NPAC FM COLOR HALFTONING FOR THE INDIGO PRESS: CHALLENGES AND SOLUTIONS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Jiayin Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

December 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jan P. Allebach, Chair

School of Electrical and Computer Engineering

Dr. Amy R. Reibman

School of Electrical and Computer Engineering

Dr. Shreyas Sundaram

School of Electrical and Computer Engineering

Approved by:

Dr. Pedro Irazoqui

Head of the School Graduate Program

To my parents Tianyun Liu and Yuanyuan Li.

ACKNOWLEDGMENTS

I would like to thank my major advisor Professor Jan P. Allebach to giving me the opportunity to work with him. Without his tremendous help, advice, and especially encouragement, this research would not have happened.

I also would like to thank our research sponsors, the Hewlett-Packard Company, which gives us generous support, without whom the research would not be possible.

I would like to thank all EISL members for their wonderful collaboration. You support me greatly and were always willing to help me.

Lastly, I would also like to thank my parents for their love and support.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

NP	Neugebauer Primary
NPAC	Neugebauer Primary Area Coverage
DBS	Direct Binary Search
PARAWACS	Parallel Random Weighted Area Coverage Selection
W	White
Y	Yellow
C	Cyan
CY	Green
M	Magenta
MY	Red
CM	Blue
CMY	Black
R	Red
G	Green
B	Blue

ABSTRACT

Liu, Jiayin M.S.E.C.E., Purdue University, December 2018. NPAC FM Color Halftoning for the Indigo Press: Challenges and Solutions. Major Professor: Jan P. Allebach.

FM halftoning is increasingly popular with traditional analog offset lithographic printing processes. There is a desire to offer this capability with digital presses based on electrophotographic printing (EP) technologies. However, the inherent instability of the EP process challenges the achievement of satisfactory print quality with dispersed-dot, aperiodic halftoning. The direct binary search (DBS) algorithm is widely considered to represent the gold standard of dispersed-dot, aperiodic halftone image quality. In this paper, we continue our previous efforts to adapt DBS to use with the Indigo liquid EP printing technology. We describe a complete color management pipeline for halftoning with a PARAWACS matrix designed using DBS. For the first time, we show actual printed patches obtained using our process. Our gamut mapping is performed in the YyCxCz color space, and is image-dependent. It incorporates several stages of alignment between the input and output spaces, as well as several stages of compression. After the gamut mapping, we tessellate the output color space into six global tetrahedra that each share the neutral axis, as an edge. Then, we determine the Neugebauer Primary Area Coverage (NPAC) for each pixel in the image to be printed by tetrahedral interpolation from the four nearest neighbors in the inverse printer mapping table. These four nearest neighbors are chosen so that only four Neugebauer primaries are used to render each pixel.

1. INTRODUCTION

Halftoning is the process of rendering a pattern with a limited number of tone levels. Because human visual system (HVS) acts like a low-pass filter, halftoning looks like a continuous-tone image at certain viewing distance.

Halftoning algorithm can be classified into dispersed-dot textures and clustered-dot textures. Dispersed-dot textures can render isolated dots and clustered-dot textures can render cluster dots. Halftoning algorithm also can be classified into Frequency modulation (FM) and Amplitude modulation (AM). Frequency modulation is to change the density of dots: aperiodic dot spacing but dot size is fixed. Amplitude modulation is to generate a regular grid of dots: dot size varies but periodic dot spacing.

Now, there are two dominant printing technologies, one is Electrophotographic (EP) process with laser and the other is Inkjet (IJ). Clustered dot textures have been use widely in electrophotographic printers and disperse dot textures haven been use widely in inkjet printers.

There are three basic architectures for halftoning algorithms: screening, error diffusion and search-based methods. Search-based methods usually are iterative and find the best halftone image by minimizing perceptive error between continuous-tone and halftone image. Direct binary search (DBS) algorithm [2] first computes mean-squared error (MSE) between filtered continuous-tone image and filtered initial halftone image. By scanning pixel by pixel in the halftone image and applying toggling and swapping to minimize MSE, algorithm stops when MSE cannot be reduced.

Recently, FM halftoning (aperiodic and disperse dots) is increasingly popular with traditional analog offset lithographic printing processes. There is a desire to offer this capability with digital presses based on electrophotographic printing (EP) technologies. However, the inherent instability of the EP process challenges the achievement of satisfactory print quality with dispersed-dot, aperiodic halftoning. The direct binary search (DBS) algorithm is widely considered to represent the gold standard of dispersed-dot, aperiodic halftone image quality.

HP Indigo's Enhanced Productivity Mode (EPM) can provide faster throughput and make productivity boost by eliminating black ink from the production process: using only three colorants rather than four colorants, Cyan (C), Magenta (M) and Yellow (Y). This mode makes total of 8 colors in the printing process, including White (W), Cyan (C), Magenta (M) and Yellow (Y), Cyan and Magenta (CM),Magenta and Yellow (MY), Cyan and Yellow (CY) and Cyan, Magenta and Yellow (CMY) which is Black (k).

In this thesis, we proposed a new color management pipeline for HP Indigo press using FM halftoning based on EP technologies.

In Chapter 2, for the first time, we show actual printed patches, forward and inverse mappings are introduced and generated in order to describe HP Indigo press printer. We also show test pages that we designed and we received actual test pages from Boise, ID USA and measured them using X-Rite DTP 70. In Chapter 3, our partially new development of Gamut mapping are introduced. The novelty part of this Gamut mapping including soft compress lightness first, then compress lightness and chroma at same time to perform center compression. The details in the image was largely preserved. In Chapter 4, based on Gamut mapping and inverse mapping in chapter 3 and chapter 2, we generate PARAWACS halftone image for the first time and procedures of PARAWACS halftone is presented. The overall result quality is good compared with original image. A discussion of future work and conclusion are in chapter 5.

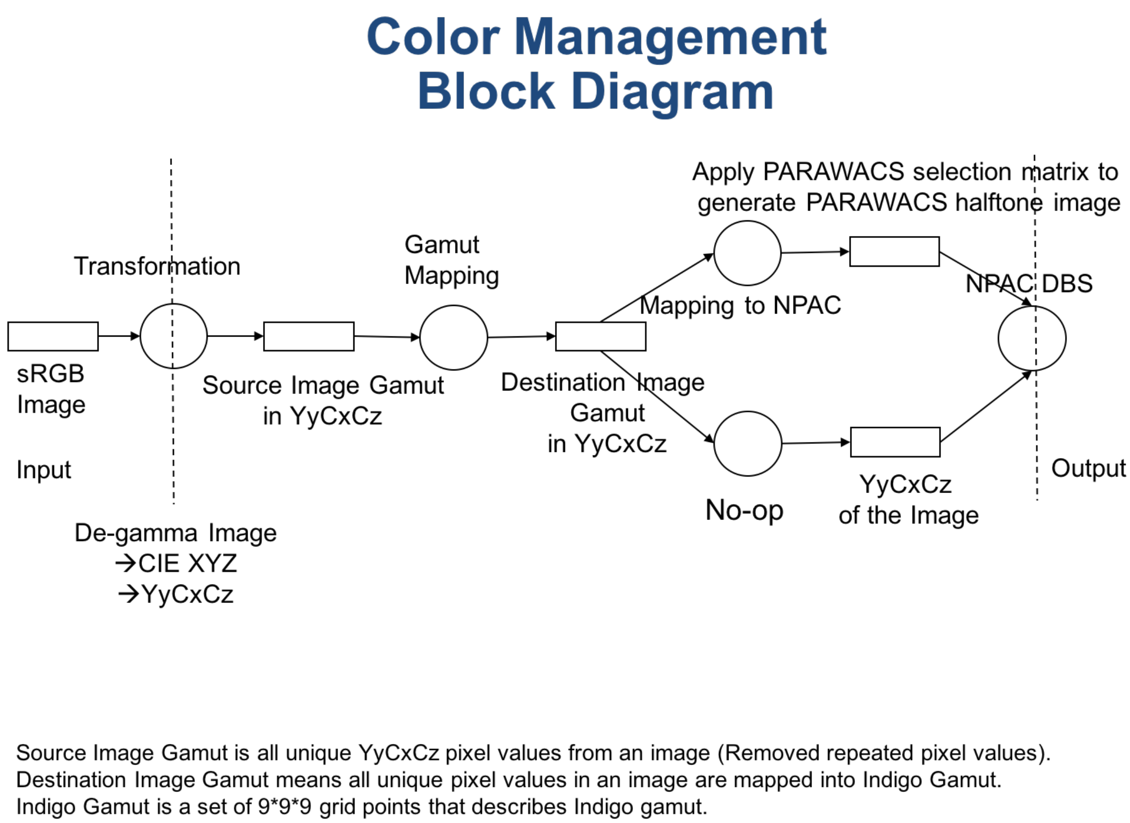


Fig. 1.1. Color Management Block Diagram

2. CHARACTERIZATION AND DEVELOPMENT OF FORWARD AND INVERSE MAPPINGS BASED ON INDIGO 7000 SERIES PRINTS

2.1 Introduction

To develop our color management framework, we want to generate a forward mapping that defines Indigo press printer gamut and given a CMY (Cyan, Magenta and Yellow) combination, find its associate NPAC (Neugebauer Primary Area Coverage) value and YyCxCz value. And a inverse mapping that given a YyCxCz value, find its NPAC value that when printed will yield the desired YyCxCz value. In Indigo press, NPAC is a 8-tuple entry that indicates how much of each NP in percentage when printed the desired YyCxCz value.

To achieve the goal, our strategy is to form a uniformly sampled $9 \times 9 \times 9$ grid points in CMY space that can represent Indigo printer gamut. We print these patches on test pages where each patch has a certain NPAC value and we use X-Rite DTP 70 to get each patchs CIEXYZ value and transfer into YyCxCz space. We also store those measured YyCxCz values and NPAC as a forward and inverse mapping.

Forward Mapping and inverse Mapping are the same, but we treat them the same but use in different perspective.

Forward mapping: In CMY space, every grid point has an associated YyCxCz value.

Inverse mapping: In YyCxCz space, every grid point has an associated CMY combination, and an NPAC value, but we are not using CMY combinations when inverse mapping in use.

2.2 Forward Mapping

For the Indigo printers, they use three colorants C,M,Y only, and they don't use K or called Black because of enhanced productivity mode (EPM). If a customer wants to print black color, Indigo press will print CMY three colorants on top of each other to generate color black. This step will also save times and generate prints faster. Therefore, CMY three combination will generate in total of 8 colors or called 8 NPs (Neugebauer Primaries). Here is all 8 possible colors:

W White

Y Yellow

C Cyan

CY Cyan and Yellow (Green)

M Magenta

MY Magenta and Yellow (Red)

CM Cyan and Magenta (Blue)

CMY Cyan, Magenta and Yellow (Black)

To start, we generated solid patches of 8 NPs, printed and measured 8 NPs XYZ values using X-Rite DTP70, and converted them into YyCxCz. This is shown in Figure 2.1.

After we have the polyhedron of Indigo press printer gamut, and we want to have a better description of it, we need to generate more points that are based on actual target halftoning algorithm – PARAWACS halftone.

Here is the procedure:

First, we uniformly sampled along each edge of cube CMY to form $9 \times 9 \times 9$ grid points in CMY space. This is shown in Figure 2.2.

Second, because we do not have a model that can represent halftone behavior, we only use four vertices and make tetrahedra rather than use more eight vertices and make only one polyhedron. Therefore, we tessellate CMY space into six tetrahedra.

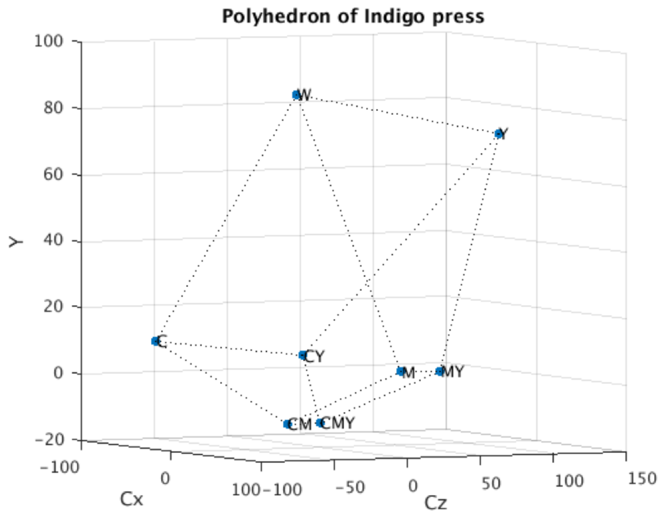


Fig. 2.1. Polyhedron of Indigo press

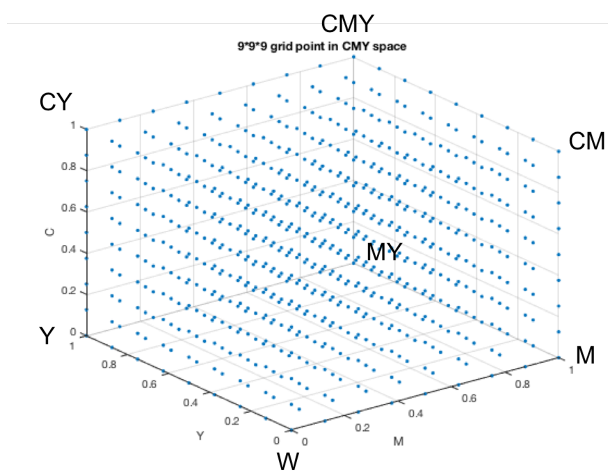


Fig. 2.2. $9 \times 9 \times 9$ grid points in CMY space

Noted that tessellation of Indigo press printer gamut is not unique. This is shown in Figure 2.3 and Figure 2.4.

Next step is for each grid point, we find one of six tetrahedra that contains it. And obtain corresponding YyCxCz tri-stimulus value via tetrahedral interpolation (2 cases).

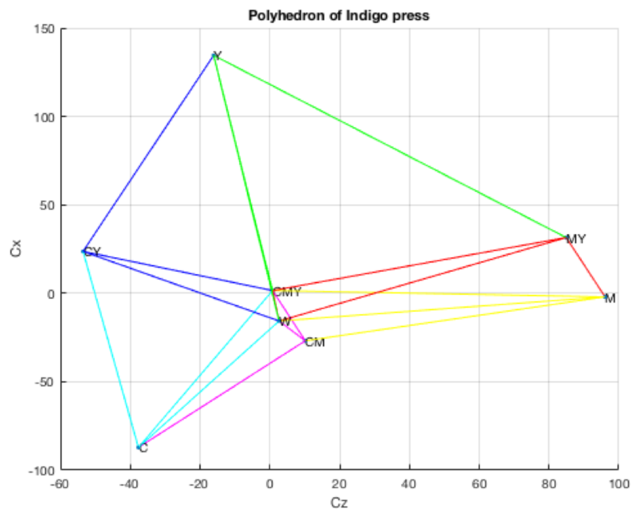


Fig. 2.3. $9 \times 9 \times 9$ grid points in CMY space

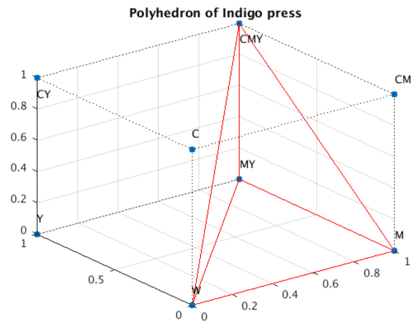
Case 1: The point lies inside tetrahedron – Do tetrahedral interpolation. This is shown in Figure 2.5.

Case 2: The point lies on a 3-D surface or an edge of the tetrahedron – Do area interpolation.

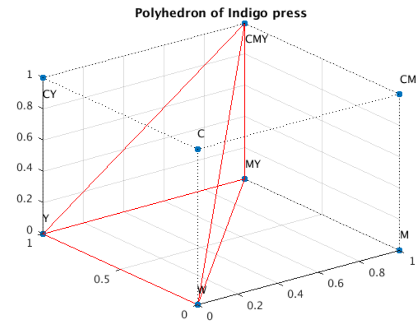
Last, find each grid points NPAC value. Its NPAC value is the interpolation value from previous step.

2.3 Generate, Print and Measure Test pages

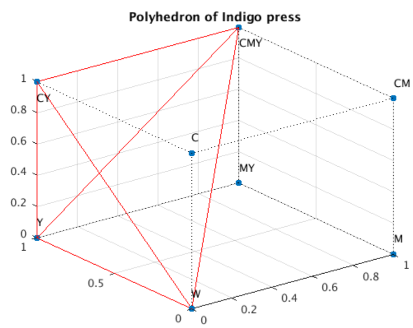
After we generate $9 \times 9 \times 9$ grid points in YyCxCz space, and each grid points has an associate NPAC. We want to print those grid points into patch and measured their actual YyCxCz value based on their NPAC, and multiply NPAC by 254 and calculate accumulated NPAC. We first generate a 600×600 continuous-tone patch which the patch's YyCxCz is the desired value and based on its NPAC and accumulated NPAC, tile PARAWACS selection matrix over the patch without any overlap. For a pixel in the patch, find corresponding value in the PARAWACS selection matrix. It will be a



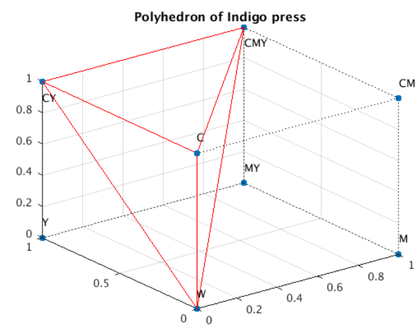
(a) W-M-MY-CMY



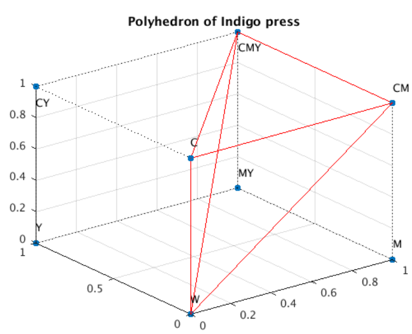
(b) W-Y-MY-CMY



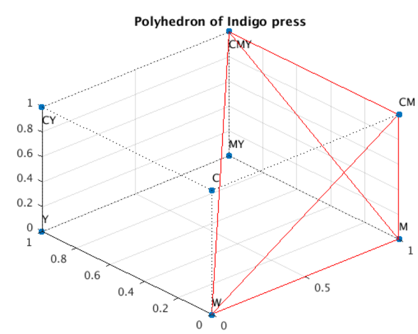
(c) W-Y-CY-CMY



(d) W-C-CY-CMY



(e) W-C-CM-CMY



(f) W-M-CM-CMY

Fig. 2.4. Six big tetrahedra.

number between 0-254. Comparing this number with its NPAC, a number between 0-254 too, find the number falls in which one of four NP ranges. Then the corresponding NP will be the halftone patch applied PARAWACS selection matrix in a pixel.

NP order is:

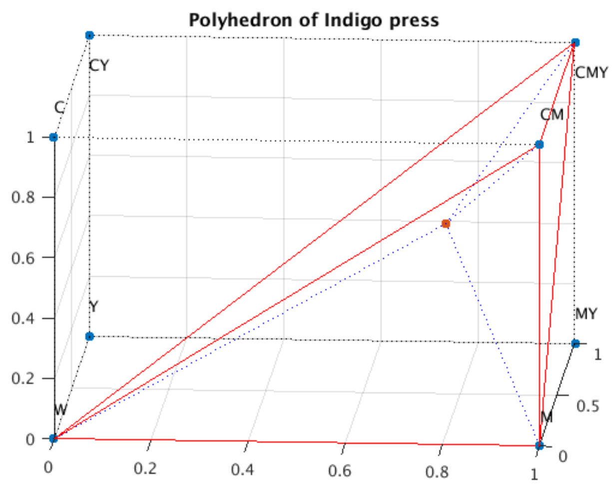


Fig. 2.5. A grid points in CMY space and its spatial location

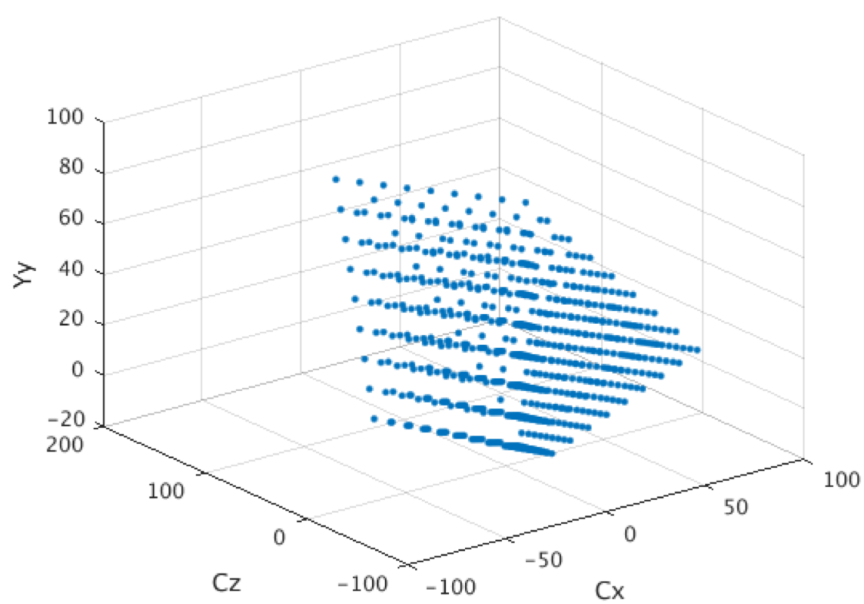


Fig. 2.6. $9 \times 9 \times 9$ grid points in YyCxCz space

$$[\text{W Y C CY(Green) M MY(Red) CM(Blue) CMY}] = [0\ 1\ 2\ 3\ 4\ 5\ 6\ 7]$$

Here is an example to illustrate this procedure. A continuous-tone patch's NPAC is:

$$[\text{W M MY CMY}] = (0.1,0.2,0.5,0.2)$$

the rest of four NPs are 0s. Or we can represent this NPAC as

$$[\text{W Y C CY M MY CM CMY}]=(0.1, 0, 0, 0, 0.2, 0.5, 0, 0.2, 0)$$

The sum of NPAC must be 1. After we multiply this NPAC by 254 and round up to integers, we will get

$$[\text{W M MY CMY}]=(25, 51, 127, 51)$$

or accumulated NPAC:

$$[\text{W M MY CMY}]=(25, 76, 203, 254)$$

After look at the same location in the PARWACS selection matrix, if the corresponding PARAWACS selection matrix pixel is, say 60. And we know that 60 falls in between 25 and 76. Therefore, in the PARAWACS halftone, the NP or color correspond should be M.

Repeat this procedure for $9 \times 9 \times 9$ patches.

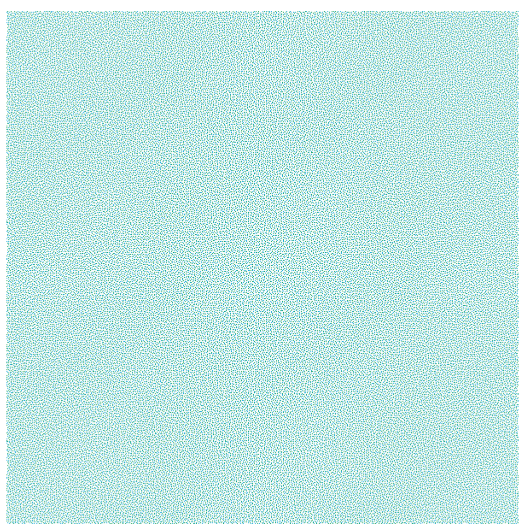


Fig. 2.7. A 600×600 halftone patch

In the test pages, Black blocks are registration marks. We add Test block for Color Plane Registration(CPR) to find how much each color plane is displaced relative to

Fig. 2.8. First page of test pages

the magenta color plane. This is shown in Fig.2.9.One big patch is 600*600, made up by 9 200*200 small patches. After we capture patches ' XYZ value form X-Rite DTP 70, we only use center ones XYZ value as this patchs XYZ. Each patch location is randomized and repeated 5 times. There are 44 test pages in total. This is shown in Fig.2.8

2.4 Display of printed halftone patches

After we received patches from Boise, we used QEA PIAS II to capture and display of those printed halftone patches, and see the printing quality. This is shown in Fig.2.10-13.

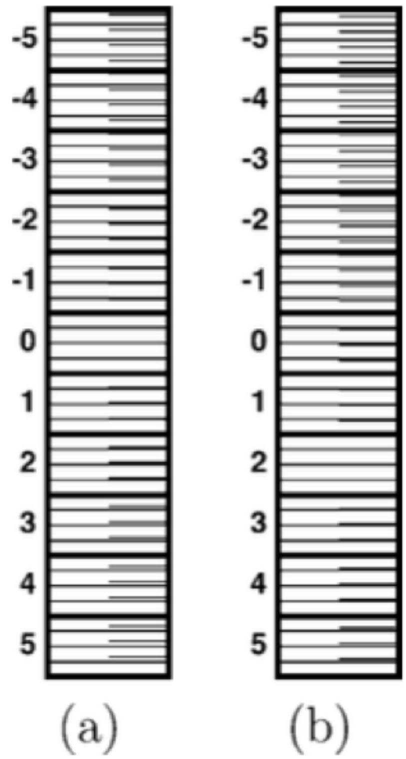
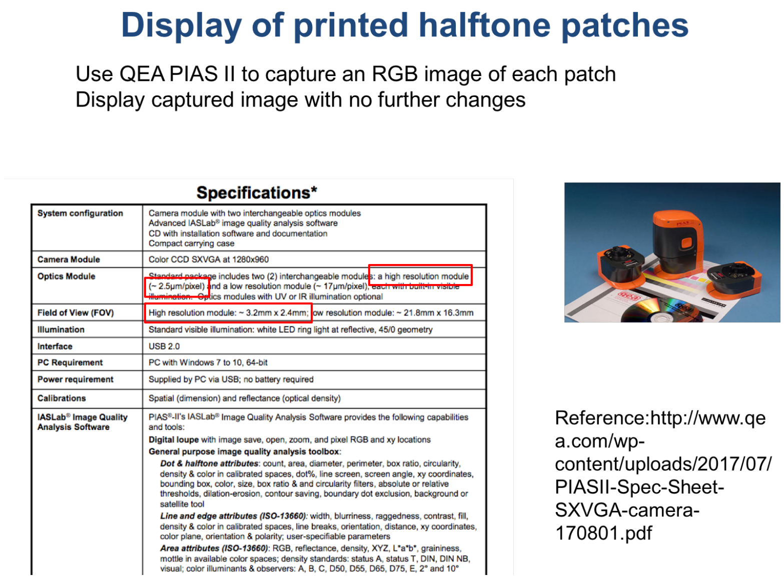


Fig. 2.9. CPR Test block

2.5 Inverse Mapping

Since Forward and Inverse Mapping are same set of data. I will illustrate the measured Inverse Mapping after we received test pages from Boise, ID. This is shown in Fig.2.14.

We think the reason that makes measured grid points did not look the same as the ideal grid points is because of dot gain, where we assume each pixel is a square, but in reality, it could be a circle dot, and to make sure we don't have any space that did not covered by ink, we need to make circle dot bigger. Or it could be the problem of Mis-registration.



System configuration	Camera module with two interchangeable optics modules Advanced IASLab® image quality analysis software CD with installation software and documentation Compact carrying case
Camera Module	Color CCD SXVGA at 1280x960
Optics Module	Standard package includes two (2) interchangeable modules: a high resolution module (~ 2.5μm/pixel) and a low resolution module (~ 17μm/pixel), each with built-in visible illumination. Optics modules with UV or IR illumination optional
Field of View (FOV)	High resolution module: ~ 3.2mm x 2.4mm; low resolution module: ~ 21.8mm x 16.3mm
Illumination	Standard visible illumination: white LED ring light at reflective, 45/0 geometry
Interface	USB 2.0
PC Requirement	PC with Windows 7 to 10, 64-bit
Power requirement	Supplied by PC via USB; no battery required
Calibrations	Spatial (dimension) and reflectance (optical density)
IASLab® Image Quality Analysis Software	PIAS®-II's IASLab® Image Quality Analysis Software provides the following capabilities and tools: **Digital loupe** with image save, open, zoom, and pixel RGB and xy locations **General purpose image quality analysis toolbox**: ***Dot & halftone attributes***: count, area, diameter, perimeter, box ratio, circularity, density & color in calibrated spaces, dot%, line screen, screen angle, xy coordinates, bounding box, color, size, box ratio & and circularity filters, absolute or relative thresholds, dilation-erosion, contour saving, boundary dot exclusion, background or satellite tool ***Line and edge attributes (ISO-13660)***: width, blurriness, raggedness, contrast, fill, density & color in calibrated spaces, line breaks, orientation, distance, xy coordinates, color plane, orientation & polarity; user-specifiable parameters ***Area attributes (ISO-13660)***: RGB, reflectance, density, XYZ, L*a*b*, graininess, mottle in available color spaces; density standards: status A, status T, DIN, DIN NB, visual; color illuminants & observers: A, B, C, D50, D55, D65, D75, E, 2° and 10°

Fig. 2.10. QEA PIAS II

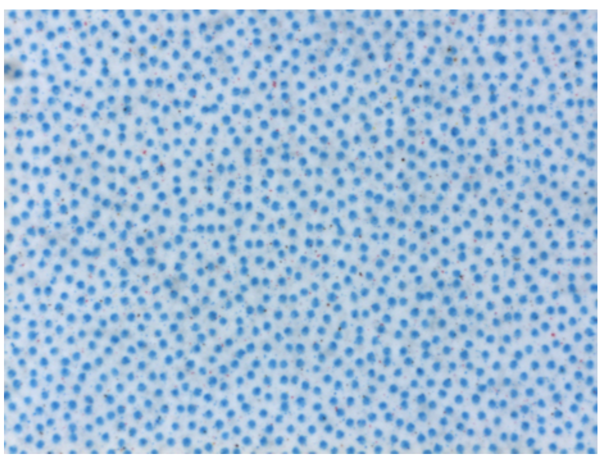


Fig. 2.11. 12.5 % of Cyan

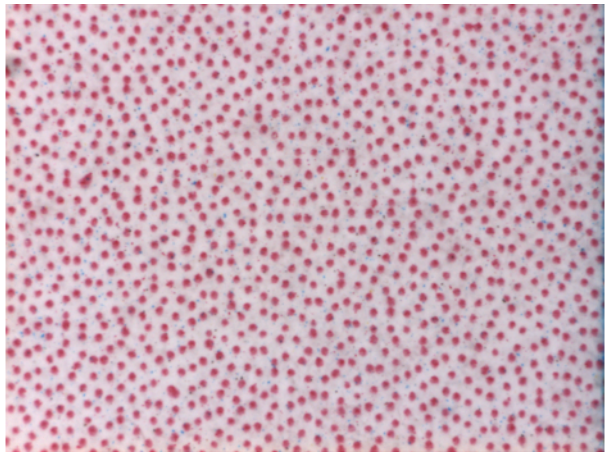


Fig. 2.12. 12.5 % of Magenta

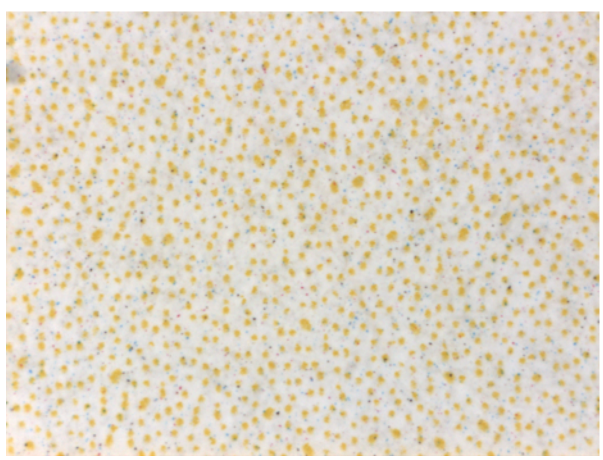


Fig. 2.13. 12.5 % of Yellow

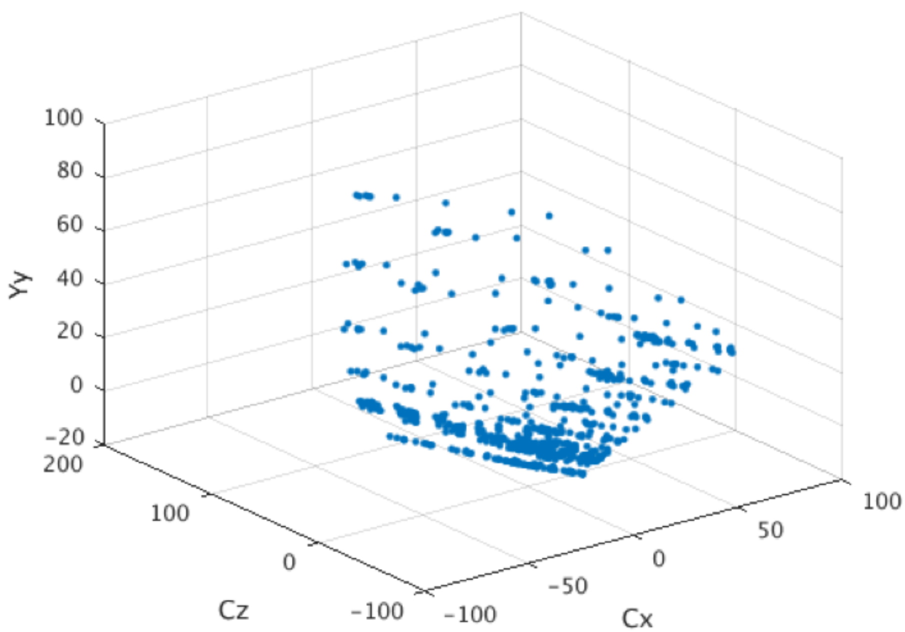


Fig. 2.14. Measured grid points

3. IMAGE-DEPENDENT GAMUT MAPPING

3.1 Introduction

Before applying halftone algorithm, there is a gamut mapping procedure need to do so that every pixel in an image can be print in Indigo press. This step requires forward mapping of Indigo press. Our gamut mapping procedures was based on two papers.

Here is the flowchart of Gamut mapping.

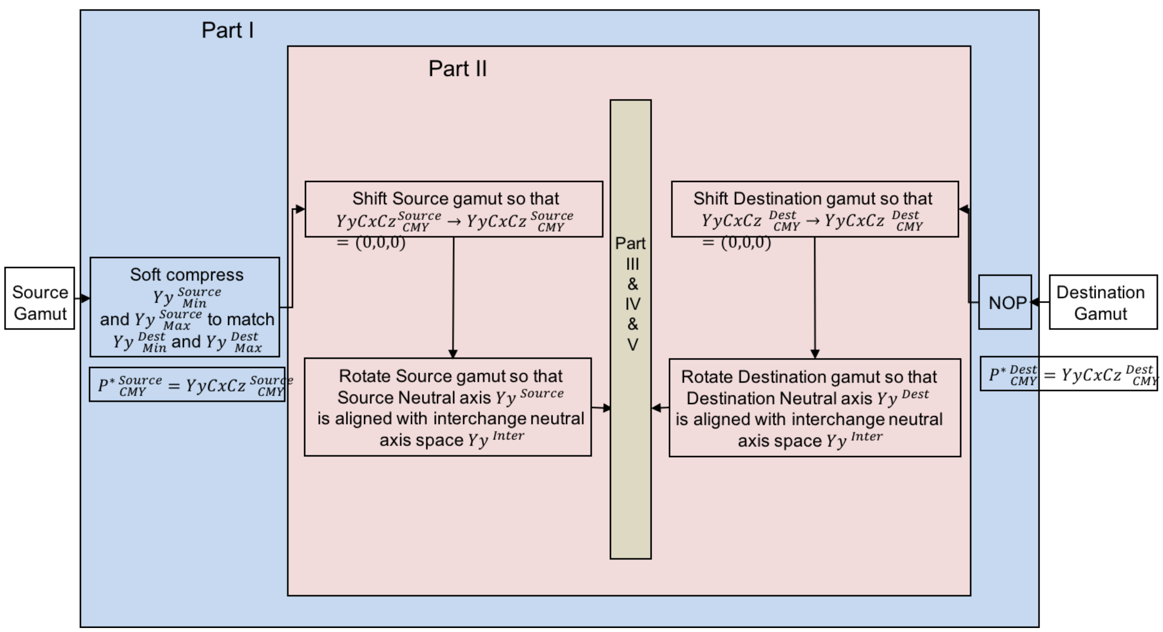


Fig. 3.1. Part I of Gamut Mapping

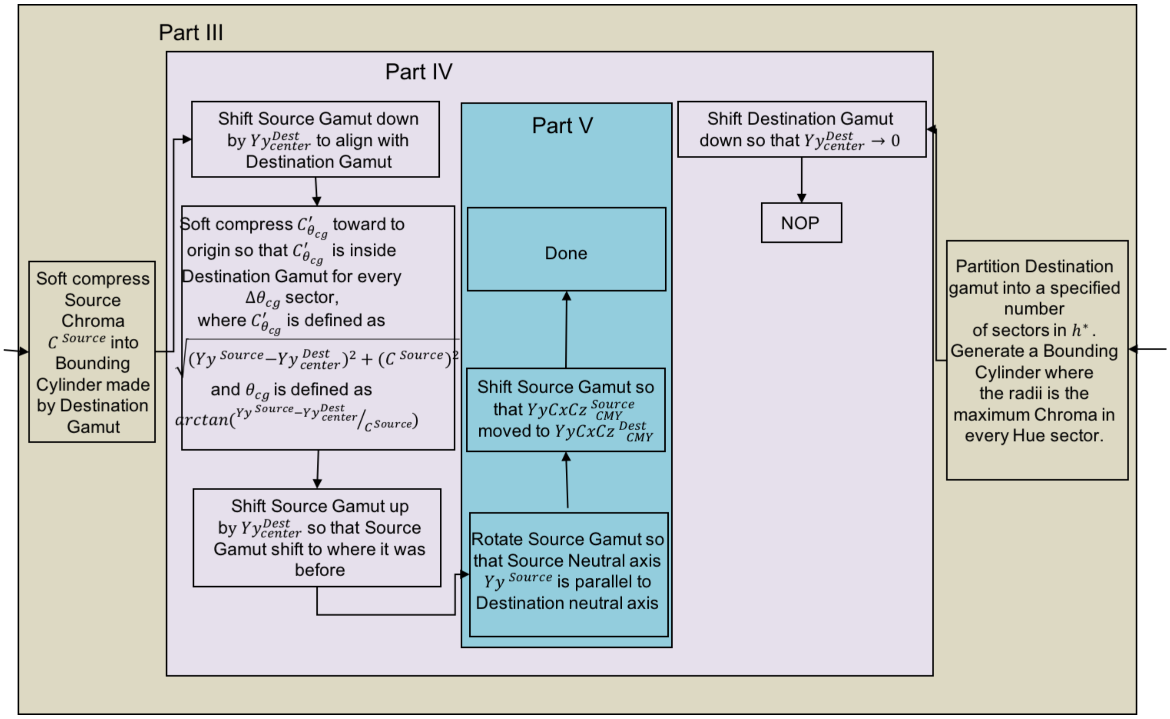


Fig. 3.2. Part II of Gamut Mapping

3.2 Before Gamut mapping

For a given image that we want to print, first we need to de-gamma the image, and then transfer image from sRGB to CIEXYZ. The reason that we don't use CIEXYZ or CIELAB to do gamut mapping is because small changes in CIEXYZ would result in small or large perceptual change. So we have to use or formulate a perceptually uniform color space. Since YyCxCz space is the linearize version of CIELAB, we then decided to use YyCxCz rather than CIELAB.

The formula for transfer sRGB and CIEXYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

where

$$\begin{bmatrix} M \end{bmatrix}^{-1} = \begin{bmatrix} 3.1338 & 1.6168 & 0.4906 \\ 0.9787 & 1.9161 & 0.0334 \\ 0.0719 & 0.2289 & 1.4052 \end{bmatrix} \tag{3.2}$$

and the formula between CIEXYZ and $Y_y C_x C_z$ is

$$\begin{bmatrix} Y_y \\ C_x \\ C_z \end{bmatrix} = \begin{bmatrix} \frac{1}{X_0} & 0 & 0 \\ 0 & \frac{1}{Y_0} & 0 \\ 0 & 0 & \frac{1}{Z_0} \end{bmatrix} \begin{bmatrix} 0 & 116 & 0 \\ 500 & -500 & 0 \\ 0 & 200 & -200 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.3}$$

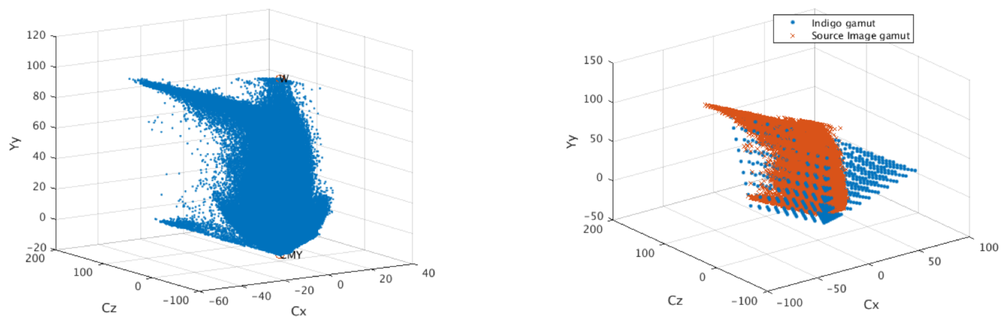
where $X_0 Y_0 Z_0$ is the Indigo D50 white point.

Fig. 3.3. Source Image

3.3 Gamut mapping

3.3.1 Step One: Soft compress source lightness

This step is our novel step in Gamut mapping. The reason we did this is because if we do compression directly, it could result in some image pixels would be still outside



(a) Source Image Gamut

(b) Source and Indigo Gamut

Fig. 3.4. Source and Indigo Gamut

of Indigo printer gamut, and compress lightness would eliminate this problem. Our goal is to soft compress source lightness Yy_{Max}^{Source} and Yy_{Min}^{Source} to match with Yy_{Max}^{Dest} and Yy_{Min}^{Dest}. where Source gamut means Image gamut and Destination means Indigo printer gamut.

Here is the equation to compress source lightness to match with Indigo printer gamut.

$$Yy^{Source} = \begin{cases} (1-\lambda)\frac{Yy_{Max}^{Dest}}{Yy_{Max}^{Source}}Yy^{Source} + \lambda Yy^{Source}, & \text{if } 0 \leq Yy^{Source} < Yy_{Max}^{Dest}. \\ (1-\lambda)\frac{Yy_{Max}^{Dest}}{Yy_{Max}^{Source}}Yy^{Source} + \lambda Yy_{Max}^{Dest}, & \text{if } Yy_{Max}^{Dest} \leq Yy^{Source} < Yy_{Max}^{Source}. \end{cases} \tag{3.4}$$

3.3.2 Step Two: Shift and Rotate

We need to shift and rotate Source gamut and Indigo gamut to align with Yy axis. First, we move both gamut CMY pixel to [0,0,0] and rotate to align with Yy axis.

3.3.3 Step Three: Compress into Bounding Cylinder

For Destination gamut: We partition Destination gamut into a specified number of divisions in h*. Each h* cell is a sector of Destination gamut with some angular extent

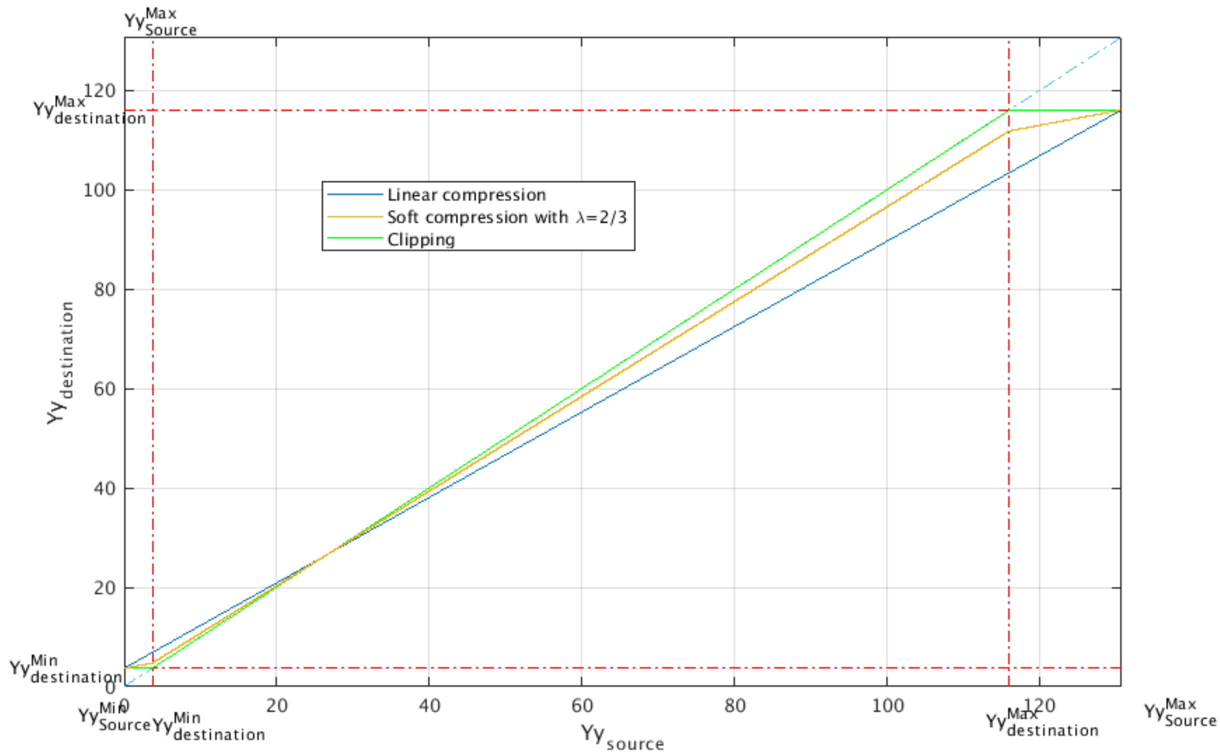
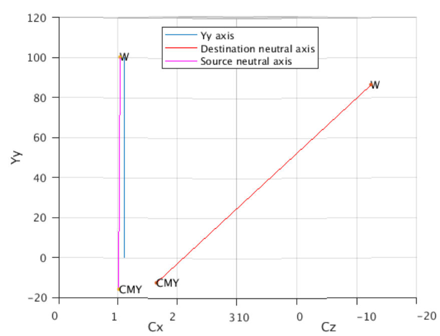
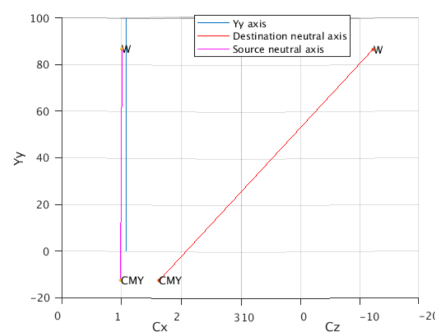


Fig. 3.5. Compression of Lightness



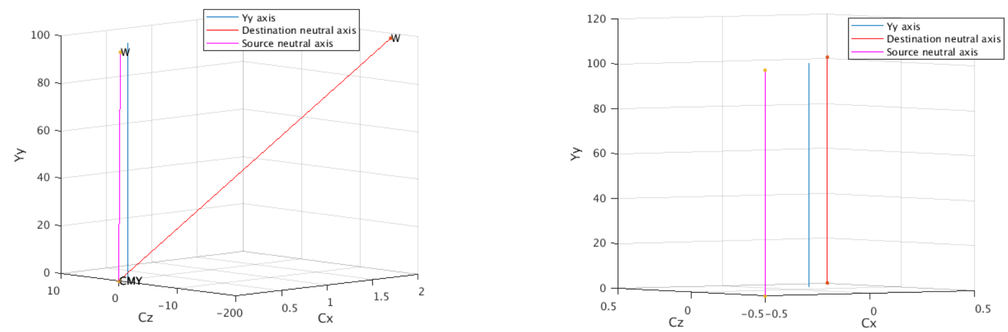
(a) Before Compress lightness



(b) After Compress lightness

Fig. 3.6. Compression of Lightness

Δh. Greatest chroma value c* within each h* cell will be the limit of the gamut, which means all the h*cells form the smallest cylinder containing the Destination gamut.

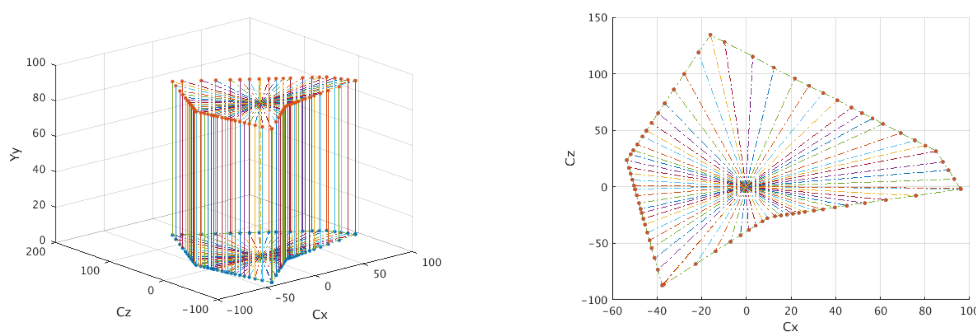


(a) Shift to origin

(b) Rotate to align with Yy axis

Fig. 3.7. Rotate and Shift Source and Indigo gamut

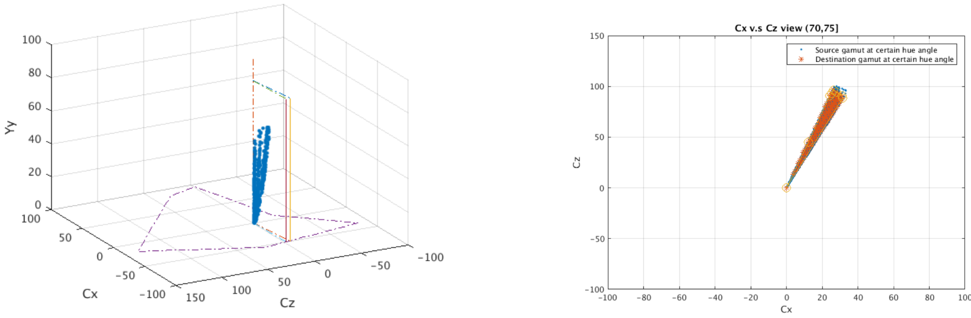
This cylinder extends from the lowest to the highest attainable device lightness and has radii which are constant with varying lightness but which vary with hue angle. For the input image (source): we partition Source gamut into a specified number of divisions in h*. Each h* cell is a sector of Source gamut with some angular extent Δh. we map input image to fit within the bounding cylinder determined in Steps 1-3 above by using the Compression (CMP) Algorithm. We repeat this gamut mapping procedure for each hue angle. Hue angle is 5 degree. This is shown in Fig.3.9 and Fig. 3.10.



(a) Bounding Cylinder

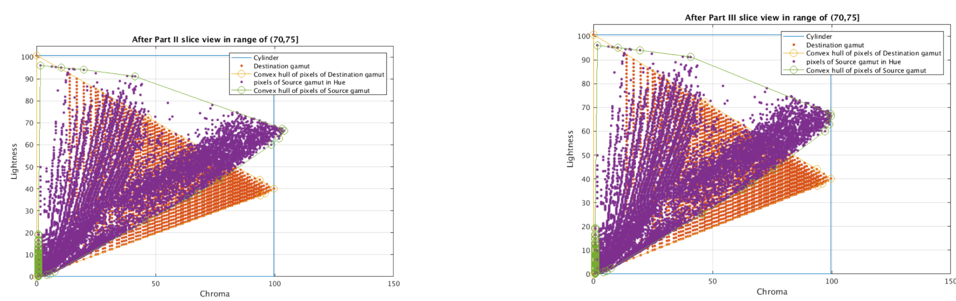
(b) Bounding Cylinder Top View

Fig. 3.8. Bounding Cylinder



(a) A sector (b) Cx v.s. Cz Top view

Fig. 3.9. Sectors view



(a) Before compression (b) After compression

Fig. 3.10. Compress Chroma only

3.3.4 Step Four: Compress into Gamut

By now, the input image pixels are inside the bounding cylinder, but not necessarily inside destination gamut. First we shift the source gamut and destination gamut down by the lightness of where cusp showed up. We use MATLAB build-in function convex hull to define boundary of each hue slice of gamut. We then divide Destination Gamut in to 180 $\Delta\theta_{cg}$ d for each $\Delta\theta_{cg}$. Use center compression to compress into the boundary. All pixels are mapped into destination gamut by changing saturation and lightness at the same time so that the pixels are compressed in the direction to the cusp. After compression, we shift both gamut up to where they were before. This is shown in Fig.3.11.

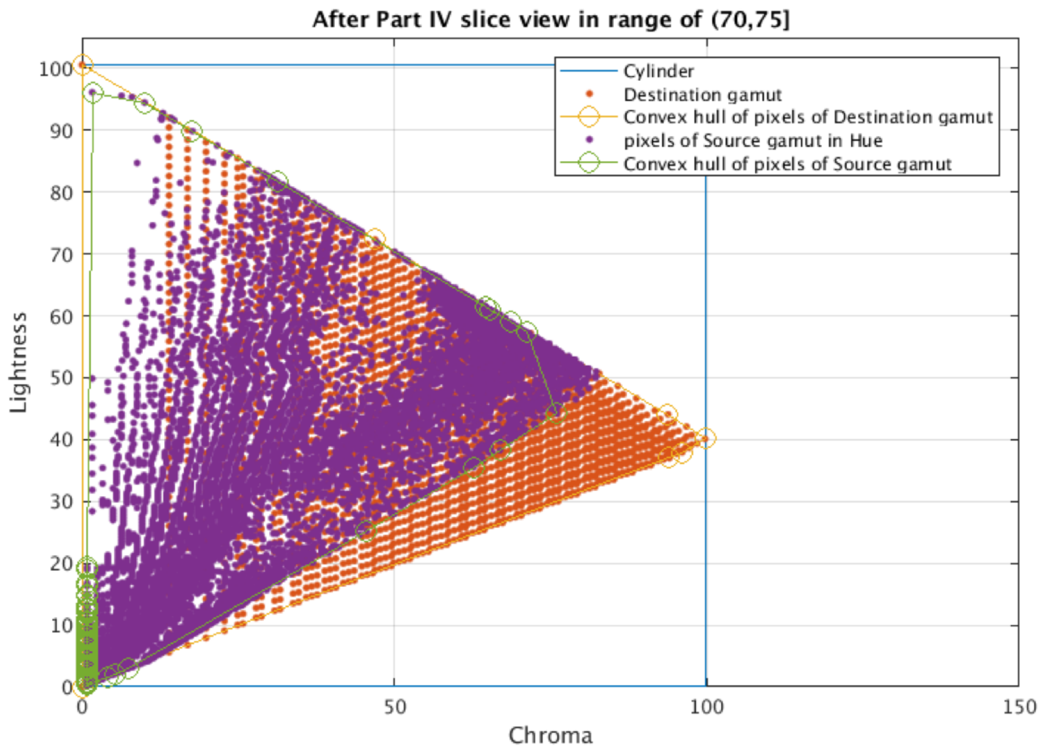


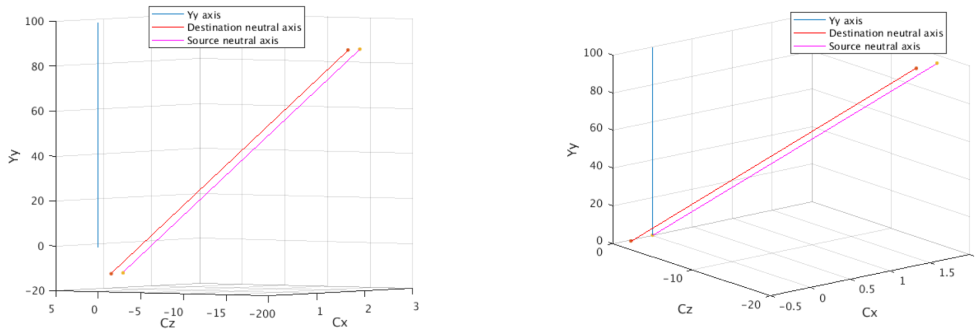
Fig. 3.11. Compress into Gamut

3.3.5 Step Five: Rotate and Shift back to Indigo press printer gamut

We rotate Source Gamut so that Source Neutral axis Yy^{Source} aligned with Destination neutral axis. We then shift Source Gamut so that Yy^{Source}_{CMY} moved to Yy^{Dest}_{CMY}. This is shown in Fig.3.12.

3.3.6 Result

This is shown in Fig.3.13.



(a) Rotate Back

(b) Shift

Fig. 3.12. Rotate and Shift back

3.3.7 Conclusion

We think that our result preserved most of detail of the original image.

(a) Before Gamut Mapping

(b) After Gamut Mapping

Fig. 3.13. Result comparison

4. PARAWACS HALFTONING

4.1 Introduction

After we finished Gamut mapping, which means every pixel in the continuous-tone image is able to be printed at Indigo press printer. Now we move to the next step, assign each pixel with one of eight NPs or called colors to make the continuous-tone image become a halftone.

4.2 Procedure

For every pixel in YyCxCz space, first find a big tetrahedron that contains the pixel. This step will make our next step faster for searching neighbors. Big tetrahedron means the vertices of the big tetrahedron are four NPs from the indigo press. Look up in the inverse mapping table, collect all grid points that are inside the same big tetrahedron and find the pixel's four nearest neighbors that formed a small tetrahedron containing the image pixel. Small tetrahedron means four vertices not necessarily are the NPs, they could be the grid points in $65 \times 65 \times 65$.

Containing means the image pixel could be either locate inside the small tetrahedron or on lie on a 3-D surface or an edge of the tetrahedron.

If an image pixel is locate inside the small tetrahedron, we could use tetrahedral interpolation to find this image pixel's NPAC. Otherwise, we could use barycentric interpolation.

After we know every image pixel's NPAC, multiply NPAC by 254 and calculate accumulated NPAC.

Tile PARAWACS selection matrix over the image without any overlap. For a pixel in the image, find corresponding value in the PARAWACS selection matrix. It will be a number between 0-254. Comparing this number with its NPAC, a number between 0-254 too, find the number falls in which one of four NP ranges. Then the corresponding NP will be the halftone image applied PARAWACS selection matrix in a pixel.

NP order is:

$$[\mathrm{W\ Y\ C\ CY(Green)\ M\ MY(Red)\ CM(Blue)\ CMY}] = [0\ 1\ 2\ 3\ 4\ 5\ 6\ 7]$$

Here is an example to illustrate this procedure. A Gamut mapped image pixel NPAC is:

$$[\mathrm{W\ M\ MY\ CMY}] = (0.1, 0.2, 0.5, 0.2)$$

the rest of four NPs are 0s. Or we can represent this NPAC as

$$[\mathrm{W\ Y\ C\ CY\ M\ MY\ CM\ CMY}] = (0.1, 0, 0, 0, 0.2, 0.5, 0, 0.2, 0)$$

The sum of NPAC must be 1. After we multiply this NPAC by 254 and round up to integers, we will get

$$[\mathrm{W\ M\ MY\ CMY}] = (25, 51, 127, 51)$$

or accumulated NPAC:

$$[\mathrm{W\ M\ MY\ CMY}] = (25, 76, 203, 254)$$

After look at the same location in the PARWACS selection matrix, if the corresponding PARAWACS selection matrix pixel is, say 60. And we know that 60 falls in between 25 and 76. Therefore, in the PARAWACS halftone, the NP or color correspond should be M.

Repeat this procedure for every pixel in the image.

4.3 Result PARAWACS halftoning

Here is the result PARAWACS halftoning and detail comparison with continuous-tone image. This is shown in Fig. 4.1. We think that we preserved most of details of

(a) Continuous-tone Image

(b) PARAWACS Halftone

Fig. 4.1. Result comparison

the image, though sky color seems a little bit off as well as brick color. Overall, we believe the quality of the image is good.

5. SUMMARY

In this thesis, we continue our previous efforts to adapt DBS algorithm to use with the Indigo liquid EP printing technology to render halftone images.

We describe a complete color management pipeline for halftoning with a PARAWACS screen matrix designed using DBS. For any given continuous-tone image, we can easily process the image through the pipeline and return a good quality halftone image.

For the propose of actually knowing the prints quality, for the first time, we show actual printed patches obtained using our process. We first tessellate the output color space into six global tetrahedra that each share the neutral axis, as an edge. By uniformly sample along CMY space, we render 729 halftone images. Those 729 printed patches describe HP Indigo press printer gamut or the output color space. This process let us know what the HP Indigo press printer gamut actually looks like rather than only a convex hull that includes only eight colors of Indigo press printer gamut. Based on those printed patches, we finally generate a inverse mapping that given a pixel value that is inside print gamut range, we can find a desired NPAC that printed will yield desired pixel value using nearest neighbor and tetrahedral interpolation method.

Our gamut mapping is performed in the YyCxCz color space, and is image-dependent. It incorporates several stages of alignment between the input and output spaces, as well as several stages of compression, including a lightness compression as the first step.

This report gives us a complete procedures on rendering halftone image based on the pipeline, it has some drawbacks too. For the future work, we can actually send the halftone image to HP Indigo press, and see whether the actually print matches with our prediction. Besides this, we can have a closer look the region of sky in the image, and find out what makes the sky color slightly different than it should be.

REFERENCES

REFERENCES

[1] R. Ulichney *Digital Halftoning* MIT press, 1987.

[2] M. Analoui and J. P. Allebach, *Model-based Halftoning using Direct Binary Search* Proceedings of SPIE Vol.1666 1992 pp.96-109.

[3] D. J. Lieberman and J. P. Allebach, *Efficient Model Based Halftoning Using Direct Binary Search* IEEE International Conference Image Processing 1997.

[4] 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 No.11, 2000 pp.1950-1963.

[5] J. P. Allebach, *DBS: retrospective and future directions* Proceedings of SPIE Vol.4300 2001 pp.358-376.

[6] J. Morovic, P. Morovic, M. G. Encrenaz, J. Lammens and Y. Richard, *Color separation into neugebauer primary area coverage vectors* U.S. Patent 8,213,055 B2, 2012.

[7] J. Morovic, P. Morovic and J. Arnabat *"HANS: Controlling ink-jet print attributes via neugebauer primary area coverages"* IEEE Transations on Image Processing, VOL. 21, NO. 2, 2012 pp.688-696

[8] P. Morovic, J. Morovic, J. Gondek, M. Gaubatz, and R. Ulichney, *PARAWACS Color halftoning with a single selector matrix* Color and Imaging Conference vol. 2016, no.1 Society for Imaging Science and Technology, 2016 pp.41-46.

[9] P. Morovic, J. Morovic, J. Gondek, M. Gaubatz, and R. Ulichney, *"Direct Pattern Control Halftoning of Neugebauer Primaries"* IEEE Transations on Image Processing, VOL. 26, NO. 9, 2017

[10] W. Xi, T. Frank, Y. Ben-Shoshan, R. Ulichney and J. P. Allebach, *"Color CLU-DBS halftoning based on neugebauer primary area coverage: Improving the breed"* Electronic Imaging, Color Imaging XXII: Displaying, Processing, Hardcopy and applications. 2018

[11] R. S. Gentile, E. Walowlt, and J. P. Allebach, *A Comparison of Techniques for Color Gamut Mismatch Compensation* Journal of Imaging Technology Volume 16, Number 5, October 1990.

[12] M. Wolski, J. P. Allebach and C. A. Bouman, *Gamut Mapping Squeezing the Most out of Your Color System* IS&T and SIDs 2nd Color Imaging Conference: Color Science, System and Application (1994)-89.

[13] J. Morovic, *Color Gamut Mapping* Wiley, 2008.

[14] W. Jang, M. Chen, J. P. Allebach and G. Chiu, *Print Quality Test Page* Journal of Imaging Science and Technology, Volume 48, Number 5, September/October

[15] O. Rodrigues, *"Des lois gometriques qui regissent les dplacements d' un systme solide dans l' espace, et de la variation des coordonnes provenant de ces dplacement considres indpendent des causes qui peuvent les produire"* The Journal de Mathmatiques Pures et Appliques 5 1840 pp.380440.

[16] J. M. Kasson, W. Plouffe, and S. I. Nin, *Tetrahedral interpolation technique for color space conversion* Device-Independent Color Imaging and Imaging Systems Integration Vol. 1909 International Society for Optics and Photonics, 1993 pp.127-139.

[17] T. Molleran and J. F. Hughes *Efficiently Building a Matrix to Rotate One Vector to Another* Journal of Graphics Tools, 4:4, 1-4 1999.