.7 (a) and (b) has $E_{norm} = 1.4263$ and $E_{norm} = 1.9696$ respectively. With $g = 1$, the halftone image is smoother and more homogeneous compared with that with $g = 4$.

We also compared the effect on halftone when we use single and separate filters for red-green and blue-yellow channels, as shown below:



(a) $Y_yC_xC_z$ color space (b) L*a*b* color space

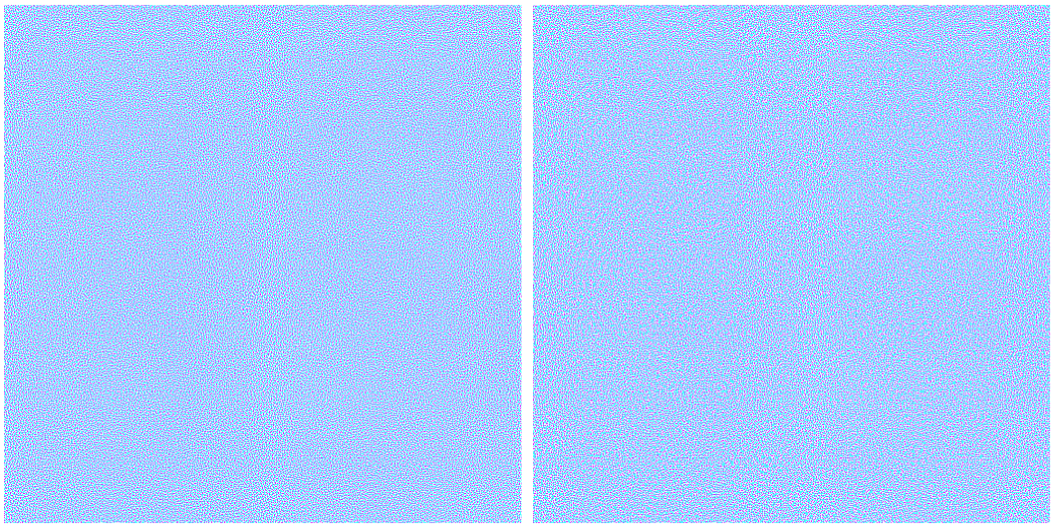
Fig. 2.8.: Comparison of halftoned images halftoned in difference color spaces.



(a) Gain factor 1 space (b) Gain factor 4

Fig. 2.9.: Comparison of halftoned images with different gain factors

The image in Fig. 2.8(a) is halftoned using single chrominance filter, and it has $E_{norm} = 1.4263$ and Fig. 2.8(b) is halftoned using separate chrominance channel filters, it has $E_{norm} = 1.3834$. We concluded that, using both red-green and blue-



(a) Single chrominance channel filter (b) Separate chrominance channel filters

Fig. 2.10.: Comparison of halftoned images with single and separate chrominance channel filters

yellow channels is better than using single chrominance channel filter in the sense that separate filters takes into account the difference between red-green and blue-yellow light absortance cones in the eye, thus the halftoned image are arrange more homogeneously than using a single chrominance filter.

2.5.2 Swapping Neighborhood and Time Complexity Reduction

Instead of swapping in a small neighborhood of 3×3. Here, the swap allowed is within 25×25 neighborhood, this allows more homogeneous halftone texture. Here's the monochrome halftone result if we allow swap only DBS in a neighborhood of 3×3:

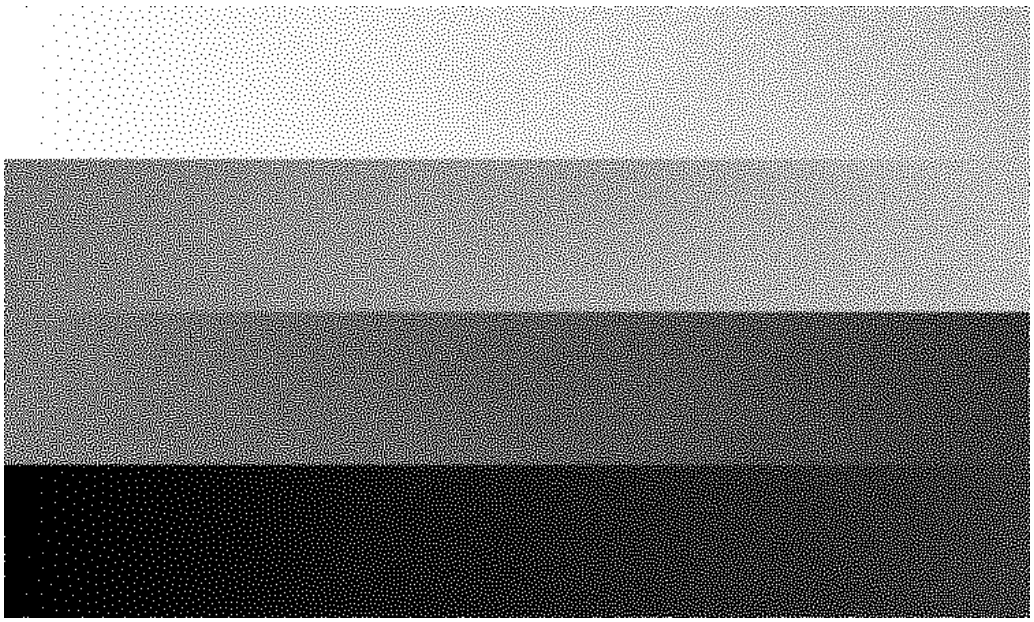


Fig. 2.11.: monochrome DBS halftone with swap neighborhood of 3×3.

Here's the halftone result when we increased the swapping neighborhood to 25×25:

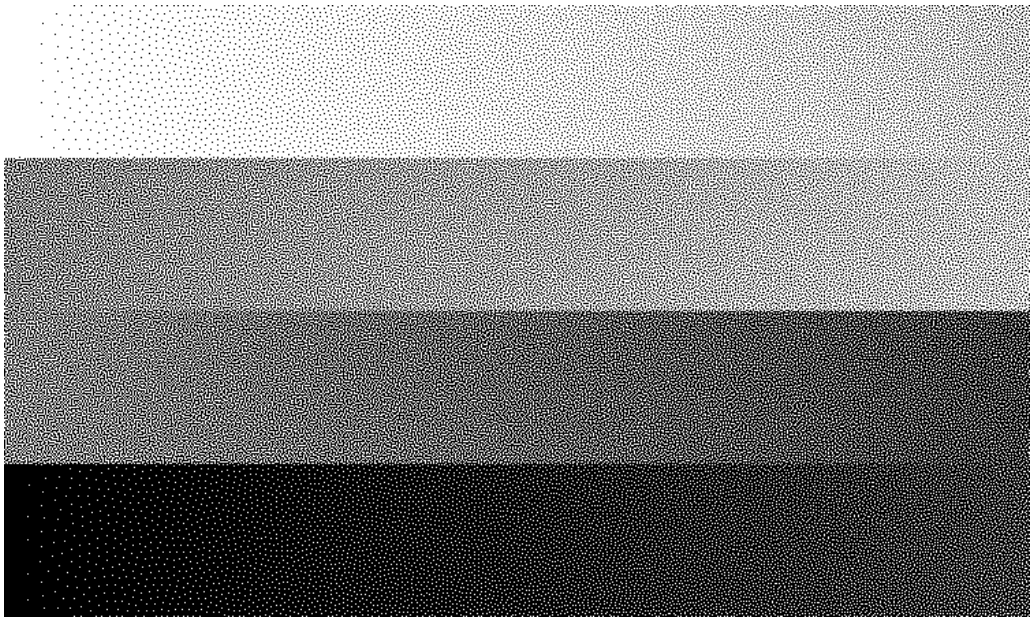


Fig. 2.12.: monochrome DBS halftone with swap neighborhood of 25×25.

It is clear that swapping in a neighborhood of 25×25 gives us more smooth halftone transitions. As Lieberman and Allebach [21] showed, the time complexity can be reduced by using a $\mathbf{c}_{\tilde{p}\tilde{e}}$ LUT, by limiting the search of possible swaps to those in the anticausal neighborhood of a given pixel, and by partitioning the pixels into small cells and processing each cell at a time in raster scan order.

2.5.3 Halftoning result of an image in sRGB color space

With Nasanen's luminance channel HVS filter and separate chrominance channel filters based on data collected by Mullen, we halftoned a colored image:

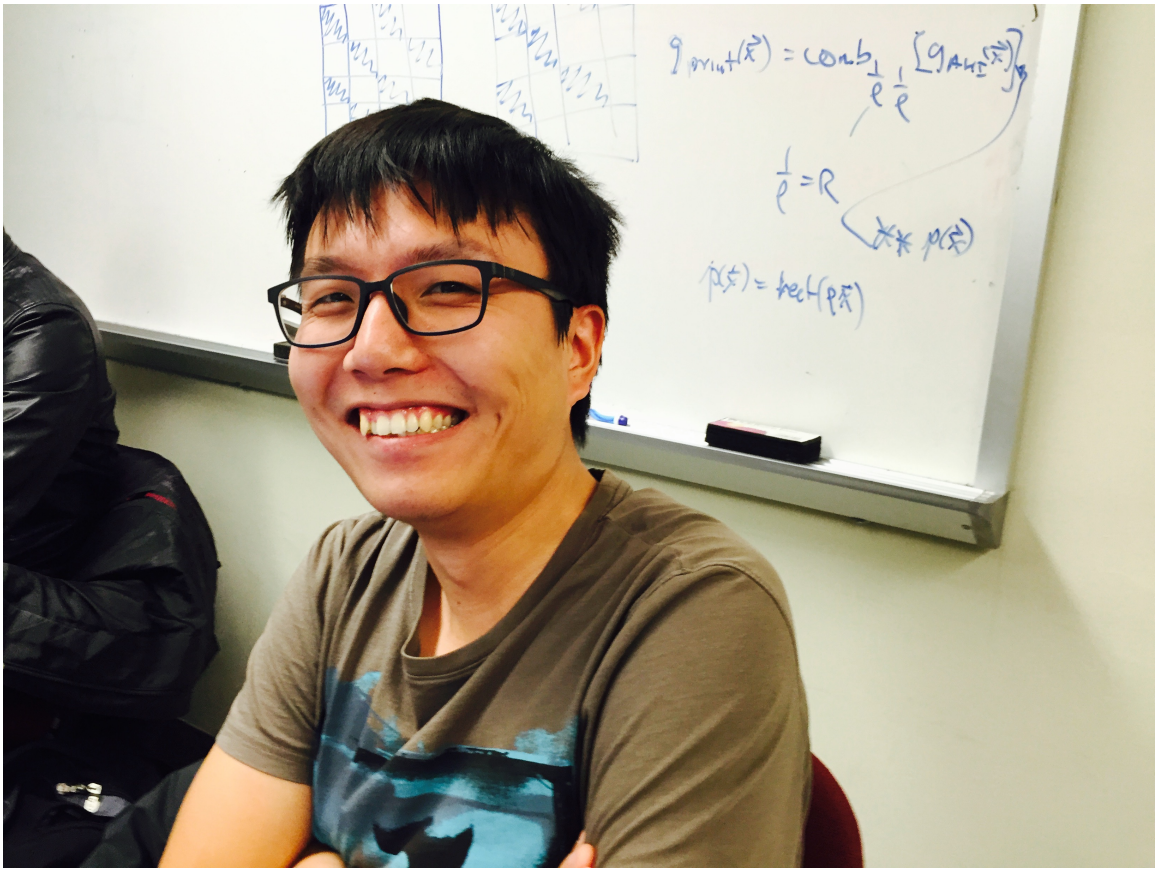


Fig. 2.13.: Original image in sRGB color space.

We have a selection matrix of size 256 × 256 which is generated by level-by-level monochrome DBS comprised of 255 levels, we can halftone the above image using the selection matrix, here's the halftone result:

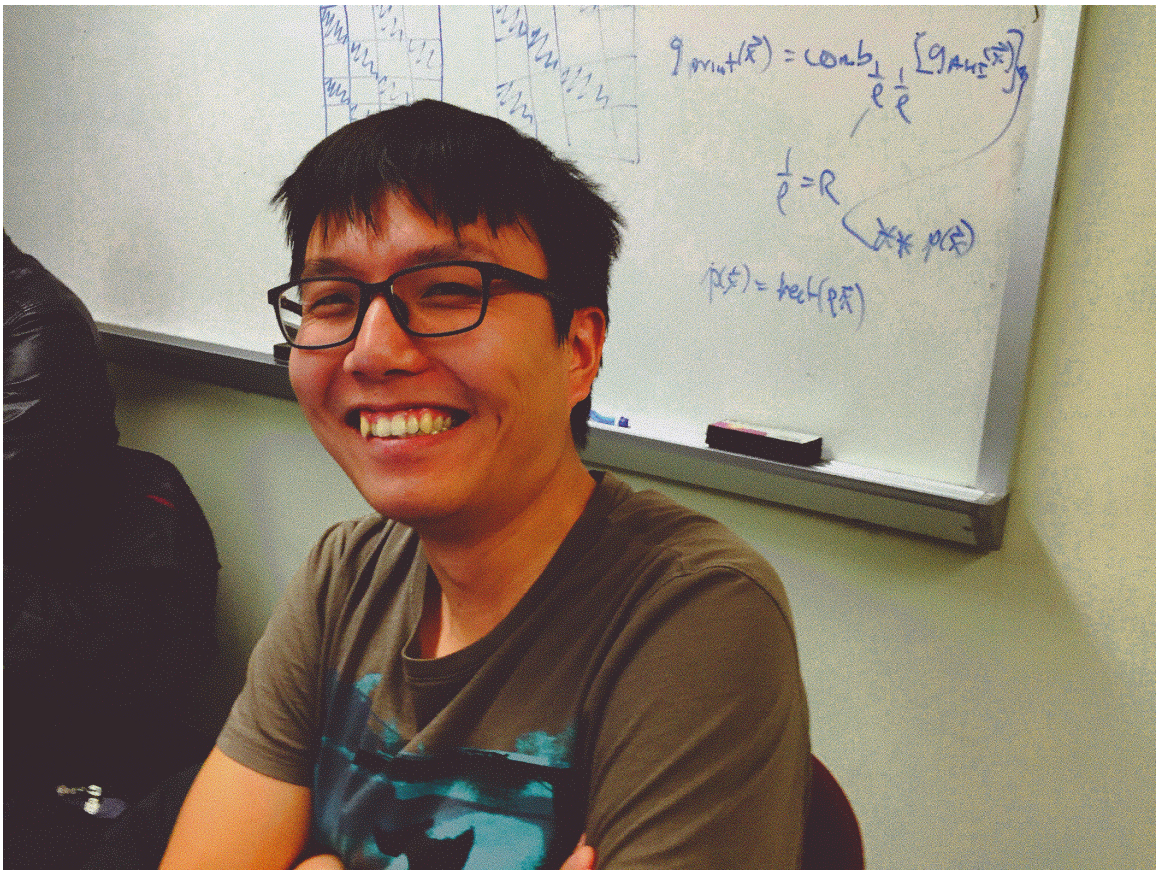


Fig. 2.14.: Halftoned image using selection matrix.

Below is an halftoned Ruiyi's image using Je-ho Lee's colorant-based DBS method [9]:

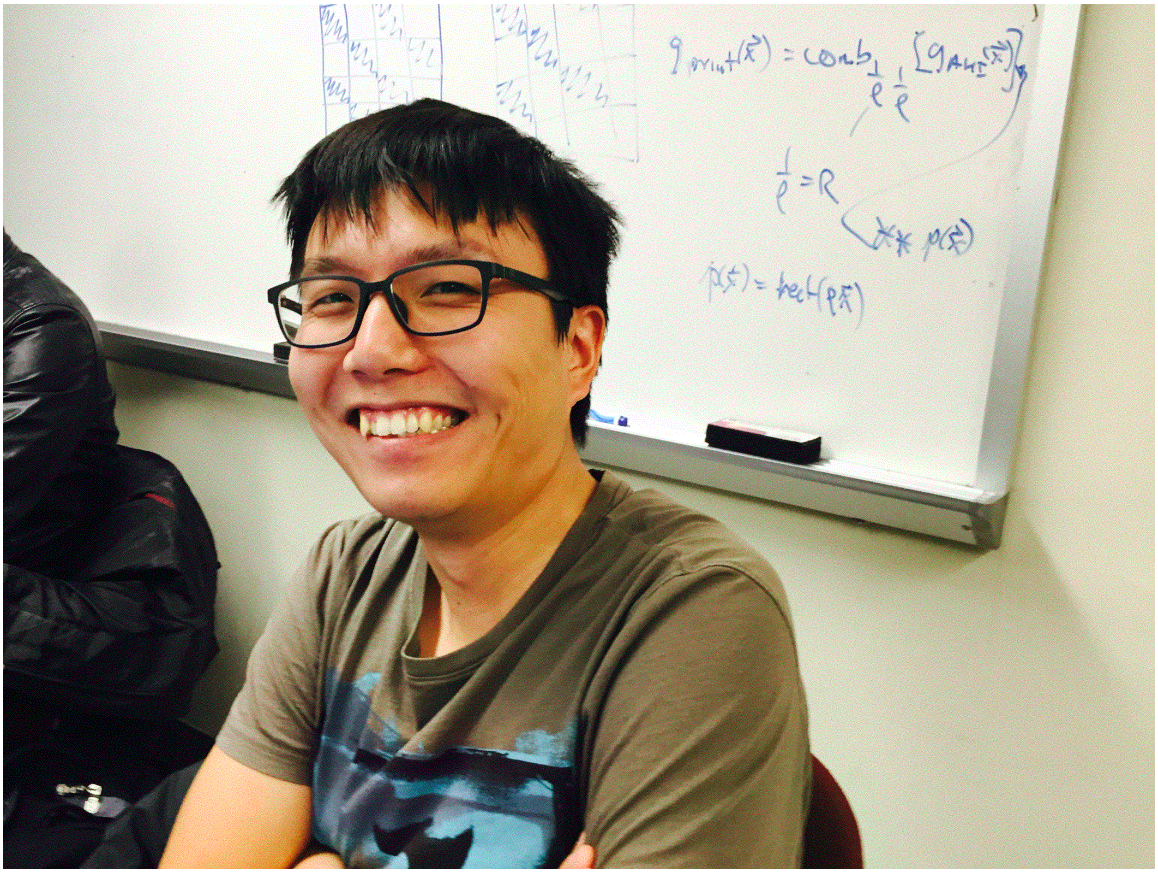


Fig. 2.15.: Colorant-based DBS Halftone.

Halftoning the original image using NPAC-DBS results in a better halftone result:

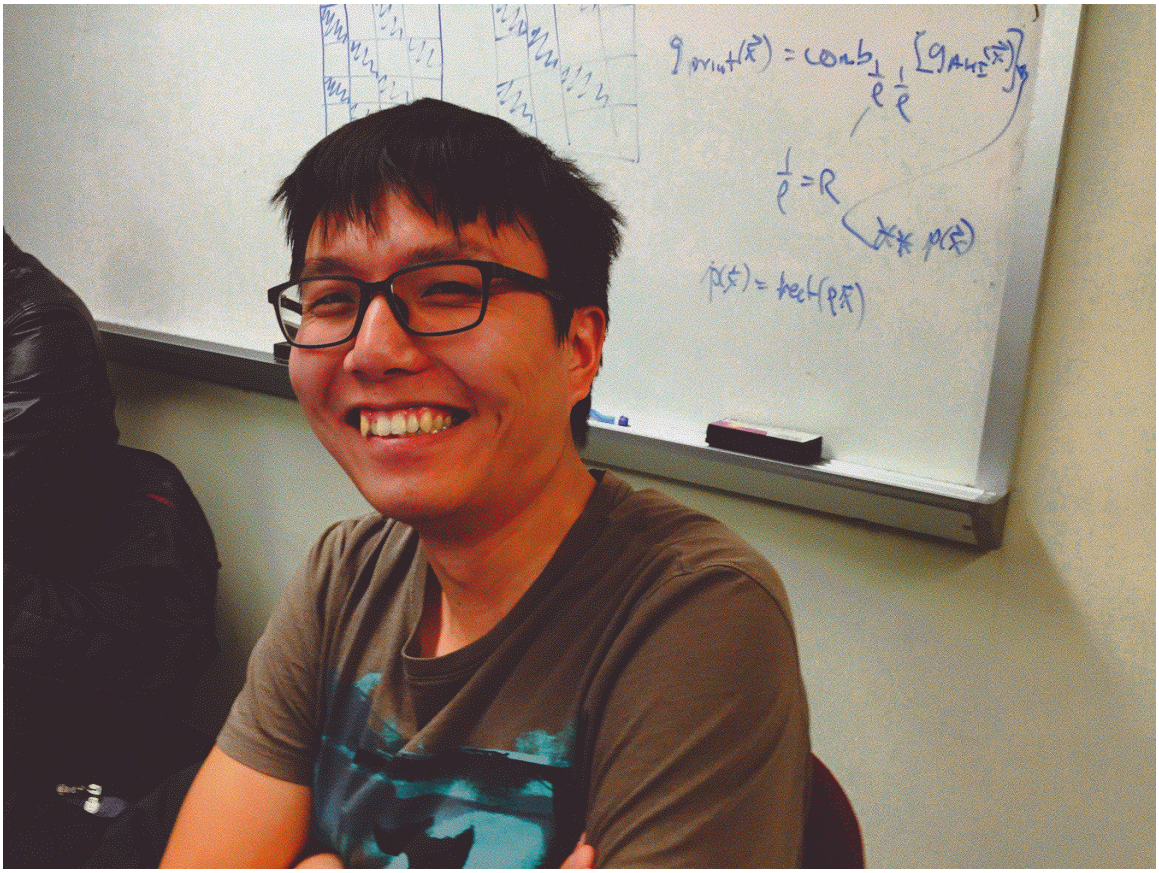


Fig. 2.16.: Halftoned image using NPAC-DBS.

Compared with halftoned image by selection matrix, NPAC-DBS is significantly better, especially in the glass area, it is more smooth and homogeneous than that of the selection matrix halftone. This is because NPAC-DBS has the knowledge of the image and halftone the image according to the texture, while selection matrix treat every pixel the same.

Compared with Je-Ho Lee's colorant-based DBS halftone [9], NPAC-DBS is also better, for its color is better in overall texture rendering. NPAC-DBS operates in YC_xC_z color space while Je-Ho Lee′s colorant-based DBS operates directly in the CMY colorant space.

2.5.4 Halftoning colored bulls-eye image

Our halftoning method can minimize moire effect on halftoned image. The original color bulls-eye image is shown below.

Fig. 2.17.: Original bull eye image.

Supercell method is a way to design screens $t[m, n]$, supercell enable us to design accurately to form an angled screen, each cell in the screen is a cluster of pixels that form a cell. With dot profiles $p[m, n; b]$, where b is the graylevel of the dot profile, which generated using supercell method, we can generate a screen by:

$$t[m,n] = 255 - \sum_{b=0}^{255} p[m,n;b] \tag{2.29}$$

Using Alty's periodical screen generated with supercell approach with parameters N1 = [9/2 -1;1 9/2] for yellow plane, N2 =[10/3 -10/3; 10/3 10/3] for Cyan plane and N3=[1 -9/2; 9/2 1] for Magenta plane to halftone the ripple image:

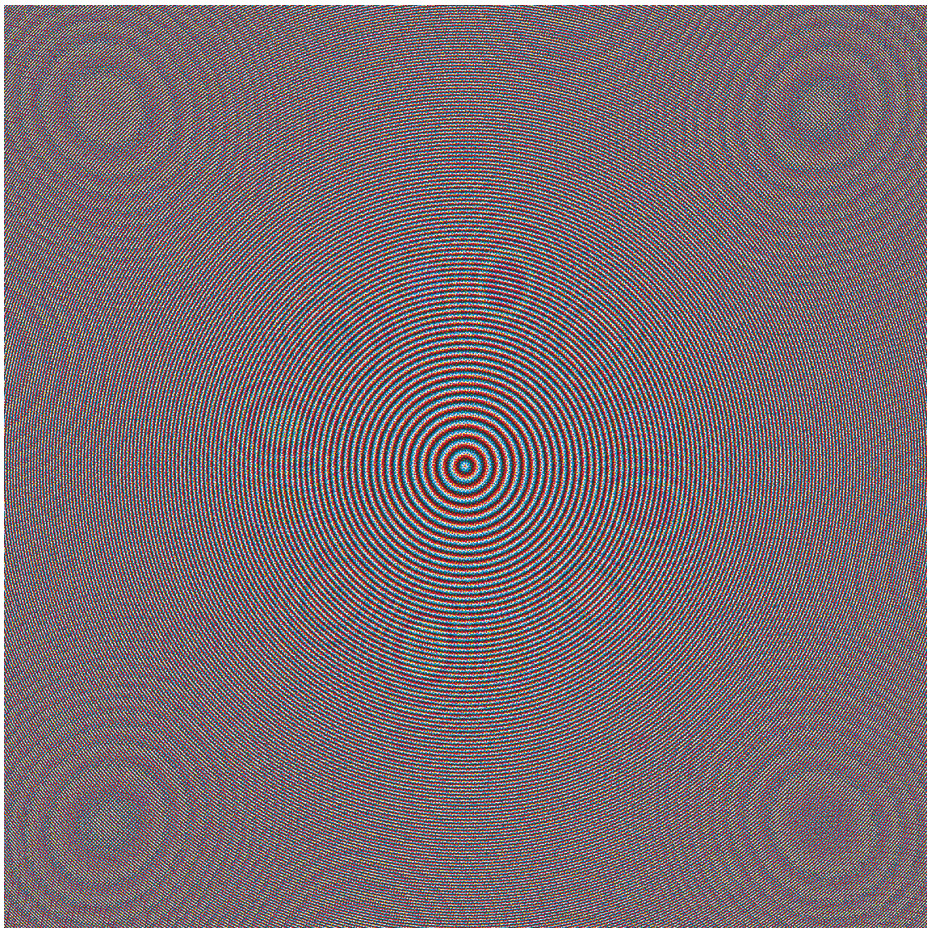


Fig. 2.18.: Halftoned images using selection matrix with supercell approach

The NPAC-DBS halftoned version of the bull′s eye image is shown below:

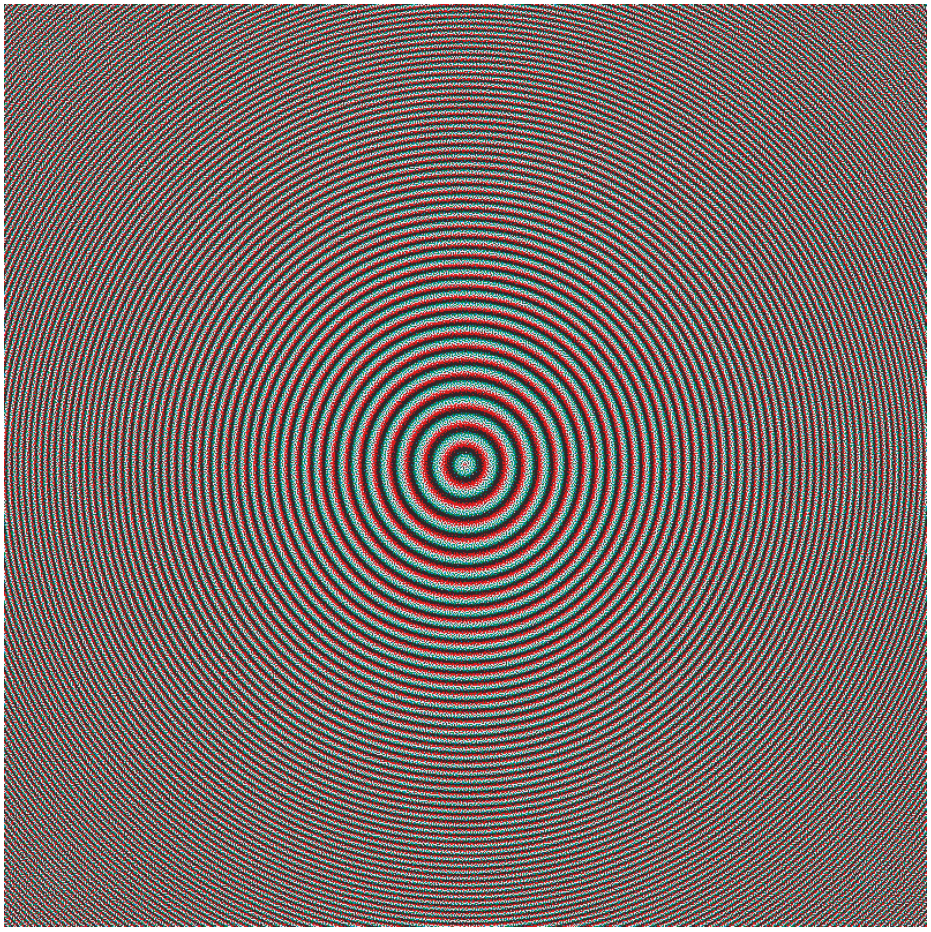


Fig. 2.19.: NPAC-DBS halftoned image.

Compared with screened halftone, the NPAC-DBS halftoned image significantly reduced moire effect.

2.6 Discussion and Conclusions

We compared different HVS filter models and concluded that using Nasanen's model for luminance channel and using separate chrominance filters for red-green and blue-yellow channel yields a more homogeneous halftone image. We also concluded that setting the luminance channel amplification factor to be one is better than four. Instead of swapping in a neighborhood of 3×3, we swap in a 25×25 neighborhood, it

also helped improving the quality of the halftoned image. Based on this observation, we developed NPAC-DBS algorithm that minimize the error metric in $Y_y C_x C_z$ color space. With sRGB to NPAC conversion pipeline, we are able to halftone any sRGB image using NPAC-DBS. We also observed that when there are closely arranged strip patterns in the original image, the moire is significantly reduced in the NPAC-DBS halftoned image. The set of Neugebauer Primaries can be potentially extended to support more colors than eight and render an image with wider gamut.

3. PARAWACS SCREEN DESIGN AND OPTIMIZATION

3.1 Introduction

Color halftoning is to create a halftoned image using limited amount of colorants, like monochrome halftoning, it needs to reproduce the color and the texture of the continuous-tone image. Different from creating monochrome halftone, it needs color management to mimic the color of the original image as much as possible.

Previously, we used NPAC DBS halftone method to create halftone using Neugebauer Primary Area Coverage (NPAC) continuous-tone image, this method creates a very smooth halftone. In other previous works, error diffusion [22] is also used in the halftoning of NPAC images. The limitation of error diffusion is obvious, the throughput is low, and there are some attempts at parallelizing error diffusions. [23] Traditionally, color halftoning is generated using plane independent halftones, when these planes combine together, a full color halftone is created. The individual halftones are created using matrices, each plane, take Cyan, Magenta, Yellow and Black for example, has its own matrix that minimize moire effects, or that has plane dependency on certain pairs of colorants [24] to minimize dot on dot print. For traditional color halftoning, we separately halftone each plane, resulting in an unknown color reproduction, and we have no control over their additive texture, only individual plane texture is arranged, when halftones of different color plane added up together, some pixels are overlapping, it is hard to control parts that overlaps, and parts that should separate. With the concept of Neugebauer Primaries (NPs) [25], we have fixed amount of colors, only white, cyan, magenta, yellow, red, green, blue, black are in the halftone. The selection matrix select from these defined or controllable set of colorants, and create halftone that follow certain texture defined by the selection matrix. The selection matrix perform a selection function, it choose the color from

the colorants in each pixel. After the halftoning, the color is well controlled and can better reflect the color of the continuous halftone.

An NPAC continuous-tone image is an image consists of N planes, where N is the number of colorants that define the halftone. For each pixel in a certain plane, the percentage of the color in the plane is specified. With a given NPAC image, we can create the halftone using the PARAWACS selection matrix, the mechanism of the way it works is illustrated by an example. Assuming that there is a sufficiently large area of a constant NPAC image, the likelihood of picking one of the NPs from the NPAC of the area is equivalent to the area coverage of the NPAC image corresponding to that NP. [26]

The way PARAWACS works is fundamentally different from traditional halftoning, this novel method proposed by Peter Morovič and Jan Morovič is fundamental to both NPAC DBS and PARAWACS. In my work, I found the optimum parameters to generate the PARAWACS matrix, including optimizing the texture using DBS and minimizing low frequency noise in the resulting matrix.

The rest of the document is organized as follows: In section 3.2, we introduce PARAWACS framework. In section 3.3, We illustrate how PARAWACS selection matrix is designed using DBS. Next, in section 3.4, we summarized the parameters we found that yield the best selection matrix. Finally, in section 3.5, we show an image halftoned using the new selection matrix by our sponsor in Barcelona.

3.2 PARAWACS framework

In this part, we illustrate the halftoning processing in formal equations. Color halftoning with a PARAWACS selection matrix is based on NPAC continuous-tone image separations. For each pixel in the NPAC continuous-tone separations there is an associated NPAC, the area coverages (AC) are less or equal to 1 and all the ACs sum to 1:

$$AC_{NP_0} + AC_{NP_1} + ... + AC_{NP_7} = 1$$

Heres an example of part of Ruiyi's NPAC image set (pixels sampled: $[1,1]$, $[1,2]$, $[2,1]$, $[2,2]$) arranged from $NP0$ to $NP7$:

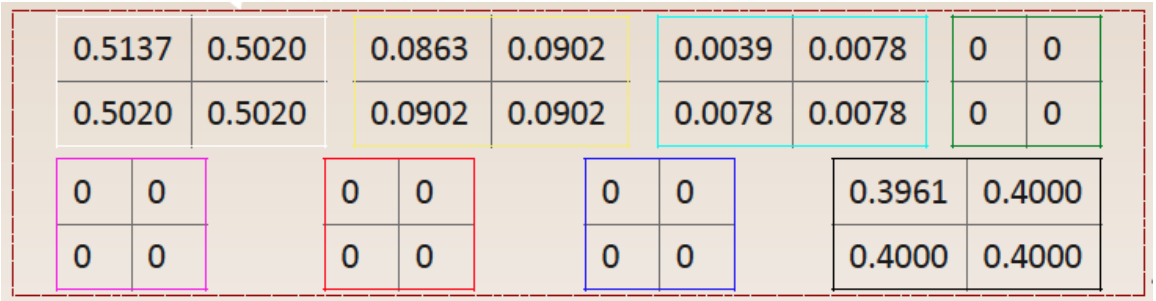


Fig. 3.1.: NPAC example

For the first pixel, we write the area coverage as follows:

$$\begin{cases} AC_{NP_0} = 0.5137 \\ AC_{NP_1} = 0.0863 \\ AC_{NP_2} = 0.0039 \\ AC_{NP_3} = AC_{NP_4} = AC_{NP_5} = AC_{NP_6} = 0 \\ AC_{NP_7} = 0.3961 \end{cases}$$

The sum of the area coverage (AC) of all the NPs at the first pixel is 1:

$$AC_{NP_0} + AC_{NP_1} + ... + AC_{NP_7} = 0.5137 + 0.0863 + 0.0039 + 0.3961 = 1$$

The NPs are arranged from lightest to darkest: W, Y, C, CY (G), M, MY (R), CM (B), CMY

PARAWACS selection matrix is a level by level designed matrix using monochrome DBS, each level tries to achieve homogeneous and uniform texture due to the virtue of DBS. We will illustrate the design process later. In the selection matrix, each element is assigned a natural number ranging from 0 to the maximum level L_{max}, in our matrix design, $L_{max} = 254$.

The way we perform halftoning based on PARAWACS selection matrix is the same as the method of the initial NPAC halftone image is generated in the previous chapter.

3.2.1 Halftoning procedure

1. Gamma uncorrect the original image, then based on the linear sRGB values, generate NPAC image set.

2. The order of NPAC continuous-tone image separation is arranged from lightest to darkest: W, Y, C, CY (G), M, MY (R), CM (B), CMY corresponding to $NP_0, NP_1, NP_2, NP_3, NP_4, NP_5, NP_6$, and NP_7 respectively.

3. Halftone the image based on the selection matrix and NPAC separation, resulting in assigning a single NP to each pixel of the image.

4. For display purpose, we need to replace each NP with the corresponding linear sRGB values to obtain the linear sRGB image.

	NP_0	NP_1	NP_2	NP_3	NP_4	NP_5	NP_6	NP_7
sR	216	236	0	0	187	186	6	7
sG	216	118	79	67	1	2	3	4
sB	223	0	177	11	44	2	41	5

5. Finally, gamma correct the linear sRGB image.

The result of halftone is in Chapter 1, Fig. 1.8.

3.3 Design PARAWACS selection matrix using DBS

The original design procedure for PARAWACS selection matrix is as follows:

1. Create seed halftone, generate the seed halftone of midtone (127).

2. Design dot profile for all levels $p[m, n; b]$ in chosen sequence, where b=0,1,,,L_{max}

(255).

3. Combine the dot profiles and form the screen $g[m, n]$:

$$t[m, n] = 255 - \sum_{b=0}^{255} p[m, n; b] \tag{3.1}$$

4. Verify screen correctness.

5. Use the screen to halftone an image.

Here, we choose the scale factor to be 3000.

3.3.1 Dot profile generation sequence

Midtone to extreme tone sequence and binary tree sequence is compared and we find that the former yield a better result. We generate the dot profile patches sequentially from midtone to the extreme tones, ordering of lower half is sequentially decreasing, and for upper half is sequentially increasing. The size of the dot profiles are 256×256. There are 256 total dot profiles generated. White dots are assigned 0 and black dots are assigned 1.

3.3.2 The stacking constraint

The following graph illustrates the stacking constraint: If the pixel in the lower level is black, all corresponding pixels in upper levels are black; If the pixel in the upper level is white, all corresponding pixels in lower levels are white; And if upper pixel is black and lower pixel is white, we can choose to set the current pixel white or black.

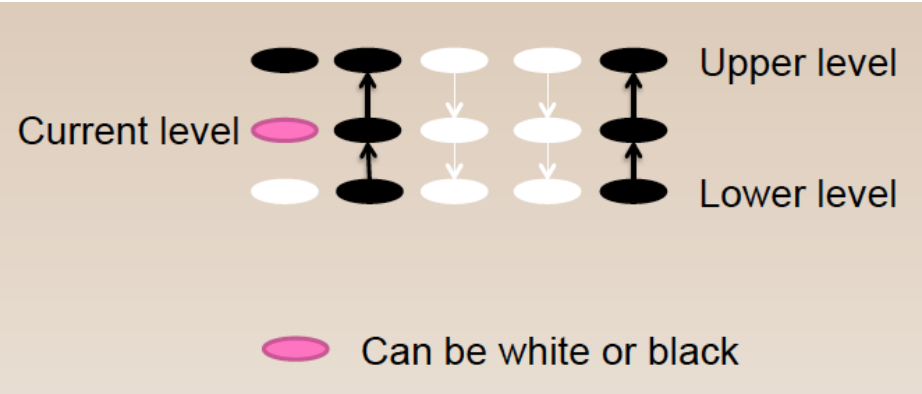


Fig. 3.2.: The stacking constraint.

Next, we compare the result of the screened ramp halftone with DBS halftoned ramp image:

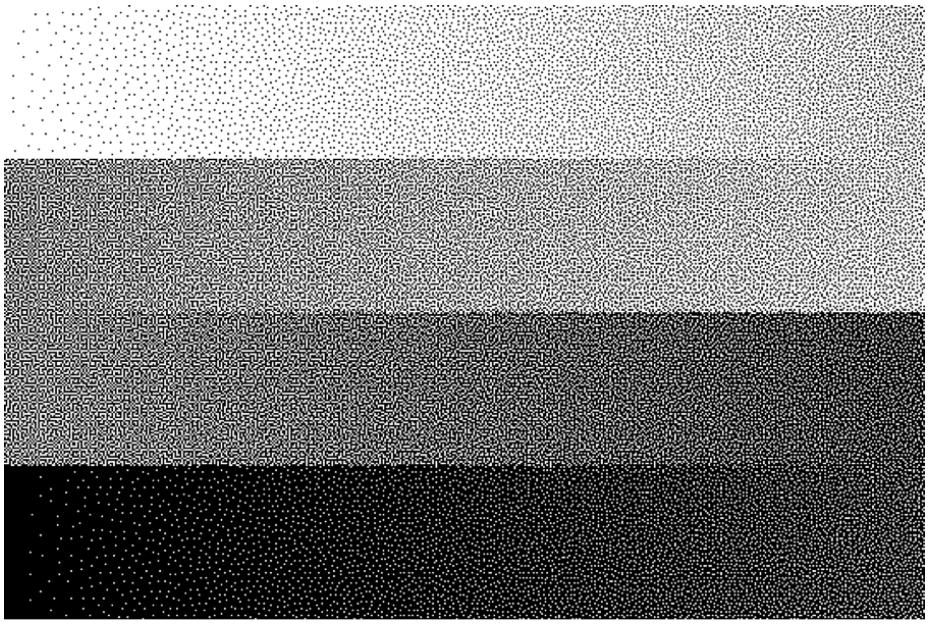


Fig. 3.3.: Screened halftone image.

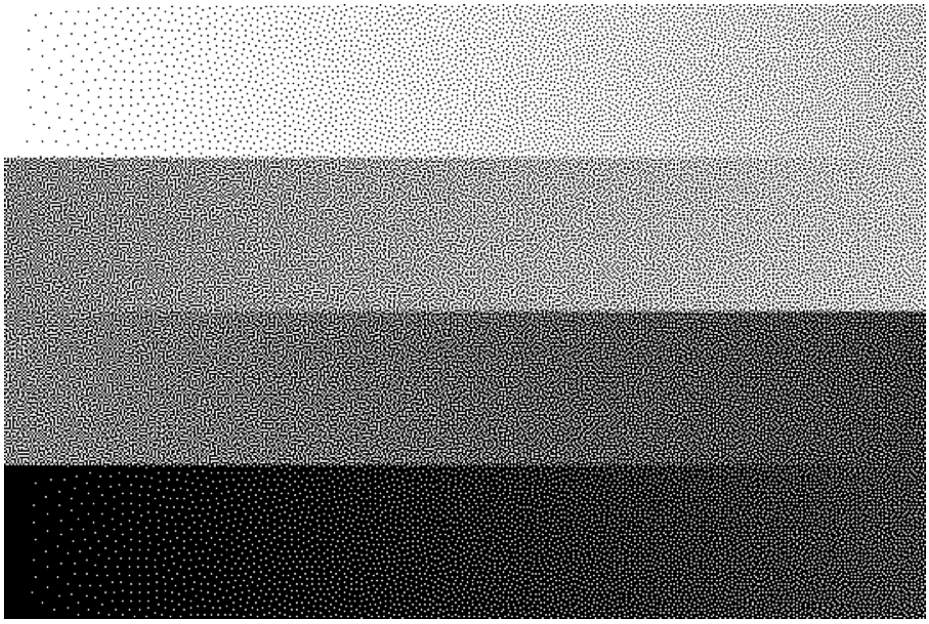


Fig. 3.4.: Halftoned ramp image using DBS.

Due to the fact that both swap and toggle is enabled during the halftoning process, the tone of the screen is shifted, the tone change while graylevel goes up is not uniform. The desired number of dots change is:

$$N = *dpl \cdot (\gamma + 1) - *dpl \cdot \gamma$$

Where γ is current dot profile level.

$$dpl = \begin{cases} \frac{height \cdot width}{MAXLEVEL+1}, \text{lower half of screen} \\ \frac{height \cdot width}{MAXLEVEL-1}, \text{higher half of screen} \end{cases}$$

Note: MAXLEVEL is an even number, and we set MAXLEVEL/2 as midtone.

3.3.3 Swap neighborhood

Increasing swapping neighborhood will result in better halftone quality, we investigated 7×7, 19×19, 25×25 swap neighborhood and found that 25×25 neighborhood is better among these choices.

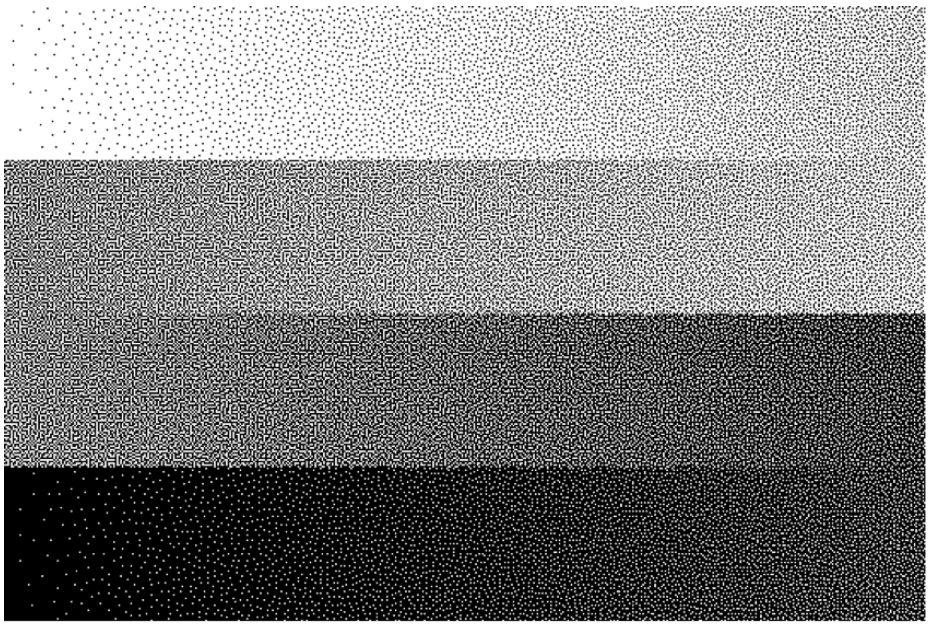


Fig. 3.5.: Screening result using screen generated with swap neighborhood of 25x25

Visually, the halftone result is similar to that generated by DBS, but when printed, we observe low frequency artifacts. In order to remove this defect, we need to find out which parameters give us this defect and change these parameters to its best configurations to eliminate the defect, further, we need to find the best parameters to improve the quality.

3.4 Improve the quality of screen

The goal of this part is to improve the design of PARAWACS due to the fact that the quality of the highlight pattern is not good, and partially because the differential

patterns also needs to be improved, we will not talk about differential patterns in this document.

First, we look at the search and cost function, and the parameters we changed are Scale factor, spread of point spread function and swap neighborhood.

In order of asses the quality of halftone, we have three ways:

1. Visual inspection

   Gold standard: single level unconstrained DBS design.

2. Characteristics of halftone spectrum

   (a) Visual inspection based on logarithmically scaled magnitude spectrum.

   (b) Ulichneys metrics:

      i. RAPS

      ii. Anisotrpy (not used here)

   (c) Changs metrics (not used here)

3. Quality of differential halftone patterns

   The reason we care about this is because these are used to represent NPs, especially for small amounts. We have come up with methods to improve this part, but the result is not finalized yet, so we do not illustrate it in this document.

Next, we will find ways to reduce low frequency energy of the screen, we compared swap only with swap and toggle methods, examined the effect of swap neighborhood and block size, and corrected tone shifts that may occur when toggling.

We observed that using swap only option in DBS PARAWACS matrix design gives us low frequency energy, which is undesirable, we explored three ways to minimize the low frequency energy:

1. Using swap and toggle

2. Increasing swap neighborhood when using swap only option
3. Changing filter size or scale factor

The optimum parameters we choose for designing screen of size 256×256 are: Scale factor = 3500, Point spread function radius = 23 pixels (99% coverage), Swap neighborhood = 255×255, Block size = 5×4, Sequence: Utpal′s alternate sequence, which is shown below:

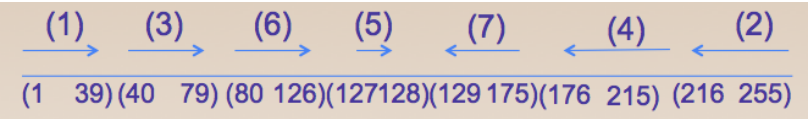


Fig. 3.6.: Utpal′s alternate sequence

3.5 Experimental result

The halftone result produced by our sponsor in Barcelona using the selection matrix generated using the optimum parameters we choose is shown below:

Fig. 3.7.: Final PARAWACS halftone result

We can see from the above figure that it is very smooth and homogeneous, the low frequency noise is successfully removed, highlight part of the halftoned image is also very good. Our objective to improve the selection matrix quality is realized.

PARAWACS can be used in secure printing, and it has some advantages over other secure printing techniques like watermarking, which is more commonly used on continuous tone data. [27] The idea of manipulating halftones directly in order to embed patterns has been explored before. [28] However, these methods are not suitable for color halftone. so PARAWACS has the advantage over other methods in secure printing.

4. NOVEL COLOR HALFTONING ALGORITHM FOR INK SAVINGS

4.1 Introduction

Halftone images are categorized according to their texture. They are distinguished by the attributes of the dots comprising the halftone, which are dispersed-dot and clustered-dot. Another way to categorize the halftone is by the periodicity of the halftone texture, whether it is periodic or aperiodic. The halftone can be any one of the four combination of these two categories. In this dissertation, we primarily discuss clustered-dot periodic halftones. We create such a halftone by generating a halftone screen that is clustered-dot and periodic. the clustered-dot screen has an advantage over dispersed-dot screens due to the fact that it is beneficial for printers with unstable output. Wang et al. discussed a new screening technology called concentric screening for inkjet printers, their idea is similar to our approach since their method divides round AM dots into rings with the same centering but it does not reduce the overlap between black and other colorants. [29]

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

Multilevel halftoning is possible with both laser electrophotographic and inkjet printers. With the former, either laser amplitude modulation or pulse-width modu-

Fig. 4.1.: HP c500 Press

lation is possible. With the latter, multiple drops can be placed at a single printer-addressable pixel.

There are two types of printers: inkjet and laserJet printers. Inkjet printers are generally smaller and cheaper compared to laserjet printers. However, the inkjet printer we target for is large-format inkjet printer as shown below:

This is a 4-ink press for high-speed printing of corrugated cartons. It prints at 245 linear ft./min. The inks are water based. It supports printing of up to 4 drops of each ink at each printer-addressable pixel, but there are ink limitation.

We introduce a novel halftoning method, ink-saving, single-frequency, single-angle, multidrop (IS-SF-SA-MD) halftonig method that targets for this printer. It reduces the overlap between different colorants and saves ink at the same time. The algorithm incorporates a multilevel halftone method to increase the number of gray levels, and to reduce the visibility of dots in highlight and shadow regions. The traditional halftone screen is replaced by a hybrid screen to optimize the texture of the printed halftone patterns. The hybrid screen design is based on Lee and Allebach's supercell method.

4.2 Design rule and design goal

4.2.1 Design rule

In order to reduce the overlap between black and other colorants, we place black dots on grid with 1/2 shift in both directions, as shown in Fig. 4.2.

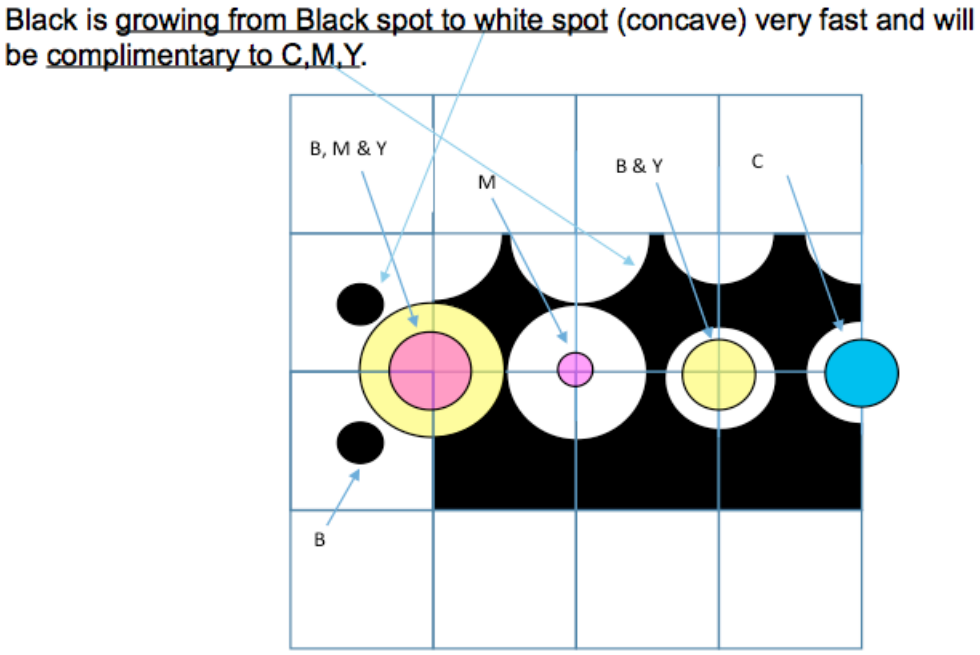


Fig. 4.2.: Design Rule

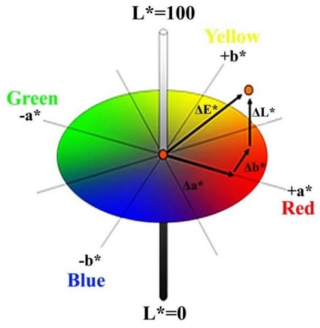


Fig. 4.3.: CIE L*a*b* color space [30]

Cyan (C) is placed first in the center of CMY, then magenta (M) is placed in a ring above C, and yellow (Y) in a ring above M.

There are several rules we are observing:

- 1. Highlight and shadow region should be homogeneous.
- 2. All colorants follow traditional 1-2 schedule.
- 3. We will not print 2 drops on any pixel until every pixel has at least 1 drop. We will not print 3 drops on any pixel until every pixel has at least 2 drops. We will not print 4 drops on any pixel until every pixel has at least 3 drops. (Leveling Rule)
- 4. We will not print anything on 3 drops of K. (Ink Efficiency Rule)
- 5. We cannot print more than 4 drops on any pixel. (Ink Limit Rule)

The reason that we do not allow any colorants to be print on 3 drops of K (Rule 4: Ink Efficiency Rule) is that it does not alter the color much when we put other colorants on 3 drops of K. Let's observe some examples in L*a*b* color space:

Here, we look at 4 cases on a*b* plane:

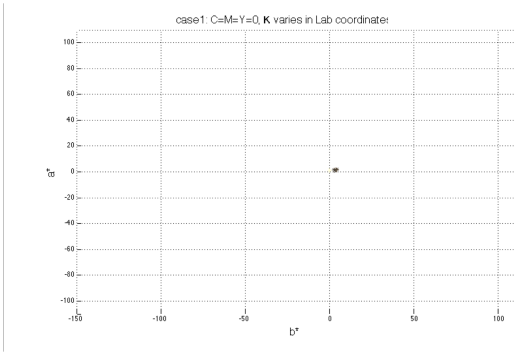
Case 1. C=M=Y=0, K increases (from 0, 1, 2 to 3 drops).

Case 2. 1 drop of C, M=Y=0, K increases.

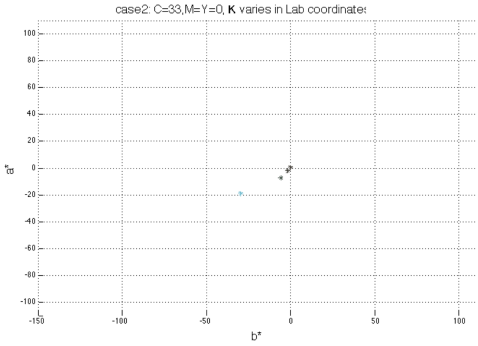
Case 3. 2 drops of C, M=Y=0, K increases.

Case 4. 3 drops of C, M=Y=0, K increases.

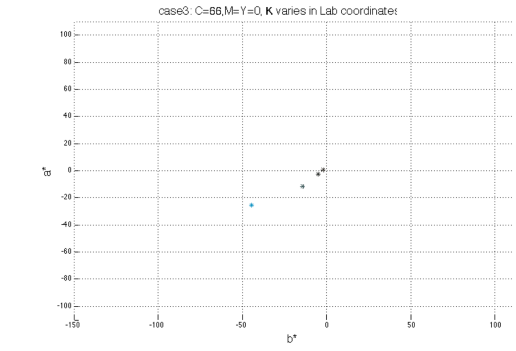
Based on the above observation, the color combinations approach the original point on a*b* plane with K increases regardless of how many drops of cyan ink there are. When there are 3 drops of K, the color combination sits in the vicinity of the



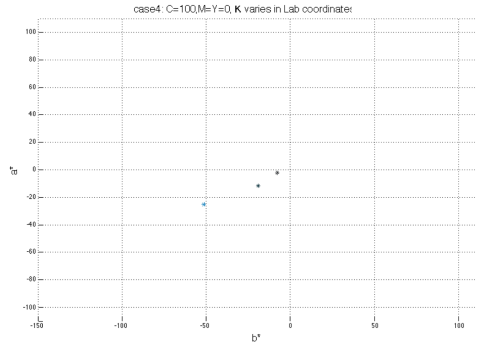
(a) Case 1. C=0, K increases



(b) Case 2. 1 drop of C, K increases



(c) Case 3. 2 drops of C, K increases



(d) Case 4. 3 drops of C, K increases

Fig. 4.4.: Observation of different color combinations with various of K in a*b* plane

original point in a*b* plane (the neutral point). This observation is true for other combinations as well. Since the change in color when there are 3 drops of K is present is minimal, we do not allow any colorant to be printed on 3 drops of K.

4.2.2 Design goal

For microcell design, we need to:

• 1. Maintain existing dots or hole cluster, we are adding new dot pixel on the boundary of existing dot or hole cluster.

• 2. Maintain symmetry of dot and hole clusters as much as possible.

• 3. We will always take into account the periodicity of halftone screen.

For hybrid screen design:

• 1. Core size should create good stochastic-dot texture.

• 2. Highlight and shadow region should be homogeneous.

4.3 Major components

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

4.3.1 A. Color management

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

4.3.2 B. Gray component replacement (GCR)

The gray component is a mixture of three colorants: cyan (C), magenta (M), and yellow (Y). With different amounts of the CMY mixture, we have different levels of lightness. The idea of substituting some amount of black for the gray component is known as gray component replacement, it reduces the expense of printing stabilize the printing process [31]. The GCR method in the work of Davor et al. is content dependent, depending on whether the content of images contain many replaceable colors or not, ink savings due to GCR can vary [32]. Content-dependent GCR method use more aggressive GCR in detail regions where K dots are not so visible and they provide higher contrast rendering of detail. GCR can also be tone-dependent.

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

$$\frac{I_{post\_gcr}}{J_{post\_gcr}} \neq \frac{I_{request}}{J_{request}} \tag{4.1}$$

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

Regardless of the amount of GCR, if $I_{request} = J_{request}$ then $I_{post\_gcr} = J_{post\_gcr}$.

Next, we discuss the way in which the GCR is performed.

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

$$I_{post\_gcr} = I_{request} - \alpha \cdot R^{max} \tag{4.2}$$

$$K_{post\_gcr} = K_{request} + \alpha \cdot R^{max} \tag{4.3}$$

where $I = C, M, Y$.

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

4.3.3 C. Ink limitation

After GCR, we enforce ink limits with continuous-tone image prior to halftoning. Let $C_{post\_gcr}, M_{post\_gcr}, Y_{post\_gcr}, K_{post\_gcr}$ be the ink amounts after GCR. And the summation is calculated as $^{total}_{post\_gcr} = C_{post\_gcr} + M_{post\_gcr}$ $+ Y_{post\_gcr} + K_{post\_gcr}$ Let $C_{actual}, M_{actual}, Y_{actual}, K_{actual}$ be the actual ink amounts that are printed. If $I^{total}_{post\_gcr} \leq \frac{4}{3}$, no ink limitation is necessary, because theoretically, we can put a maximum of 4 drops at any given pixel. However, the halftoning algorithm will restrict the gamut even more. And we will discuss this in Section 4.4. In the case where ink limitation is not needed, we have

$$\begin{cases} C_{actual} = C_{post\_gcr}, \\ M_{actual} = M_{post\_gcr}, \\ Y_{actual} = Y_{post\_gcr}, \\ K_{actual} = K_{post\_gcr} \end{cases} \tag{4.4}$$

When $I^{total}_{post\_gcr} > \frac{4}{3}$ We perform ink limitation. The process of how we perform ink limitation is as follows:

The actual amount of colorant that is input to halftoning algorithm is

$$I_{actual} = I_{post\_gcr} \frac{4/3}{\bar{I}^{total}_{post\_gcr}} \tag{4.5}$$

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

$$\bar{I}^{total}_{actual} = C_{actual} + M_{actual} + Y_{actual} + K_{actual} \tag{4.6}$$

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

$$\frac{I_{actual}}{J_{actual}} = \frac{I_{post\_gcr} \frac{4/3}{\bar{I}^{total}_{post\_gcr}}}{J_{post\_gcr} \frac{4/3}{\bar{J}^{total}_{post\_gcr}}} = \frac{I_{post\_gcr}}{I_{post\_gcr}} \tag{4.7}$$

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

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

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

4.3.4.1 D.i. Traditional concepts

4.3.4.1.1 Specify lpi and angle

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

4.3.4.1.2 Choose tile vectors

A periodic screen can be defined by two vectors, $\mathbf{z} = [z_i, z_j]$ and $\mathbf{w} = [w_i, w_j]$ as shown in Fig. 4.5. Here, vectors $\mathbf{z}$ and $\mathbf{w}$ are screen tile vectors, and the shape formed by these vectors defined the continuous-parameter halftone cell (CPHC). Screen tile vectors have the property that if we shift and copy the CPHC by $n_1\mathbf{z}+n_2\mathbf{w}, (n_1, n_2) \in \mathbb{Z}^2$, then we can cover the entire spatial domain.

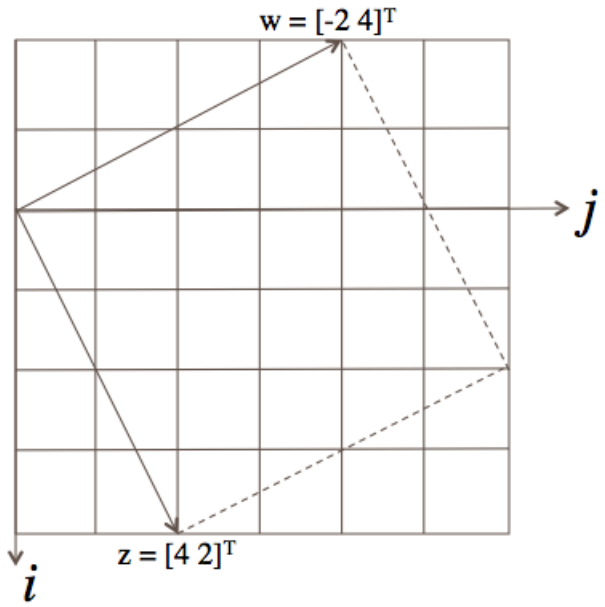


Fig. 4.5.: Continuous-parameter halftone cell (CPHC) of microcell design with parameters $\mathbf{z} = [4, 2]$ and $\mathbf{w} = [-2, 4]$.

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

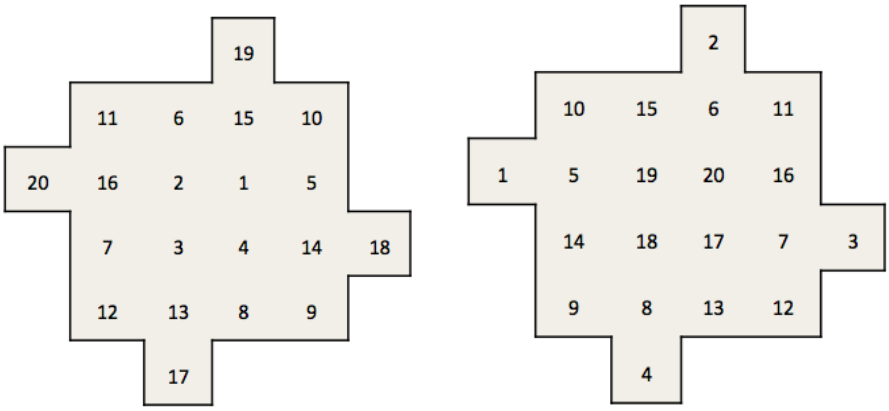
and $\mathbf{z} \cdot \mathbf{w} = 0$, we have a rectangular screen, and if $|\mathbf{z}| \neq |\mathbf{w}|$ and $\mathbf{z} \cdot \mathbf{w} \neq 0$, we have a parallelogram screen with unequal sides.

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

4.3.4.1.3 Determine the discrete microcell size and shape

Based on the CPHC design, we need to determine the shape of the block with boundaries lying on the grid of printer-addressable points, which also matches with the CPHC. This is defined to be the discrete-parameter halftone cell (DPHC) design corresponding to the CPHC, with the same frequency and angle parameters.

The DPHC is not rectangular shaped, but in order to be used for computation purpose, we need to tile the microcell to form a rectangular screen using the Holladay method. The halftone cell defined by CPHC has fraction of a pixel inside the halftone cell, the DPHC cell is chosen based on the shape of CPHC. Fractional area of the pixel covered by the CPHC is not calculated, and the pixels that has more area covered by the CPHC may not be the ones chosen to be included. In the DPHC cell chosen, the total area of the CPHC and the DPHC are the same.



(a) DPHC for black (b) DPHC for CMY

Fig. 4.6.: Discrete-parameter halftone cell (DPHC) of microcell designed with parameters $\mathbf{z} = [4, 2]$ and $\mathbf{w} = [-2, 4]$

4.3.4.1.4 Determine BSB

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

$$H_B = \left| \frac{N_m}{gcd(z_j, w_j)} \right|, W_B = \left| \frac{N_m}{gcd(z_i, w_i)} \right| \tag{4.8}$$

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

4.3.4.1.5 Determine the dot growth rules

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

• Maintain existing dot- or hole-clusters by adding new dots on the periphery of the dot-cluster and adding new holes on the periphery of the hole-cluster. This rule is essential for maintaining the clustered-dot/clustered-hole structure of the halftone patterns.

• Maintain symmetry of dot and hole clusters as much as possible. This is especially essential in midtones, where a perfectly complementary-symmetric design will form a checkerboard pattern.

• Take into account the periodicity of the halftone screen. According to these rules, we designed the microcell growing sequence, numbered 1 to 20, as shown in Fig. 4.6. Note that the design is not unique.

4.3.4.2 D.ii. Generalizations

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

4.3.4.2.1 Supercell design

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

$$H = M_H H_B, W = M_W W_B \tag{4.9}$$

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

Next, we need to design the macrocell sequence. This sequence is the order in which dots are added to each microcell. For example, starting in the highlights, we turn on the first pixel (numbered 1 in each microcell) in the macrocell sequence. In

order to have a good quality of the first dot turn-on sequence, we use DBS to determine the macrocell sequence.

4.3.4.2.2 Hybrid screen

Our hybrid screen creation is similar to Lee and Allebach's hybrid screen generation method [33]. The target hybrid screen has the property of complementary symmetry.

4.3.4.2.2.1 Highlights and shadows generation

The highlight region, corresponding to pixels with indices 1 through 4 in Fig. 4.3, is generated using DBS. The total number of levels L is the total number of pixels in the hybrid screen plus one.

$$L = L_M N_m + 1 \tag{4.10}$$

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

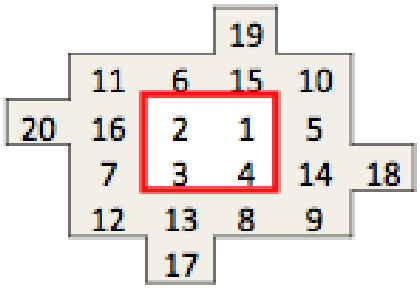


Fig. 4.7.: Highlight core.

The dot profiles are generated following the stacking constraint. The index matrix for highlights can be generated using the dot profile.

$$d^h[\mathbf{m}] = \begin{cases} N_h L_M - \sum_{i-1}^{N_h L_M} p[\mathbf{m}; \frac{i}{L-1}] & \text{if } \mathbf{m} \in \Omega_h \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

where Ω_h is the highlight core region.

The shadow core region, as shown below, can be generated in similar way as the highlight core.

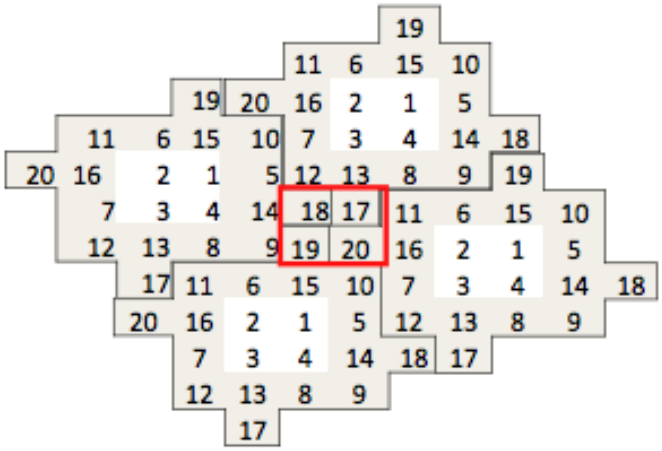


Fig. 4.8.: Shadow core.

Following Lee and Allebach's approach, the shadow core region is given by:

$$d^s[\mathbf{m}] = \begin{cases} L - 1 - d^h[(\mathbf{m} - \mathbf{m}_c) mod [H, W]] & \text{if } \mathbf{m} \in \Omega_s \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

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

4.3.4.2.2.2 Hybrid screen generation

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

$$d^m[\mathbf{m}] = \begin{cases} L_M d_m[\mathbf{m}] + d_M[\mathbf{m}] & \text{if } \mathbf{m} \in \Omega_m \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

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

$$d[\mathbf{m}] = d^h[\mathbf{m}] + d^m[\mathbf{m}] + d^s[\mathbf{m}] \tag{4.14}$$

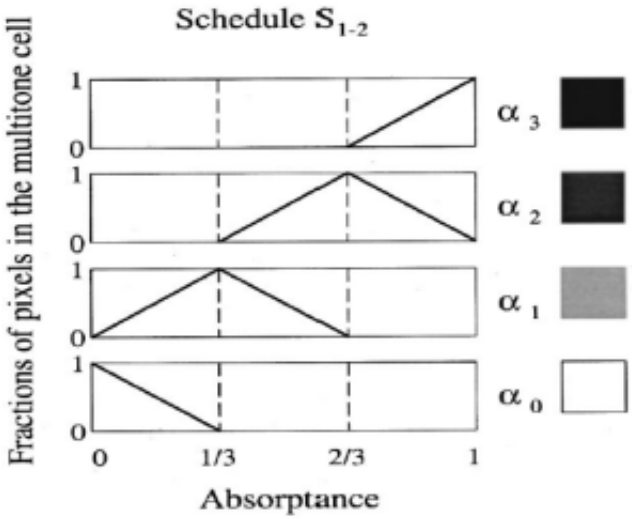
This index matrix is used for our novel halftoning algorithm, which we will discuss in Section 4.4.

4.3.4.2.2.3 Generalize concept of core

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

4.3.5 E. Multilevel halftoning

The traditional multilevel halftoning method [34], shown in Fig. 4.9, is incorporated in our new halftoning algorithm. The idea in the traditional multilevel halftoning method is that we have no drop (absorptance 0), 1 drop (absorptance $\frac{1}{3}$), 2 drops (absorptance $\frac{2}{3}$), and 3 drops (absorptance 1). The total number of levels as a result of multilevel halftoning is $D \cdot N + 1$ without supercell, where D is maximum number of drop per pixel, and N is number of pixels in the microcell.



.

Fig. 4.9.: Schedule S_{1-2} for traditional one-two tone mixture multitone screen [**?**]

4.3.6 F. Display image

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

4.3.7 G. Estimate ink saving

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

4.4 Novel halftoning method

4.4.1 Introduction

After the GCR and ink limitation steps, which serve as preprocessing for the halftoning algorithm, we now discuss the novel halftoning method. The novel idea is to minimize overlap between the C, M, Y, and K colorants with multilevel halftoning and ink limitation. The general idea is shown in Fig. 4.2.

We place black ink and the C, M and Y inks separately, because black overlapping with other colorants reduces the effectiveness of the C, M, and Y colorants. In our method, we also separate the C, M, and Y colorants as much as possible, as shown in Fig. 4.2.

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

1. Minimize the overlap between C,M,Y,K colorants.
2. All colorants follow the traditional 1-2 schedule for multilevel halftoning.
3. We will not print 2 drops on any pixel until every pixel has at least 1 drop. We will not print 3 drops on any pixel until every pixel has at least 2 drops. We will not print 4 drops on any pixel until every pixel has at least 3 drops (Leveling Rule).
4. We will not print anything on 3 drops of K (Ink Efficiency Rule).
5. We cannot print more than 4 drops on any pixel (Ink Limit Rule).
6. Save ink.

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

• Case 1: when $0 \leq \bar{I}_{half}^{total} \leq \frac{1}{3}$, we can print the requested amounts of C, M, Y, K with only 1 drop at each pixel of the halftone cell.

• Case 2: when $\frac{1}{3} < \bar{I}_{half}^{total} \leq \frac{2}{3}$, we can satisfy the request by printing 1 or 2 drops at every pixel in the halftone cell.

• Case 3: when $\frac{2}{3} < \bar{I}_{half}^{total} \leq 1$, we can satisfy the request by printing 2 or 3 drops at every pixel in the halftone cell.

• Case 4: when $1 < \bar{I}_{half}^{total} \leq \frac{4}{3}$, we can satisfy the request by printing 3 or 4 drops at every pixel in the halftone cell.

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

Fig. 4.10.: Halftone growth rules for Cases 1, 2 and 3.

As shown in Fig. 4.10, in Case 1, we place all colorants separate from each other, black grows from the center of K, and cyan grows from the center of the C, M, and Y colorants, followed by magenta and yellow. To determine the exact location of C, M, Y and K, we refer to the hybrid index matrix $d_K[\mathbf{m}]$ as the hybrid screen for K, and create another hybrid screen $d_{CMY}[\mathbf{m}]$, centered at the location of the C, M and Y colorants, which is located 45 relative to the center of CMY. The placement of C, M, Y, K is based on the idea that each colorant is assigned to a set of pixels in the hybrid index matrix according to its relative percentage. For black, we calculate the percentage of K_{half} relative to $\bar{I}_{half}^{total}$ and assign $L \cdot \frac{K_{half}}{\bar{I}_{half}^{total}}$ pixels starting at the center of the black region to the black colorant. For cyan, we assign indices $0, 1, ...L \cdot \frac{C_{half}}{\bar{I}_{half}^{total}} - 1$ in $d_{CMY}[\mathbf{m}]$ to the cyan colorant. For magenta, we assign $L \cdot \frac{M_{half}}{\bar{I}_{half}^{total}}$ pixels starting

at index $L \cdot \frac{C_{half}}{I_{half}^{total}}$ to magenta. For yellow, we assign $L \cdot \frac{Y_{half}}{I_{half}^{total}}$ pixels starting at index $L \cdot \frac{C_{half}+M_{half}}{I_{half}^{total}}$ to yellow.

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

4.4.2 Case by case study

4.4.2.1 Case 1 ($0 \leq I_{half}^{total} \leq \frac{1}{3}$)

We describe the halftone ink placement by providing several examples. First, if there is only one colorant, assuming K here. The ink placement is as illustrated in Fig. 4.11. Starting from the center of K, as ink K grows, more ink is placed in the outer ring. Finally, when ink K reaches $\frac{1}{3}$, the whole plane is covered by one drop of K. The growing sequence is from Fig 4.11 (a) to (c).

(a) (b) (c)

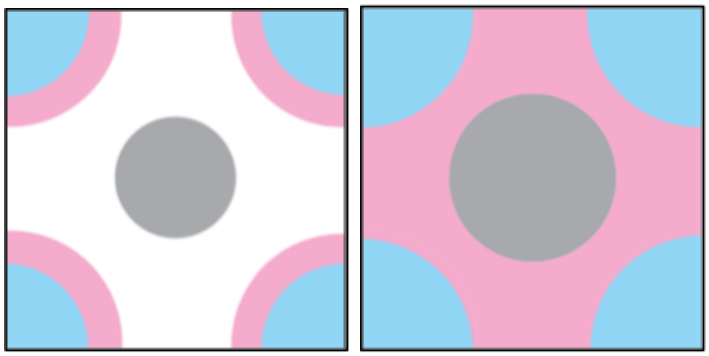
Fig. 4.11.: halftone illustration as black ink grows.

Now, we illustrate when there are two colorants. As shown in Fig. 4.12 (a) through (c), Both K and C grows simultaneously. K start from the center of K and C starts from the center of CMY. As they grows, more black ink is added to the outer ring of existing black ink and more cyan ink is added to the outer ring of existing cyan ink. Finally, Cyan and black cover the whole plain as shown in Fig. 4.12 (c).

(a) (b) (c)

Fig. 4.12.: halftone illustration as black and cyan ink grows.

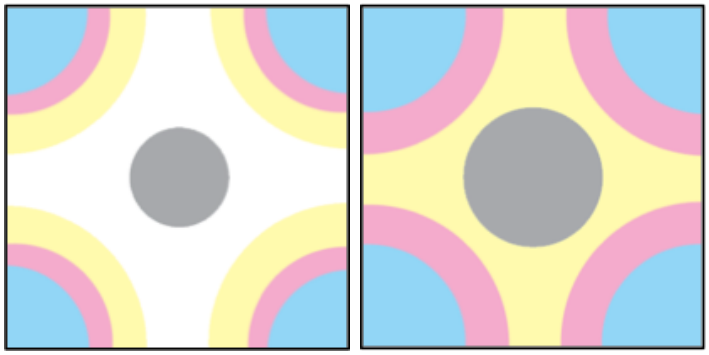
Then, we add magenta into the study. Black and cyan grow from the center of K and CMY respectively as in previous case. Since magenta has a lower priority than cyan, we put magenta on the outer ring of cyan, as shown in Fig. 4.13 (a). When all three colorants add up to $\frac{1}{3}$, we have the relative ink placement as shown in Fig. 4.13 (b).



(a) (b)

Fig. 4.13.: halftone illustration as black, cyan and magenta ink grows.

The last example include all four colorants, similar to the previous example, yellow ink is placed on the outer ring of magenta since it has the lowest priority. The rest remain the same.



(a) (b)

Fig. 4.14.: halftone illustration as black, cyan, magenta and yellow ink grows.

4.4.2.2 Case 2 ($\frac{1}{3} \leq I_{half}^{total} \leq \frac{2}{3}$)

In this case, we can satisfy the request by printing one or row drops at every pixel in the halftone cell. We need to apportion each colorant between layers one and two.

Recall that $I_{half}^{total} = C_{half} + M_{half} + Y_{half} + K_{half}$. For each colorant, we get the amount to put on layer one by the equation below:

$$I_{half}^{layer1} = I_{half} \frac{1/3}{\bar{I}_{half}^{total}} \tag{4.15}$$

Where $I = C, M, Y, K$.

That is, for layer one, each colorant gives up the same proportion of its requested amount.

We know that:

$$_{layer1}I_{half}^{layer1} = C_{half}^{layer1} + M_{half}^{layer1} + Y_{half}^{layer1} + K_{half}^{layer1} = \frac{1}{3} \tag{4.16}$$

Then, we can calculate layer 2 colorant amount:

$$I_{half}^{layer2} = I_{half} - I_{half}^{layer1}, I = C, M, Y, K \tag{4.17}$$

Note that $I_{half}^{layer2} \geq 0$, because $I_{half} \geq I_{half}^{layer1}$. This follow from the fact that $\bar{I}_{half}^{total} \geq \frac{1}{3}$. Therefore, $I_{half}^{layer1} = I_{half} \frac{1/3}{I_{half}^{total}} \leq I_{half} \frac{1/3}{1/3} = I_{half}$, and $I_{half}^{layer2} = I_{half} - I_{half}^{layer1} \geq 0$

4.4.2.2.1 Fitting layer 2 on top of layer 1

In this section, we discuss how to fit ink of layer 2 on top of layer 1. But first, we need to prove that for each colorant, layer 2 will fit on top of layer 1.

According to the design rules:

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

And that $\bar{I}_{half}^{total} \geq \frac{1}{3}$, we have the first layer fully filled, $_{layer1}\bar{I}_{half}^{total} = \frac{1}{3}$. Note that $\frac{1}{3} \leq \bar{I}_{half}^{total} \leq \frac{2}{3}$, which can be rearranged to yield $\frac{1}{2} \leq \frac{1/3}{I_{half}^{total}} \leq 1$, thus we have:

$$\begin{cases} \mathrm{I}_{half}^{layer1} = I_{half} \left(\frac{1/3}{I_{half}^{total}} \right) \geq \frac{I_{half}}{2}, \text{ where } I_{half} = C_{half}, M_{half}, Y_{half}, K_{half} \\ \mathrm{I}_{half}^{layer2} = I_{half} - I_{half}^{layer1} = I_{half} \left(1 - \frac{1/3}{I_{half}^{total}} \right) \leq \frac{I_{half}}{2}, \text{ since } \frac{1/3}{I_{half}^{total}} \geq \frac{1}{2} \end{cases} \tag{4.18}$$

From the above two lines, it follows that:

$$I_{half}^{layer1} \geq I_{half}^{layer2} \text{ and } I_{half}^{layer1} = I_{half}^{layer2} \text{ when } I_{half}^{total} = \frac{2}{3} \tag{4.19}$$

So, for any colorant, layer 2 will fit on top of layer 1 of the same colorant.

Now we know how much ink to put on each layer, we need to discuss where the ink is placed. We know that when $I_{half}^{total} = \frac{2}{3}$ layer 2 is completely covering layer 1, and layer 2 and layer 1 are the same. If $I_{half}^{layer2} < I_{half}^{layer1}, I = C, M, Y$ or K, layer 2 will not completely cover layer 1. In this case, for each colorant, we grow layer 2 in a similar way that we grew layer 1. Cyan grow outward from the center of CMY; Magenta grow from inner circle of magenta (the outer circle of cyan on the first layer) and grow outward; Yellow grow from the inner circle of yellow (the outer circle of magenta on the first layer) and grow outward toward the center of black. Black has a different center than CMY, and we place black ink on the center of K and grow outward towards the outer circle of black, as shown in Fig. 4.10.

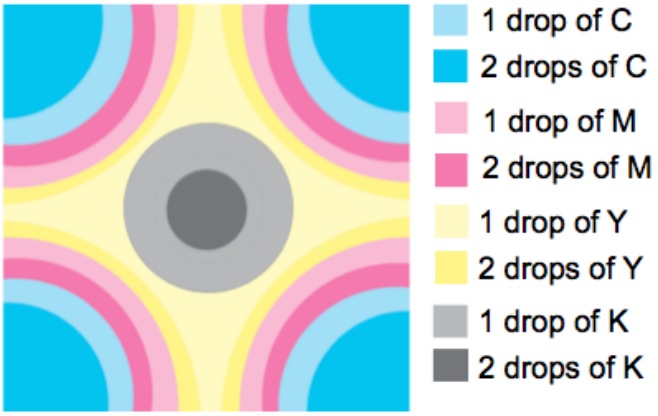


Fig. 4.15.: Case 2 illustration when layer 2 does not completely cover layer 1

4.4.2.3 Case 3 ($\frac{2}{3} \leq \bar{I}_{half}^{total} \leq 1$)

Similar in Case 2, we need to apportion each colorant between layers 1,2 and 3.

In this case, we can satisfy request by printing 2 or 3 drops at every pixel in the halftone cell.

Recall eq. (4.15) through eq. (4.17) in Case 2. For Case 3, we are printing at least 2 drops at every pixel. Thus:

$$\begin{aligned} {}_{layer1}\bar{I}^{total}_{half} +_{layer2} \bar{I}^{total}_{half} &= C^{layer1}_{half} + M^{layer1}_{half} + Y^{layer1}_{half} + K^{layer1}_{half} \\ &+ C^{layer2}_{half} + M^{layer2}_{half} + Y^{layer2}_{half} + K^{layer2}_{half} = \frac{2}{3} \end{aligned} \tag{4.20}$$

And we have $I^{layer1}_{half} = I^{layer2}_{half}$ when $\bar{I}^{total}_{half} = \frac{2}{3}$. In this case,

$$I^{layer1}_{half} = I^{layer2}_{half} = I_{half}\frac{1/3}{I^{total}_{half}} = I_{half}\frac{1/3}{2/3} = \frac{1}{2}I_{half} \tag{4.21}$$

Where $I = C, M, Y, K$.

When $\bar{I}^{total}_{half} > \frac{2}{3}$, both I^{layer1}_{half} and I^{layer2}_{half} remain the same as they are for $\bar{I}^{total}_{half} = \frac{2}{3}$. And $I^{layer3}_{half} = I_{half} - I^{layer1}_{half} - I^{layer2}_{half}, I = C, M, Y, K$

4.4.2.3.1 Fitting layer 3 on top of layer 2

In order to fit layer 3 on top of layers 1 and 2, we need to prove that for each colorant, layer 3 will fit on top of layer 2. Note that:

$$\frac{2}{3} \leq \bar{I}^{total}_{half} \leq 1 \tag{4.22}$$

This can be rearranged to $\frac{1/3}{\bar{I}^{total}_{half}} \geq \frac{1}{3}$, then we have

$$I^{layer1}_{half} = I^{layer2}_{half} = I_{half}\frac{1/3}{I^{total}_{half}} \geq \frac{I_{half}}{3} \tag{4.23}$$

By rearranging Eq. (4.22) again, we have $0 \leq \left(1 - \frac{2/3}{I^{total}_{half}}\right) \leq \frac{1}{3}$ Thus,

$$I^{layer3}_{half} = I_{half} - I^{layer1}_{half} - I^{layer2}_{half} = I_{half}\left(1 - \frac{2/3}{I^{total}_{half}}\right) \leq \frac{I_{half}}{3} \tag{4.24}$$

We have shown that $I^{layer1}_{half} \geq \frac{I_{half}}{3}$ and $I^{layer3}_{half} \leq \frac{I_{half}}{3}$. Thus, $I^{layer2}_{half} \geq I^{layer3}_{half}$. So, for any colorant, layer 3 will fit on top of layer 2 of the same colorant.

And when layer 3 completely cover layer 2, it can be shown that $I^{layer2}_{half} = I^{layer3}_{half}$ when $I^{total}_{half} = 1, I^{layer1}_{half} = I^{layer2}_{half} = I_{half}\frac{1/3}{1} = \frac{1}{3} \cdot I_{half}$.

When layer 3 does not completely cover layer 2 ($I^{layer3}_{half} < I^{layer2}_{half}$), the ink placement is shown in Fig 4.6 (c). In this case, we grow layer 3 in the same way that we grew layer 2.

4.4.2.4 Case 4 ($1 \leq I_{half}^{total} \leq \frac{4}{3}$)

Recall that we are not allowed to put 4 drops of any single colorant on the same pixel. In this case, we can satisfy request by printing 3 or 4 drops at every pixel in the halftone cell.

Similar to the apportioning in Case 3, in Case 4, we are printing at least 3 drops at every pixel. Thus:

$$\begin{aligned} {}_{layer1}\bar{I}_{half}^{total} +_{layer2} \bar{I}_{half}^{total} +_{layer3} \bar{I}_{half}^{total} = {} & C_{half}^{layer1} + M_{half}^{layer1} + Y_{half}^{layer1} + K_{half}^{layer1} \\ & + C_{half}^{layer2} + M_{half}^{layer2} + Y_{half}^{layer2} + K_{half}^{layer2} \\ & + C_{half}^{layer3} + M_{half}^{layer3} + Y_{half}^{layer3} + K_{half}^{layer3} = 1 \end{aligned} \tag{4.25}$$

From Case 3, we know that layer 1, 2 and 3 are equal for each colorant when $\bar{I}_{half}^{total} = 1$. For Case 4, the ink amount for the first 3 layers remain the same, and it can be shown that:

$$\begin{cases} I_{half}^{layer1} = I_{half}^{layer2} = I_{half}^{layer3} = I_{half} \frac{1/3}{\bar{I}_{half}^{total}} \\ I_{half}^{layer4} = I_{half} - I_{half}^{layer1} - I_{half}^{layer2} - I_{half}^{layer3} \end{cases} \tag{4.26}$$

Where $I = C, M, Y, K$.

4.4.2.4.1 Fit layer 4 on top of layer 3

Since we do not allow more than three drops of the same colorant to print on the same pixel, we need to put the colorants on locations other than itself for layer 4.

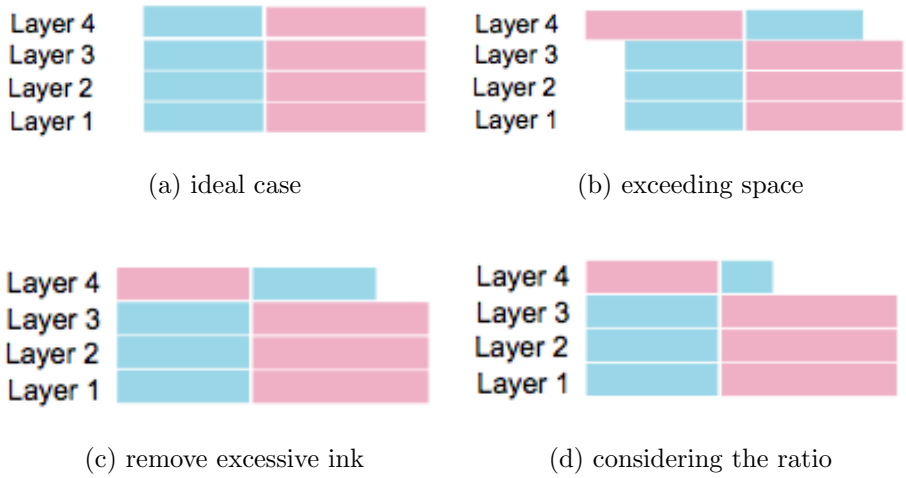
Staring with K, we find locations other than K, i.e. put K on C, M or Y. Then, we put C on locations other than C, i.e. put C on K, M or Y. Next, we put M on locations other than M, i.e. put M on K, C or Y. Finally, we put Y on locations other than Y, i.e. put C on K, C or M.

In order to fit layer 4 on top of layer 1, 2 and 3 for Case 4, we put colorant I on J, where $I \neq J$, and we also need to maintain the ratio between different colorants.

However, directly applying the colorants on layer 4 cannot guarantee that all colorants has the space needed, thus the ratio of different colorants cannot stay the same as requested. We illustrate how we calculate the amount of colorants on layer 4 using an example:

Consider a simple case where there are only C and M. We know that when $C_{half} \neq M_{half}$, we cannot guarantee that there will always be available space for the colorant exceeding the space available on layer 4.

Suppose $C_{half} = \frac{1}{3}, M_{half} = 1$. Then ideally, we would like to put 4 layers of C on $\frac{1}{4}$ of the halftone cell, and 4 layers of M on $\frac{3}{4}$ of the halftone cell, as shown below:



(a) ideal case (b) exceeding space

(c) remove excessive ink (d) considering the ratio

Fig. 4.16.: Fitting C and M on layer 4 example.

But we cannot print 4 layers of the same colorant on any pixel. So we swap the 4^{th} layer shown above to put M on top of C and C on top of M, as shown in Fig 4.11 (b). On the left side of the halftone cell, M takes more space than C. So, we need to reduce the amount of M as shown in Fig 4.11(c). Then we need to take the ratio of C and M into account. Since we removed the excessive part of M, we also need to reduce C proportionally.

Now, consider putting C and K on layer 4. Similar to the previous example of C and M, we put C on top of 3 drops of K, and K on 3 drops of C, as shown in

Fig 4.12(a). Since we do not allow colorants to be put on 3 drops of K, in order to maintain the ratio of C and K, we can not put K on 3 drops of C either. As shown in Fig 4.12(b).

(a) Considering the ratio of C and K (b) No colorant on 3 drops of K

Fig. 4.17.: Fitting C and K on layer 4 example.

This leads to another constraint for the 4^{th} layer. Since we do not allow C, M, Y colorants to be put of 3 layers of K, the total ink amount of C, M, Y, K we can put on top of C, M, Y is:

$$Total_{base}^{CMY} = C_{base} + M_{base} + Y_{base} \tag{4.27}$$

where I_{base} is the ink amount on the 3^{rd} layer for colorant I, where $I = C, M, Y$.

We calculate the amount of K on top of C, M and Y by the ratio of C, M and Y:

$$\begin{cases} K_{on\,C} = K_{layer4} \frac{C_{base}}{Total_{base}^{CMY}} \\ K_{on\,M} = K_{layer4} \frac{M_{base}}{Total_{base}^{CMY}} \\ K_{on\,Y} = K_{layer4} \frac{Y_{base}}{Total_{base}^{CMY}} \end{cases} \tag{4.28}$$

Next, we calculate the amount of C on top of M and Y by the ratio of M and Y (Here, $Total_{base}^{MY} = M_{base} + Y_{base}$):

$$\begin{cases} C_{on\,M} = C_{layer4} \frac{M_{base}}{Total_{base}^{CMY}} \\ C_{on\,Y} = C_{layer4} \frac{Y_{base}}{Total_{base}^{CMY}} \end{cases} \tag{4.29}$$

Similarly, we calculate the amount of M on top of C and Y b the ratio of C and Y (Here, $Total_{base}^{CY} = C_{base} + Y_{base}$):

$$\begin{cases} M_{on\,C} = M_{layer4} \frac{C_{base}}{Total_{base}^{CMY}} \\ M_{on\,Y} = M_{layer4} \frac{Y_{base}}{Total_{base}^{CMY}} \end{cases} \tag{4.30}$$

Finally, we calculate the amount of Y on top of C and M by the ratio of C and M (Here, $Total_{base}^{CM} = C_{base} + M_{base}$):

$$\begin{cases} Y_{on\,C} = Y_{layer4} \frac{C_{base}}{Total_{base}^{CMY}} \\ Y_{on\,Y} = Y_{layer4} \frac{M_{base}}{Total_{base}^{CMY}} \end{cases} \tag{4.31}$$

Now, we can calculate the total ink we printed on C, M and Y respectively on the 3^{rd} layer:

$$\begin{cases} KMY_{on\,C} = K_{on\,C} + M_{on\,C} + Y_{on\,C} \\ KCY_{on\,M} = K_{on\,M} + C_{on\,M} + Y_{on\,M} \\ KCM_{on\,Y} = K_{on\,Y} + C_{on\,Y} + M_{on\,Y} \end{cases} \tag{4.32}$$

Then, we can get the amount of ink that is not able to be put on C, M and Y on the 3^{rd} layer:

$$\begin{cases} KMY_{on\,C}^{exceed} = KMY_{on\,C} - C_{base} \\ KCY_{on\,M}^{exceed} = KCY_{on\,M} - M_{base} \\ KCM_{on\,Y}^{exceed} = KCM_{on\,Y} - Y_{base} \end{cases} \tag{4.33}$$

Next, we calculate the maximum percentage of exceeding colorants:

$${}_{pct}KMY_{on\,C}^{exceed} = \begin{cases} KMY_{on\,C}^{exceed} / CMY_{on\,C} & , \text{if } KMY_{on\,C} \neq 0 \\ 0 & , else \end{cases} \tag{4.34}$$

$$ {}_{pct}KCY^{exceed}_{on\,M} = \begin{cases} KCY^{exceed}_{on\,M}/KCY_{on\,M} & \text{, if } KCY_{on\,M} \neq 0 \\ 0 & \text{, } else \end{cases} \tag{4.35} $$

$$ {}_{pct}KCM^{exceed}_{on\,Y} = \begin{cases} KCM^{exceed}_{on\,Y}/KCM_{on\,Y} & \text{, if } KCM_{on\,Y} \neq 0 \\ 0 & \text{, } else \end{cases} \tag{4.36} $$

We choose the maximum of the three percentages to apply on all colorants:

$$ Exceed^{pct}_{max} = max\left({}_{pct}KMY^{exceed}_{on\,C}, {}_{pct}KCY^{exceed}_{on\,M}, {}_{pct}KCM^{exceed}_{on\,Y}\right) $$

And when $Exceed^{pct}_{max} \neq 0$, the amount of ink we retain is:

$$ Amount^{pct}_{retain} = \begin{cases} C_{base}/CMY_{on\,C} & \text{, if } Exceed^{pct}_{max} =_{pct} KMY^{exceed}_{on\,C} \\ M_{base}/KCY_{on\,M} & \text{, if } Exceed^{pct}_{max} =_{pct} KCY^{exceed}_{on\,M} \\ Y_{base}/KCM_{on\,Y} & \text{, if } Exceed^{pct}_{max} =_{pct} KCM^{exceed}_{on\,Y} \\ 0 & \text{, if } K_{on\,I} \neq 0 \text{ and } I_{layer4} \neq 0 \text{ and } J = 0 \end{cases} \tag{4.37} $$

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

And when $Exceed^{pct}_{max} = 0$, we have $Amount^{pct}_{retain} = 1$. We multiply each colorant on the 4^{th} layer by $Amount^{pct}_{retain}$ to get the actual ink amount we put on the 4^{th} layer. So the actual amount of colorants we put on the 4^{th} layer is:

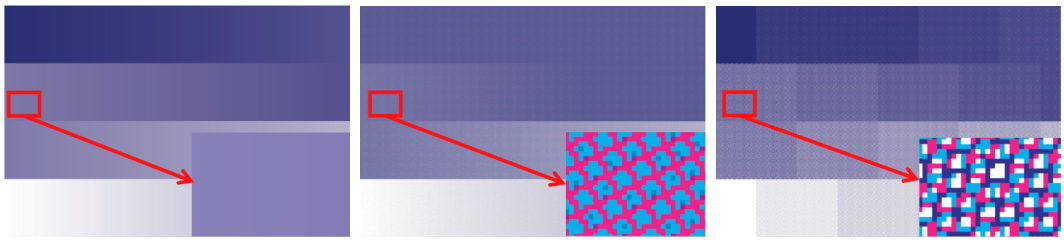
$$ I^{act}_{layer4} = I^{org}_{layer4} \cdot Amount^{pct}_{remain} \tag{4.38} $$

Where $I = C, M, Y, K$.

4.5 Experimental results

We compare our halftoning algorithm result with a conventional clustered-dot channel-independent halftoning method [35] [36] result using a comparable screen frequency. We first compare halftoned images of a ramp that consists of cyan and

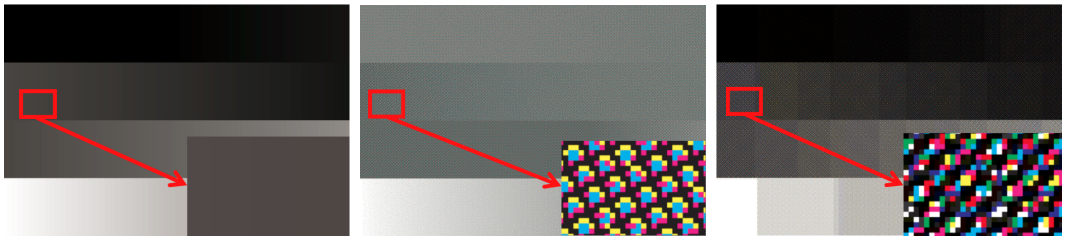
magenta equally increasing from absorptance 0 to 1. The comparison between the novel and traditional halftoning algorithms is shown below:



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

Fig. 4.18.: Comparison of ramp images with C and M equally increasing.

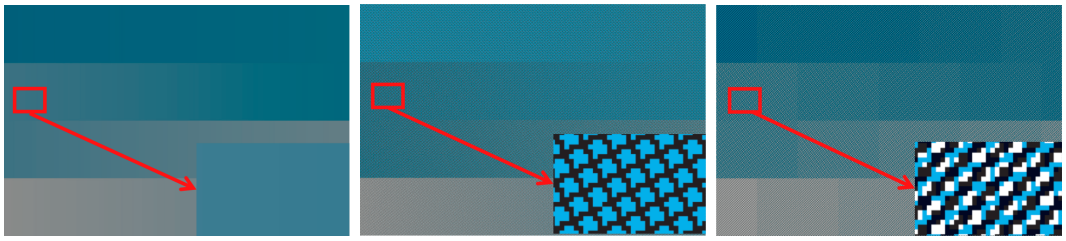
Next, we show the halftoned images of a ramp with C, M, Y, K equally increasing from absorptance 0 to 1.



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

Fig. 4.19.: Comparison of ramp images with K, C, M, Y equally increasing.

Then, we show the halftoned images of a ramp with K fixed (absorptance is 0.5), while C increases from absorptance 0 to 1.



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

Fig. 4.20.: Comparison of ramp images with K fixed, C increasing.

The ink savings for the ramps shown in Figs. (4.18-4.20) is listed in Table. 1.

Table 1: Ink savings with novel halftoning algorithm compared to the traditional algorithm for the ramps shown in Figs. 4.18-4.20.

Percentage of ink saving (%)				
Fig.	C	M	Y	K
4.18	10.48	10.51	0	0
4.19	66.52	66.48	66.03	32.48
4.20	15.07	0	0	8.47

Note that the traditional halftone images shown in Fig.(4.18) - Fig.(4.20) are not based on a supercell. So they exhibit contouring artifacts due to quantization. Also, the novel halftone ramps have not been color managed. As a consequence of the rules described in Sec. 4.2.1, these ramps are not color matched to the continuous-tone images. At this point, we have not investigated what is the gamut for our novel halftoning algorithm. If it is deficient in some aspects, it may be necessary to modify the design rules described in Sec. 4.2.1.

4.6 Conclusions

We have introduced a novel single-angle periodic, clustered-dot halftoning method that eliminates the traditional rosette artifacts due to overlapping halftone patterns

for each colorant. Colorant overlap is minimized in order to improve ink efficiency. Our method supports multi-drop halftoning. We show significant potential ink savings.

5. SUMMARY

In this dissertation, we discussed three projects related to digital color halftoning: Color halftoning based on Neugebauer Primary Area Coverage, fast Parawacs screen halftone method and novel color halftoning for ink savings.

All of the color halftoning methods involves color management. When the color space or Neugebauer Primary changes, a new lookup table needs to generated for mapping the color between original image and the printer color space. Once this step is done, the halftoning algorithm that is based on Neugebauer Primaries are very scalable, they can be extended to any number of Neugebauer Primaries. The novel color halftoning method do not allow scaling to more colorants as easily, it is under more constraints, such as the lpi which determines the DPHC cell size. No colorants are allowed to overlap with 3 drops of black is reasonable and reduce the cost for printing. The ink limit also reduce the amount of ink greatly, and it also makes the image look much lighter than the original image, the color management part needs to be improved for this method, especially taking into account of the halftoning algorithm with the GCR and ink limitation, which drastically change the gamut.

The iterative method takes more time, but currently it yield the best result in terms of smoothness and texture. And also the algorithm can be parallelized, making it fast enough for fast printing. While parawacs screen allows fast halftone with or without parallelism with little compromise on the halftone quality. The novel halftoning method is highly parallelizable as well, with a given CMYK input, the halftone result can be stored in a table, allowing fast access even without computation.

The Neugebauer Primary and parawacs based methods are dispersed dot halftoning method, they can give detailed textures, while the novel halftoning method is designed for large format printing, it is periodic with fixed lpi, so it cannot provide

the detail as dispersed dot methods, but the larger clustered dot make this method more stable and less prone to variations in the printing process. Multilevel halftoning helps to reduce the visibility in the highlight and shadow regions and creates more graylevels at the same time. Hybrid screen helps to generate more graylevels and improve the texture as well.

REFERENCES

REFERENCES

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

[2] G.-Y. Lin and J. P. Allebach, "Generating stochastic dispersed and periodic clustered textures using a composite hybrid screen," *IEEE Trans. Image Processing*, vol. 15(12), December 2006.

[3] , "Multilevel screen design using direct binary search," *J. Opt. Soc. Am. A*, vol. 19(10), October 2002.

[4] , "Generating stochastic dispersed and periodic clustered textures using a composite hybrid screen," *Proc. SPIE, Color Imaging IX: Processing, Hardcopy, and Applications*, vol. 5293, 2004.

[5] Q. Lin and J. P. Allebach, "Color fm screen design using dbs algorithm," *Proc. SPIE, Color Imaging:Device Independent Color, Color Hardcopy, and Graphic Arts III*, vol. 3300, 1998.

[6] P. Li and J. P. Allebach, "Look-up-table based halftoning algorithm," *IEEE Trans. Image Processing*, vol. 9(9), September 2000.

[7] , "Tone-dependent error diffusion," *IEEE Trans. Image Processing*, vol. 13(2), 2004.

[8] T. N. Pappas, J. P. Allebach, and D. L. Neuhoff, "Model-based digital halftoning," *IEEE Signal Processing Magazine*, vol. 20(4), July 2003.

[9] J.-H. Lee and J. P. Allebach, "Cmyk halftoning algorithm based on direct binary search," *Journal of Electronic Imaging*, vol. 11(4), p. 517 527, Oct. 2002.

[10] A. U. Agar and J. P. Allebach, "Model-based color halftoning using direct binary search," *IEEE Transactions on Image Processing*, vol. 14(12), pp. 1945 – 1959, 2005.

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

[12] R. Nsnen, "Visibility of halftone dot textures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-14, no. 6, pp. 920–924, Nov 1984.

[13] B. Kolpatzik and C. Bouman, "Optimized error diffusion for image display," *Journal of Electronic Imaging*, vol. 1, no. 3, pp. 277 – 92, 1992/07/. [Online]. Available: http://dx.doi.org/10.1117/12.60027

[14] K. T. Mullen, "The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings," *The Journal of Physiology*, vol. 359(1), p. 381400, 1985.

[15] X. Zhang and B. A. Wandell, "A spatial extension of cielab for digital color image reproduction," *Journal of the Society for Information Display*, vol. 5(1), pp. 61–63, 1997.

[16] B. Wandell, *Foundations of Vision*. Sinauer Associates, 1995. [Online]. Available: https://books.google.com/books?id=dVRRAAAAMAAJ

[17] S. H. Kim and J. P. Allebach, "Impact of hvs models on model-based halftoning," *IEEE Transactions on Image Processing*, vol. 11(3), pp. 258–269, 2002.

[18] J. Wanling and J. P. Allebach, "Color halftoning based on neugebauer primary area coverage," *Proc. of IS&T Symposium of Electronic Imaging, Color Imaging XXII*, pp. 91–102(12), Jan. 2017.

[19] J. Morovič, P. Morovič, and J. Arnabat, "Hans: Controlling ink-jet print attributes via neugebauer primary area coverages," *IEEE Transactions on Image Processing*, vol. 21(2), pp. 688–696, 2012.

[20] J. P. Allebach, "Dbs: retrospective and future directions," *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI*, vol. 4300, pp. 358–376, 2001.

[21] D. J. Lieberman and J. P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Transactions on Image Processing*, vol. 9(11), pp. 1950–1963, 2000.

[22] Ulichney, *Digital Halftoning*. Cambridge, MA,USA: MIT Press, 1987.

[23] Y. Zhang, J. L. Recker, R. Ulichney, and J. Owens, "Plane-dependent error diffusion on a gpu," *SPIE Proceedings*, vol. 8295:8295b-59, pp. 1–10, 2012.

[24] J. Morovič, *Basic Computational Geometry for Gamut Mapping*, *1*st ed. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: Wiley, 2008.

[25] U. Sarkar, *HANS and PARAWACS*, 2015, hP-Barcelona.

[26] P. Morovič, J. Morovič, J. Gondek, M. Gaubatz, and R. Ulichney, "Parawacs: color halftoning with a single selector matrix," Nov. 2016.

[27] Kacker and J. P. Allebach, "Joint halftoning and watermarking," *IEEE Trans. Signal Processing*, vol. 51(4), p. 1054 1068, Apr. 2003.

[28] P. Chiang, N. Khanna, A. Mikkilineni, M. Segovia, S. Suh, J. Allebach, G. Chiu, and E. Delp, "Printer and scanner forensics," *IEEE Signal Processing Magazine*, Mar. 2009.

[29] Q. Wang, X. F. Zhou, and Q. Yi, "Research on parameters and printing reproduction characteristics of concentric screening," *Applied Mechanics and Materials*, vol. 262, pp. 334–339, 2013.

[30] N. Thi Hai van, N. Tat Thang, J. Di, and M. Guo, "Predicting color change in wood during heat treatment using an artificial neural network model," *Bioresources*, vol. 13, pp. 6250–6264, 06 2018.

[31] W. contributors, "Grey component replacement — Wikipedia, the free encyclopedia," 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Grey_component_replacementoldid=785274671

[32] D. Donevski, A. Poljicak, and M. S. Kurecic, "Colorimetrically accurate gray component replacement using the additive model," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 40–49, 2017, exported from https://app.dimensions.ai on 2019/02/23. [Online]. Available: https://app.dimensions.ai/details/publication/pub.1008230147

[33] C. Lee and J. P. Allebach, "The hybrid screenimproving the breed," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 435–450, Feb 2010.

[34] G.-Y. Lin and J. P Allebach, "Multilevel screen design using direct binary search," *Journal of the Optical Society of America. A, Optics, image science, and vision*, vol. 19, pp. 1969–82, 11 2002.

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

[36] X. Zhang, A. Veis, R. Ulichney, and J. P. Allebach, "Multilevel halftone screen design: Keeping texture or keeping smoothness?" in *2012 19th IEEE International Conference on Image Processing*, Sep. 2012, pp. 829–832.

APPENDIX

A. APPENDIX

Here, we describe the original work of DBS [8] where the NPAC-DBS and PARAWACS screen design is based on.

A.1 DBS

A.1.1 Perceived Halftone Image

Direct binary search compare the perceived original image with the perceived halftone image. We can get the perceived image by convolving an image using a HVS point spread function. Here, we also use $[\mathbf{m}] = [m, n]^T$ to denote discrete coordinates and use $(\mathbf{x}) = (x, y)^T$ to denote continuous spatial coordinates. In addition, $(\mathbf{u}) = (u, v)^T$ is used to denote spatial frequency coordinates.

Let the rendered halftone image be $g_r(\mathbf{x})$, the HVS point spread function (PSF) be $p_{hvs}(\mathbf{x})$, then the perceived halftone image $\tilde{g}(\mathbf{x})$ can be calculated:

$$\tilde{g}(\mathbf{x}) = g_r(\mathbf{x}) * p_{hvs}(\mathbf{x}) \tag{A.1}$$

Next, we look at the printer model. This model give us the rendered halftone image $g_r(\mathbf{x})$ from the halftone image in discrete coordinates. The interaction between neighboring dots is assumed to be additive.

$$g_r(\mathbf{x}) = \sum_m g[\mathbf{m}] p_{dot}\left(\mathbf{x} - \mathbf{X}\mathbf{m}\right) \tag{A.2}$$

Where $\mathbf{X}$ is the periodicity matrix whose column comprise a basis for the lattice of printer addressable dots.

The perceived halftone can be equivalently expressed as:

$$\begin{aligned} \tilde{g}(\mathbf{x}) &= g_r(\mathbf{x}) * \tilde{p}\left(\mathbf{x} - \mathbf{X}\mathbf{m}\right) \\ \tilde{p}(\mathbf{x}) &= p_{hvs}(\mathbf{x}) * p_{dot}(\mathbf{x}) \end{aligned} \tag{A.3}$$

For high resolution printers:

$$\tilde{p}(\mathbf{x}) \approx p_{hvs}(\mathbf{x}) \tag{A.4}$$

A.1.2 HVS Model and Scale Parameter

Nasanen's HVS model is used in the original DBS, because it gives the best result for DBS. We have described Nasanen's HVS model in NPAC-DBS, we rewrite here for convenience:

$$H(\mathbf{u}) = a\Gamma^b exp(-\frac{|\mathbf{u}|}{\lfloor c\ln\Gamma + d\rfloor}) \tag{A.5}$$

Were $a = 131.6$, $b = 0.3188$, $c = 0.525$, $d = 3.91$, and Γ is the average luminance of the light reflected from the print in cd/m^2, usually set to 11, and $\mathbf{u}$ is the spatial frequency coordinates in cycles/degree subtended at the retina.

The PSF $\tilde{p}(\bar{\mathbf{x}})$ or $h(\bar{\mathbf{x}})$ is the inverse Fourier transform of $H(\mathbf{u})$:

$$h(\bar{\mathbf{x}}) = F^{-1}H(\mathbf{u}) \tag{A.6}$$

where $\bar{\mathbf{x}}$ is the angular degree of a length $\mathbf{x}$ in inches when viewed at a distance D inches. They have the relationship:

$$|\bar{\mathbf{x}}| = \frac{180}{\pi} arctan\left(\frac{|\mathbf{x}|}{D}\right) \approx \frac{180|\mathbf{x}|}{\pi D}, \text{ for } \mathbf{x} \ll D \tag{A.7}$$

Thus, we have:

$$\tilde{p}(\mathbf{x}) = h(\mathbf{x}) = \frac{180^2}{(\pi D)^2} h\left(\frac{180}{\pi D}\mathbf{x}\right) = \frac{180^2}{(\pi D)^2} F^{-1}H(\mathbf{u}) \tag{A.8}$$

A.1.3 Error Metric

The error metric of DBS is calculated as follows:

1. Get the error between the original image $f[\mathbf{m}]$ and the halftone image $g[\mathbf{m}]$:

$$e[\mathbf{m}] = g[\mathbf{m}] - f[\mathbf{m}] \tag{A.9}$$

2. Obtain the perceived error using the error image and the HVS model:

$$\tilde{e}(\mathbf{x}) = \sum_{\mathbf{m}} e[\mathbf{m}]\tilde{p}(\mathbf{x} - \mathbf{X}\mathbf{m}) \tag{A.10}$$

3. The error metric can be calculated using step 2 result:

$$E = \int |\tilde{e}(\mathbf{x})|^2 d\mathbf{x} \tag{A.11}$$

A.1.4 Evaluation of A Trial Swap and A Trial Toggle

For a trial swap, the change of error is:

$$\Delta E = (a_0^2 + a_1^2)c_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2a_0 c_{\tilde{p}\tilde{e}}[\mathbf{m}_0] + 2a_1 c_{\tilde{p}\tilde{e}}[\mathbf{m}_1] + 2a_0 a_1 c_{\tilde{p}\tilde{p}}[\mathbf{m}_1 - \mathbf{m}_0] \tag{A.12}$$

where $a_0 = 1$ when $g[\mathbf{m}_0]$ is changed from 0 to 1, and $a_0 = 0$ when $g[\mathbf{m}_0]$ is changed from 1 to 0. Since we only swap pixels different from one another, we have $a_0 = -a_1$.

For a trial toggle, the change of error is:

$$\Delta E = a_0^2 c_{\tilde{p}\tilde{p}}[\mathbf{0}] + 2a_0 c_{\tilde{p}\tilde{e}}[\mathbf{m}_0] \tag{A.13}$$

where a_0 value is the same as in a trial swap. We evaluate all the changes in a certain neighborhood, for example, a 4×5 neighborhood, and we accept the change of a trial swap or a trial toggle if the ΔE can bring the Error metric down the most.

VITA

VITA

Wanling Jiang obtained my BSEE degree from Dalian University of Technology, Liaoning, China in 2011. she obtained MSEE degree in 2013 from University of Southern California, Los Angeles, CA. She is now pursuing her PhD degree in Electrical and Computer Engineering at Purdue University, West Lafayette, IN. Her research interests includes image quality, color science and halftoning.