

NPAC FM color halftoning for the Indigo press: challenges and solutions*

Jiayin Liu¹, Tal Frank², Yotam Ben-Shoshan², Robert Ulichney³, and Jan Allebach¹; ¹Purdue University (United States), ²HP Inc. (Israel), and ³HP Labs, HP Inc. (United States)

Abstract

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 $Y_3C_xC_z$ 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.

Introduction

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

Halftoning algorithms can be classified according to whether they produce dispersed-dot textures or clustered-dot textures. Dispersed-dot textures are comprised of isolated printer-addressable dots and clustered-dot textures are comprised of clustered printer-addressable dots. Halftoning algorithms also can be classified as Frequency modulation (FM) or Amplitude modulation (AM). Frequency modulation changes the density of dots: aperiodic dot spacing but dot size is fixed. Amplitude modulation generates a regular grid of dots: dot size varies but periodic dot spacing is fixed.

Presently, there are two dominant printing technologies, one is the electrophotographic (EP) process with a laser writing system and the other is inkjet (IJ). Clustered-dot textures have been used widely in electrophotographic printers and dispersed-dot textures have been used widely in inkjet printers.

* Research supported by the HP Indigo Division, Rehovot, ISRAEL.

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 the perceived error between the continuous-tone and halftone images. The direct binary search (DBS) algorithm [1] [2] [3] [4] first computes the mean-squared error (MSE) between the filtered continuous-tone image and the filtered initial halftone image. Then, by scanning pixel by pixel through the halftone image and applying toggling and swapping to minimize the MSE, the algorithm stops when the MSE cannot be further reduced.

HP Indigo's Enhanced Productivity Mode (EPM) can provide faster throughput and obtain 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 yields 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).

Therefore, in this paper, we propose a new color management pipeline for the HP Indigo press using FM halftoning based on EP technologies. This work is a continues work that was reported in [5] [6].

Image-dependent gamut mapping

Due to the color gamut mismatch between display and print, gamut mapping is needed to display and print with a satisfactory level of visual quality. The case we focus on is that some input colors are not realizable with our Indigo 7000 series printing system. Our gamut mapping method is based on [6] [7] [8].

For a given source image that we want to print, first we need to de-gamma the image, transfer the image from sRGB to CIEXYZ, and then from CIEXYZ to $Y_3C_xC_z$ with our measured D50 white point. A flowchart of our method is shown at the end of this paper.

Part One: Soft compress source lightness

For a source image, we would like to soft compress lightness in every pixel because if we do compression in both lightness and chroma at the same time, it could result in some image pixels still being outside of Indigo printer gamut. Compressing lightness first will eliminate this problem. Our goal is to soft compress the source lightness Y_{Max}^{Source} and Y_{Min}^{Source} to match with Y_{Max}^{Dest} and Y_{Min}^{Dest} . Here in (1), the source gamut means the source image gamut and the destination gamut means the Indigo press printer gamut.

Figure 1 is the plot of equation (1). Here we compress source

lightness to match with the destination gamut lightness with control parameter λ which is $\frac{2}{3}$ in our case [7].

$$Y_y^{Dest} = \begin{cases} (1 - \lambda) \frac{Y_y^{Dest}}{Y_y^{Source}} Y_y^{Source} + \lambda Y_y^{Source}, & \text{if } 0 \leq Y_y^{Source} < Y_y^{Dest} \\ (1 - \lambda) \frac{Y_y^{Dest}}{Y_y^{Max}} Y_y^{Source} + \lambda Y_y^{Dest}, & \text{if } Y_y^{Dest} \leq Y_y^{Source} < Y_y^{Max} \end{cases} \quad (1)$$

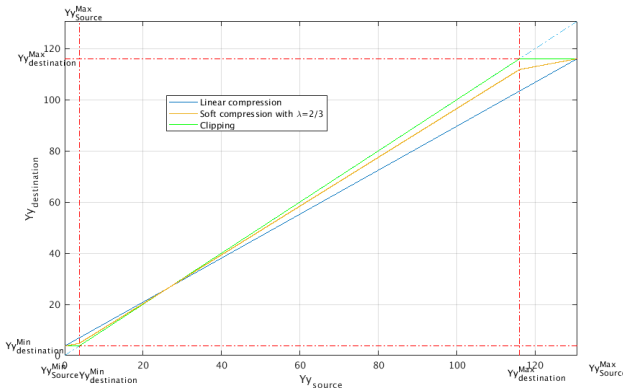


Figure 1: Compression of lightness.

For the rest, we followed the method in [8] for compression.

Part Two: Shift and rotate

We want to shift and rotate the source gamut and the destination gamut to align with the Y_y axis so that compression can be made. We move both source and destination gamut's CMY pixel values to the origin and rotate them to align with the Y_y axis [9] [10].

Part Three: Compress into bounding cylinder

For destination gamut, we use MATLAB built-in function convex hull to define the boundaries of the destination gamut based on 8 NPs (Neugebauer Primaries: W, Y, C, CY, M, MY, CM, and CMY). Then, we partition the destination gamut into a specified number of divisions in h^* . Each h^* cell is a sector of the destination gamut with some angular extent Δh . We chose to have 72 sectors and each hue sector angle is 5 degrees, as a top view and a side view both shown in Figure 2. The largest chroma value c^* within each sector will be the boundary of the gamut, which means 72 sectors form the smallest cylinder that contains the destination gamut [8].

For the source image, shown in Figure 3, we partition the corresponding source gamut into 72 sectors. Figure 4 shows a sector slice view of both the source and the destination gamut, we map the source gamut (in purple dots) to fit within the bounding cylinder (shown as the blue line) using the horizontal compression method from [8]. The compression result is shown in Figure 5. We repeat this gamut mapping process for each hue sector.

Part Four: Compress into gamut

By now, the source gamut is inside the bounding cylinder, but not necessarily inside the destination gamut. We divide the destination gamut into 180 angular slices for each hue sector and use

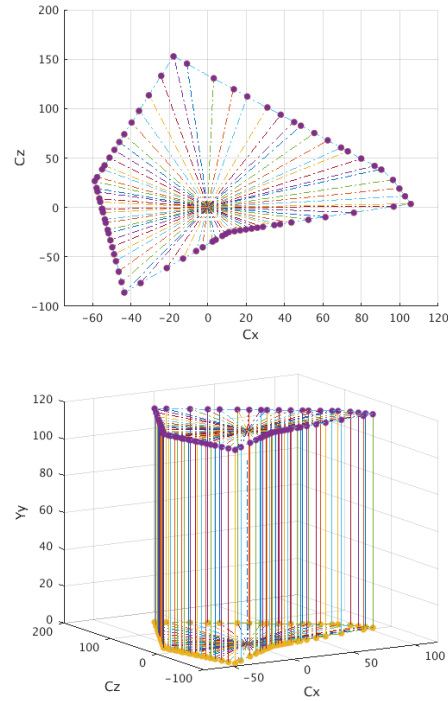


Figure 2: Bounding cylinder top view (up); and side view (down).

center compression to compress source gamut into the destination gamut for every slice. All pixels are mapped into the destination gamut by compressing the saturation and the lightness at the same time so that the pixels are compressed toward the center where the cusp occurs. The result is shown in Figure 6.

Part Five: Rotate and shift back to Indigo press gamut

We rotate the source gamut so that the source neutral axis Y_y^{Source} is aligned with the destination neutral axis. We then shift the source gamut so that $Y_y^{Source}_{CMY}$ is moved to $Y_y^{Dest}_{CMY}$. If we transfer every pixel back to sRGB from $Y_y C_x C_z$, the source image after gamut mapping can be viewed, as shown in Figure 7. This may be compared with the original image shown in Figure 3.

Characterization and development of forward and inverse mappings based on Indigo 7000 series prints

To develop our color management framework, we first want to generate a forward mapping that defines the Indigo press (destination) gamut, and given a CMY (Cyan, Magenta and Yellow) printing combination, find its associated NPAC (Neugebauer Primary Area Coverage) [11] [12] value and $Y_y C_x C_z$ value. Based on this result, we then develop an inverse mapping that given a $Y_y C_x C_z$ value, will yield its NPAC value that when printed will yield the desired $Y_y C_x C_z$ value. In the Indigo press, the NPAC is an 8-tuple entry that indicates how much of each NP in percentage should be printed to yield the desired $Y_y C_x C_z$ value.

To achieve this goal, our strategy is to form a uniformly sampled $9 \times 9 \times 9$ grid of points in CMY space that can represent the Indigo press gamut. We print these patches on test pages where each patch has a certain NPAC value. We then use the spectropho-



Figure 3: Source image.

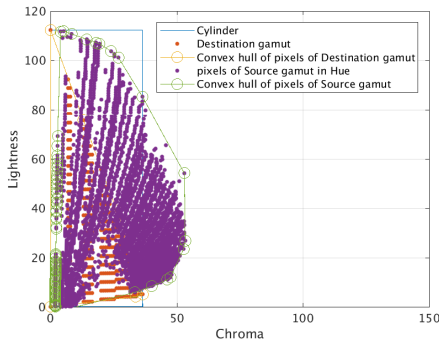


Figure 4: Before compress chroma.

tometer X-Rite DTP 70 to get each patch's CIEXYZ value and transfer it into $Y_y C_x C_z$ space. We store those measured $Y_y C_x C_z$ values and NPACs as forward and inverse mappings. The forward and inverse mappings are based on the same set of data, but viewed from different perspectives.

Forward mapping

As shown in Figure 8, what we have is 8 NPs, W, Y, C, CY, M, MY, CM, and CMY, and their $Y_y C_x C_z$ values. We want to generate a forward mapping and a destination gamut based on these eight NPs. Table 1 shows the Y_y, C_x, C_z coordinates of these 8 NPs, listed in order of decreasing lightness.

	C	M	Y	Y_y	C_x	C_z
W	0	0	0	116.000	0.000	0.000
Y	0	0	1	99.356	-17.926	152.892
C	1	0	0	31.799	-43.584	-86.167
CY	1	0	1	23.879	-58.991	26.737
M	0	1	0	23.362	105.657	3.551
MY	0	1	1	23.223	93.804	34.652
CM	1	1	0	3.792	10.897	-25.040
CMY	1	1	1	3.711	0.925	1.962

Table 1: Y_y, C_x, C_z coordinates of the 8 CMY Neugebauer primaries for the Indigo Press.

First, we uniformly sample along each edge of cube CMY to form $9 \times 9 \times 9$ grid of points in CMY space, which is the same as sampling $9 \times 9 \times 9$ grid of points in $Y_y C_x C_z$ space. Sec-

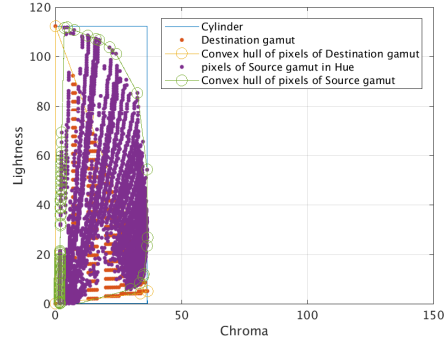


Figure 5: After compress chroma.

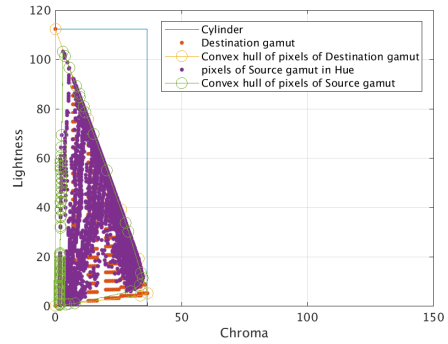


Figure 6: Compress into destination gamut.

ond, as shown in Figure 9, we tessellate the CMY space into six global tetrahedra where the W-CMY is an axis shared by each tetrahedron. The six tetrahedra are: W-Y-MY-CMY, W-MY-M-CMY, W-M-CM-CMY, W-CM-CY-CMY, W-CY-Y-CMY. Each vertex has an associated $Y_y C_x C_z$ value too. Note that the tessellation of Indigo press gamut is not unique.

Third, for every grid point, we find one of the six global tetrahedra described above that contains it and obtain the corresponding $Y_y C_x C_z$ tristimulus value via tetrahedral interpolation based on the tetrahedron's four $Y_y C_x C_z$ vertices. There are three cases. Case 1: the grid point lies inside a global tetrahedron: do tetrahedral interpolation [17]. An example is shown in Figure 10. Case 2: the grid point lies on a 3-D surface: do area interpolation. Case 3: the grid point lies on an edge of a tetrahedron: do line interpolation. Last, find every grid point's NPAC value by volume ratios to the global tetrahedron [17]. A global tetrahedron means its vertices are four NPs from the Indigo press.

Table 2 shows one entry of the forward and inverse mappings: the first three columns are one grid point's $Y_y C_x C_z$ values and remaining 8 columns are its NPAC.

After we generate $9 \times 9 \times 9$ grid points in the CMY / $Y_y C_x C_z$ space, and every grid points has an associated NPAC, we want to print every grid point based on its $Y_y C_x C_z$ and measure its actual $Y_y C_x C_z$ to form an measured inverse mapping.

Every pixel in a color halftone patch is a color, one of W, Y, C, CY, M, MY, CM, or CMY. Every color has a $Y_y C_x C_z$ coordinate

Y_y	C_x	C_z	W	Y	C	CY	M	MY	CM	CMY
81.226	36.658	9.107	0.625	0.000	0.000	0.000	0.125	0.250	0.000	0.000

Table 2: Sample entry of mapping between the Y_y, C_x, C_z coordinates and the corresponding NPAC.



Figure 7: Source image after gamut mapping.

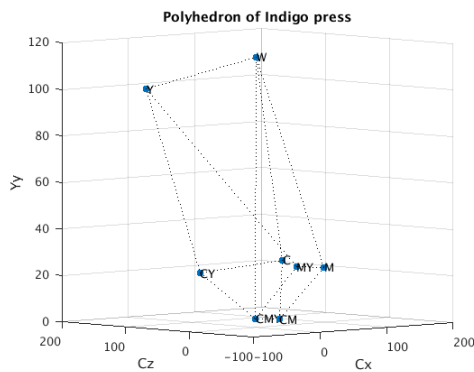


Figure 8: Polyhedron of Indigo press.

which is a fixed value. The way to calculate this halftone patch's ideal $Y_y C_x C_z$ value is to sum up all colors' $Y_y C_x C_z$ values, and divide by the total number of pixels.

We want to generate a 600×600 pixels patch, which is the patch's desired $Y_y C_x C_z$ value, based on its NPAC. We first tile the PARAWACS [5] [13] [14] selection matrix over the patch without any overlap. To find out which color should be printed at a pixel in the patch, we find corresponding value in the PARAWACS selection matrix. It will be a number between 0-254. Comparing this number with this patch's accumulated NPAC, which is a number between 0-254 too, we find in which one of four NP ranges the number falls. Then the corresponding NP will be printed at this pixel. The NP order is: [W Y C CY M MY CM CMY] = [0 1 2 3 4 5 6 7].

An Example

Here is an example to illustrate this process. After the tetrahedral interpolation, we find that a grid point's NPAC is:

$$[W \ M \ MY \ CMY] = (0.625, 0.125, 0.25, 0)$$

where the remaining of four NPs are 0s and the sum of an NPAC must be 1. After we multiply this NPAC by 254 and round up to integers, we will get

$$[W \ M \ MY \ CMY] = (158, 32, 64, 0)$$

or the accumulated NPAC as:

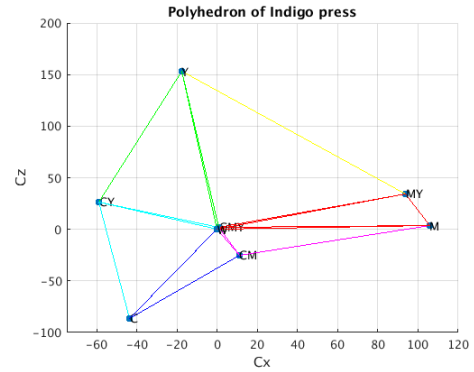


Figure 9: Tessellation of Indigo press 7000 series gamut.

$$[W \ M \ MY \ CMY] = (158, 190, 254, 254)$$

We check the same pixel location in the PARAWACS selection matrix, suppose the corresponding PARAWACS selection matrix pixel is, say 170. Since 170 falls in between 158 and 190, the NP or color corresponding to this pixel in the halftone patch should be M. Figure 11 is the halftone patch that is generated using above method. We repeat this procedure to generate $9 \times 9 \times 9$ patches.

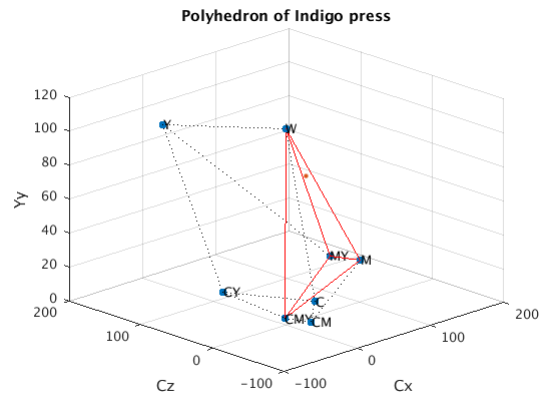


Figure 10: The grid point $[W \ M \ MY \ CMY] = (0.625, 0.125, 0.25, 0)$ in $Y_y C_x C_z$ space and its spatial location in the tetrahedron W-M-MY-CMY.

In our test page design, we add black blocks on both sides of the page, which are registration marks that are required if we use the X-Rite DTP 70. We also add a test blocks to assess Color Plane Registration (CPR) [16]. These test blocks enable us to find how much each color plane is displaced relative to the magenta color plane. The first page of our test pages is shown in Figure 12. After we receive the printed test pages, we use the QEA PIAS II to capture and display a magnified view of those printed halftone patches. There are three captures shown in Figure 13, 14, and 15.

Measured inverse mapping

Figure 16 is the plot of the measured points for the inverse mapping. We also update the $Y_y C_x C_z$ values in our inverse mapping from their ideal values to their measured values. Each dot in the plot has a measured $Y_y C_x C_z$ value and an associated NPAC. The measured values do not match the ideal values due to dot

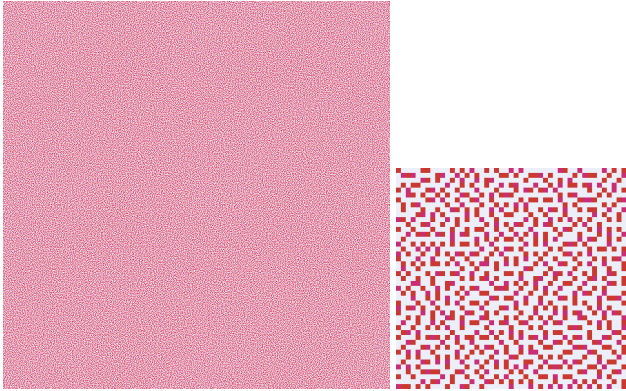


Figure 11: The halftone patch with NPAC [W M MY CMY] = (0.625,0.125,0.25,0) (left); and a zoomed-in version of the halftone patch (right). Note that it is difficult to distinguish the pixels that are red from those that are magenta.

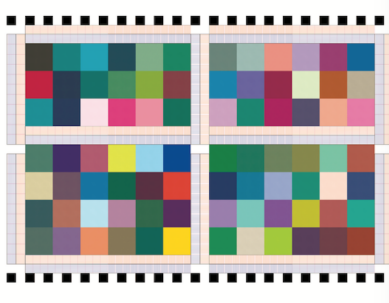


Figure 12: The first page of test pages.

gain, misregistration between color planes, and other aspects of non-ideal press behavior.

PARAWACS halftoning

In PARAWACS halftoning section, we assign every pixel in the gamut mapped source image with one of eight NPs to make the continuous-tone image become a halftone, just as we did in previous section to generate the constant-tone patches for the test pages.

Procedure

For every gamut mapped source image pixel in $Y_y C_x C_z$ space, we first find one of the six global tetrahedra that contains the pixel. Here, “containing” means a pixel could be located inside a small tetrahedron, lie on a 3-D surface, or lie on an edge of a tetrahedron. This step will make our next step faster for searching neighbors. Global tetrahedron means its vertices are four NPs from the Indigo press. We look up in the inverse mapping table, collect all grid points that are inside the same global tetrahedron and find that pixel’s four nearest neighbors in $Y_y C_x C_z$ space that form a small tetrahedron containing the image pixel. Small tetrahedron means four vertices that are not necessarily one of the NPs. They could be the points from the $9 \times 9 \times 9$ grid.

If this pixel is located inside one of the small tetrahedron in $Y_y C_x C_z$ space, we can use tetrahedral interpolation to find its NPAC by interpolating the four nearest neighbors’ NPACs. Otherwise, we use barycentric interpolation to find its NPAC.

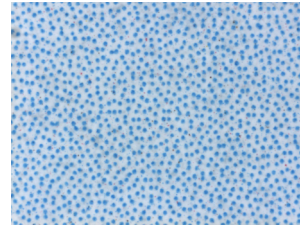


Figure 13: 12.5 % of Cyan.

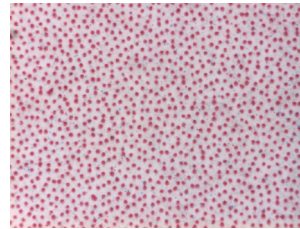


Figure 14: 12.5 % of Magenta.

After we have every pixel’s NPAC, we multiply the NPAC by 254, and calculate the accumulated NPAC. We tile the PARAWACS selection matrix over the gamut mapped source 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 the pixel’s accumulated NPAC, a number between 0-254 too, we find that this number falls into one of the four NP ranges. Then, the corresponding NP will be the halftone value at this pixel location.

Result PARAWACS halftone image

Figure 17 shows the resulting PARAWACS halftone image. Generally speaking, the image quality is quite good.

Conclusion

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 simulated halftone image. For the propose of actually knowing the print quality, we can analyze actual printed patches using our process. This work gives us a complete procedure for rendering a halftone image based on our color management pipeline.

References

- [1] M. Analoui and J. P. Allebach, “Model-based Halftoning using Direct Binary Search,” *Proceedings of SPIE*, vol. 1666, pp. 96-109, 1992.
- [2] D. J. Lieberman and J. P. Allebach, “Efficient Model Based Halftoning Using Direct Binary Search,” *IEEE International Conference on Image Processing*, 1997.
- [3] 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, pp. 1950-1963, 2000.
- [4] J. P. Allebach, “DBS: retrospective and future directions,” *Proceedings of SPIE*, vol. 4300, pp. 358-376, 2001.
- [5] W. Jiang, W. Xi, U. Sarkar, R. Ulichney and J. P. Allebach, “Color halftoning based on Neugebauer Primary Area Coverage,” *Electronic Imaging, Color Imaging XXII: Displaying, Processing, Hard-*



Figure 15: 12.5 % of Yellow.

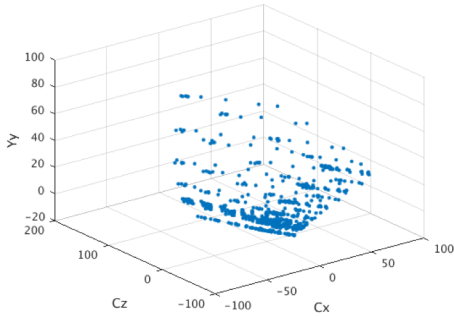


Figure 16: Measured grid points / inverse mapping. Each measured point has an associated NPAC.

copy and applications, vol. 21, no. 2, pp. 688-696, 2017.

- [6] 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*, pp. 427, 2018.
- [7] R. S. Gentile, E. Walowitz, and J. P. Allebach, "A Comparison of Techniques for Color Gamut Mismatch Compensation," *Journal of Imaging Technology*, vol. 1077, no. 5, pp. 342, 1990.
- [8] M. Wolski, J. P. Allebach and C. A. Bouman, "Gamut Mapping: Squeezing the Most out of Your Color System," *Color Imaging Conference: Color Science, System and Application*, pp. 89-92, 1994.
- [9] O. Rodrigues, "Des lois geometriques qui regissent les dplacements d' un systme solide dans l' espace, et de la variation des coordonnees provenant de ces dplacement considres independent des causes qui peuvent les produire," *The Journal de Mathematiques Pures et Appliques*, vol. 5, pp. 380-440, 1840.
- [10] T. Molleran and J. F. Hughes "Efficiently Building a Matrix to Rotate One Vector to Another," *Journal of Graphics Tools*, vol. 4, no. 4, 1999.
- [11] 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.
- [12] J. Morovic, P. Morovic and J. Arnabat, "HANS: Controlling ink-jet print attributes via Neugebauer primary area coverages," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 688-696, 2012.
- [13] P. Morovic, J. Morovic, J. Gondek, M. Gaubatz, and R. Ulichney, "PARAWACS Color halftoning with a single selector matrix," *Color Imaging Conference*, vol. 2016, no. 1, pp. 41-46, 2016.
- [14] P. Morovic, J. Morovic, J. Gondek, M. Gaubatz, and R. Ulichney, "Direct Pattern Control Halftoning of Neugebauer Primaries," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4404-4413, 2017.
- [15] J. Morovic, *Color Gamut Mapping*, Wiley, 2008.
- [16] W. Jang, M. Chen, J. P. Allebach and G. Chiu, "Print Quality Test

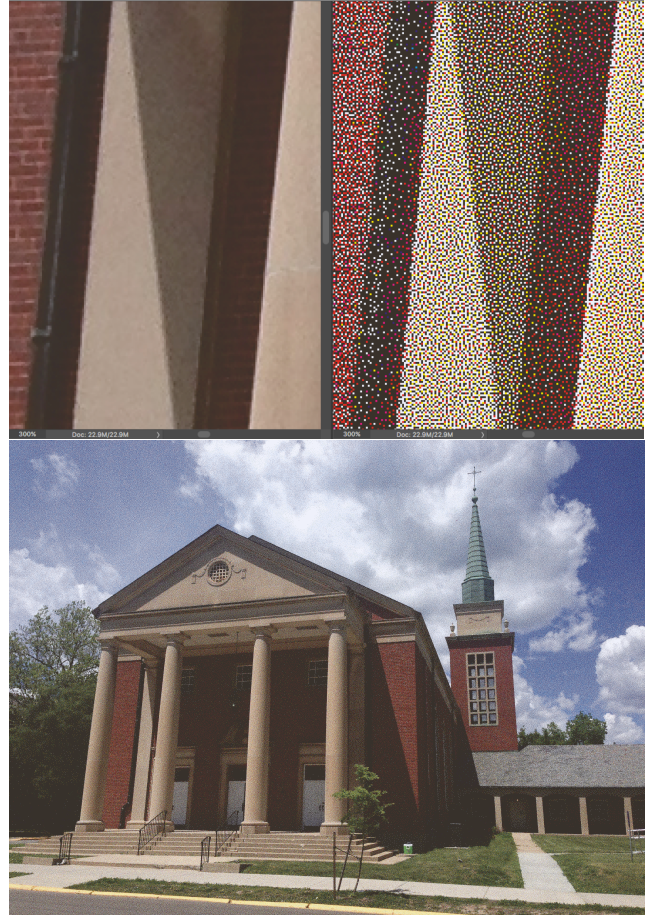


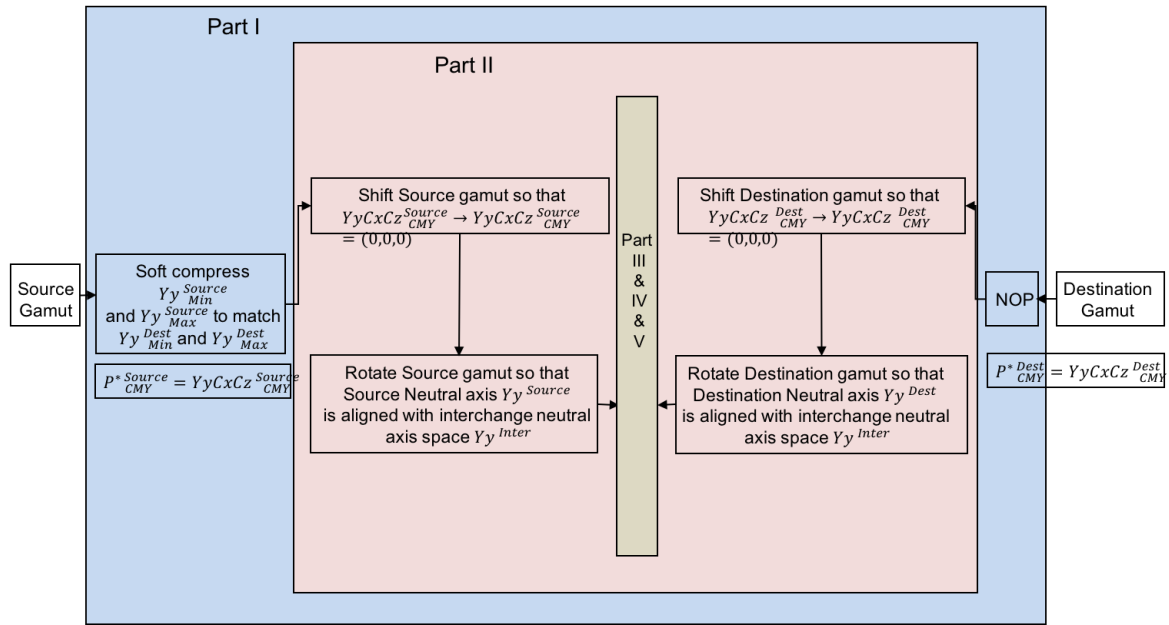
Figure 17: A detailed comparison between the gamut mapped image and the PARAWACS halftone image (up); and PARAWACS halftone image (down).

Page," *Journal of Imaging Science and Technology*, vol. 48, no. 5, pp. 688-696, 2004.

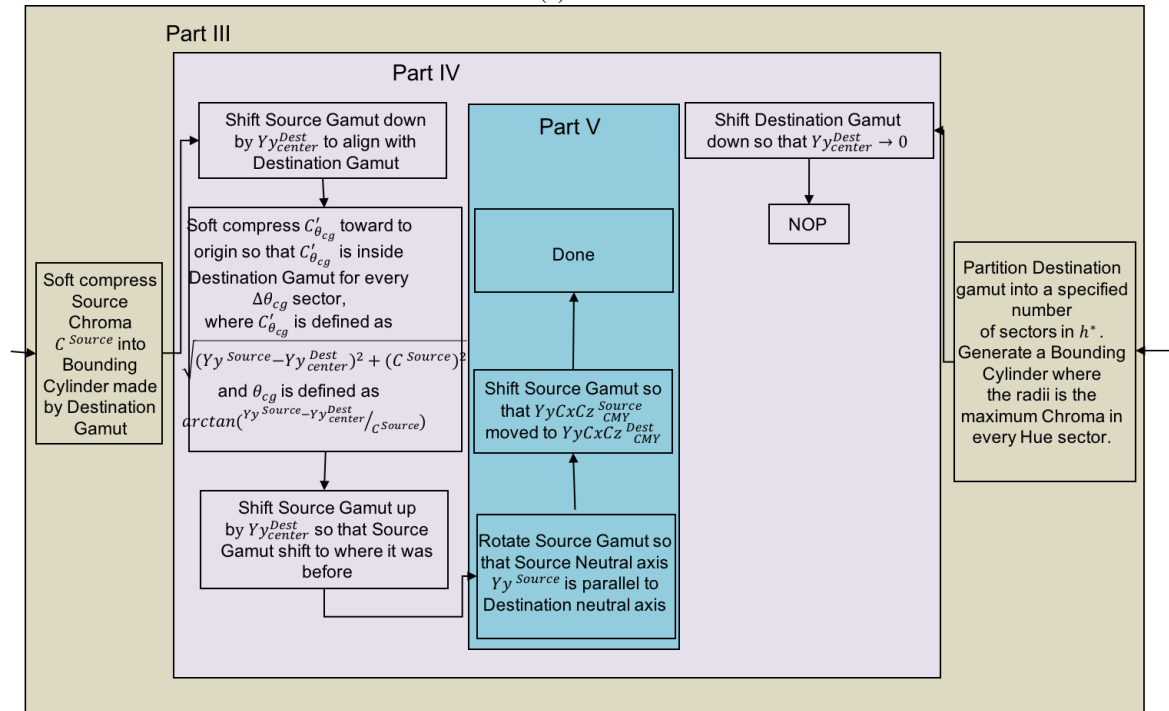
- [17] J. M. Kasson, W. Plouffe, and S. I. Nin, "Tetrahedral interpolation technique for color space conversion," *Proc. SPIE 1909, Device-Independent Color Imaging and Imaging Systems Integration*, vol. 26, no. 9, pp. 127-139, 1993.
- [18] R. Ulichney, *Digital Halftoning*, MIT Press, 1987.
- [19] MATLAB, MATLAB R2017b, *The MathWorks, Inc.*, Natick, Massachusetts, United States.

Author Biography

Jiayin Liu is a graduate student majoring electrical engineering at Purdue University. She received B.S. and M.S. from Purdue University in 2015 and 2018, respectively. Her research interests include color management, halftoning and image processing.



(a) Part I and II



(b) Part III through V

Figure 18: Gamut mapping procedures

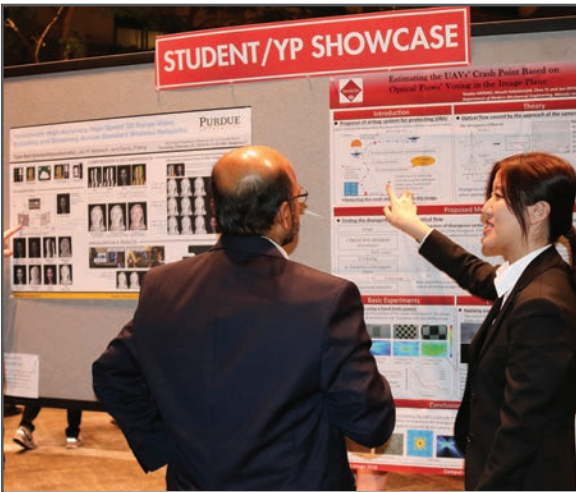
JOIN US AT THE NEXT EI!

IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!



- **SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •**
- **INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •**

www.electronicimaging.org

