https://doi.org/10.2352/ISSN.2470-1173.2019.14.COLOR-103

# Vector Multilevel Tone-Dependent Error Diffusion in the YyCxCz Color Space

Chin-Ning Chen, Zhen Luan, and Jan P. Allebach
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA

## Abstract

*In this paper, we present a tone-dependent fast error diffusion algorithm for color images, in which the quantizer is based on a simulated linearized printer space and the filter weight function depends on the ratio of the luminance of the current pixel to the maximum luminance value. The pixels are processed according to a serpentine scan instead of the classic raster scan. We compare the results of our algorithm to those achieved using the fixed Floyd-Steinberg weights and processing the image according to a raster scan ordering.*

## 1. Introduction

Halftoning remains a critical aspect of contemporary printing devices. These devices increasingly have multilevel capabilities. We develop a vector error diffusion solution that operates in the YyCxCz linearzed uniform color space [9]. Our results are based on a simulated CMY printer that uses inks from an Indigo 7000 press. We assume a 16-level output for each of the C, M, Y channels; so each output pixel consists of a 4 bit word.

The classic error diffusion proposed by Floyd and Steinberg [1] is an algorithm that generates a binary image by processing the continuous-tone image with neighborhood operations moving through the image in raster scan order, quantizing each pixel in the scan line, and diffusing the error ahead to the neighboring pixels. However, this algorithm also generates worm-like artifacts and some visible structure. There are many papers proposing to solve these problems. In this paper, we are not going to mention how the artifact problems are solved, but focus how the error diffusion is used in a multilevel color context, rather than a monochrome binary context.

The previous works focusing on mutilevel halftoning and color halftoning are [2–6]. Monga et al. [3] and Yu et al. [6] both use a Neugebauer printer model. In [3], the Neugebauer model was used in the middle step by transforming from CMY to YyCxCz first, next to predict the colorimetric response of the printer and then further to train the error diffusion filter to minimize the error metric. In [6], the authors utilized the Neugebauer model in RGB color space by using trilinear interpolation based on Neugebauer primaries to represent an arbitrary input color. Our method operates with gamut mapping [7] [8] first and then quantizes the input pixel based on sampling the Neugebauer printer model. Both the gamut mapping and the quantization operate in the YyCxCz color space.

Figure 1 shows our system pipeline. The input is an sRGB image; and we transform it from sRGB to CIE XYZ [10] to YyCxCz standing for our source gamut. Also, we are given the destination gamut, which is the Indigo gamut; and then we do the gamut mapping to fit the source gamut into the destination gamut. After gamut mapping, we do tone-dependent fast error diffusion (TDFED), which is also in the YyCxCz color space. Finally, we transform the result of TDFED from YyCxCz back to sRGB to display it.

The rest of this paper is organized as follows. Section 2 outlines the process of gamut mapping. Section 3 first introduce the YyCxCz color space and our printer output space, and then nail into our TDFED algorithm. Also, we will compare our TDFED with classic Floyd-Steinberg Error Diffusion (FSED). Section 4 shows and compares the resulting images after gamut mapping, TDFED, and FSED. Finally, Section 6 concludes this paper.

## 2. Gamut Mapping

We want to generate a mapping so that given an sRGB value, we can find its corresponding printable value based on our printer output space. In our system, we are given the destination gamut from the Indigo 7000 press. A brief description of overall process can be split into five part:

Part 1 Soft compress the source lightness from input image to match the destination lightness.

Part 2 Let the source neutral axis and destination neutral axis to align with the Yy axis.

Part 3 Soft compress the source chroma to fit it into the bounding cylinder constructed by the destination gamut. Unfortunately, so far, the source gamut is not actually fit into the destination gamut at every hue angle.

Part 4 We do central compression for lightness and chroma on source gamut to let it actually fit into the destination gamut within each hue sector.

Part 5 Finally, we rotate and shift to move the source gamut to the destination gamut.

The details of the gamut mapping can be seen in Figure 2. Also, refer to [7, 8] .

## 3. Tone-Dependent Fast Error Diffusion

Our motivation to use tone-dependent fast error diffusion (TDFED) is based on the comparison between Floyd-Steinberg algorithm and TDFED in binary image, as shown in [2]. There are obvious artifacts in the sky in the figure generated by Floyd-Steinberg but the figure generated by TDFED does not have this

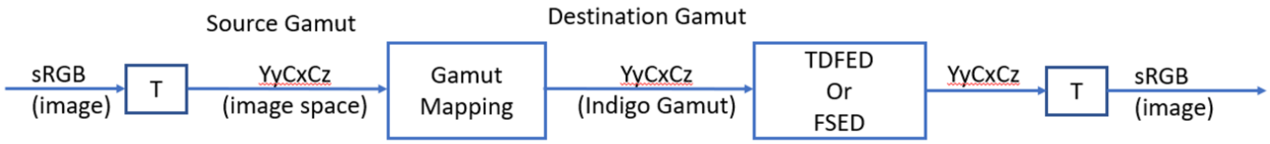


Figure 1: System pipeline

problem. As a result, we try to use TDFED system on the color image.

## YyCxCz Color Space

Before we go to the details of TDFED, here we explain the relation between different color spaces and the reason why we operate in the YyCxCz color space.

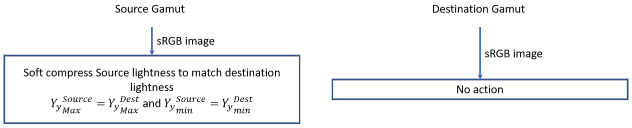
The YyCxCz color space is the linearized CIE Lab color space which can be obtained from the CIE XYZ color space, as shown in Equations (1), (2), (3)

$$Yy = 116\frac{Y}{Y_n} - 16 \tag{1}$$

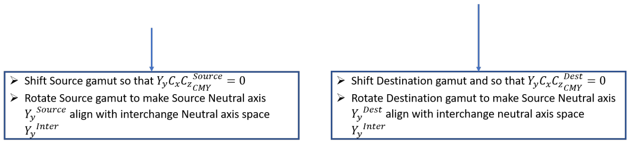
$$Cx = 500\left(\frac{X}{X_n} - \frac{Y}{Y_n}\right) \tag{2}$$

$$Cz = 200\left(\frac{Y}{Y_n} - \frac{Z}{Z_n}\right) \tag{3}$$

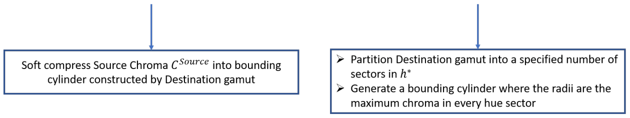
The Yy value represents the luminance, and the Cx and Cz values are the red-green and blue-yellow chrominance components, respectively. Because YyCxCz is a linearized transformation, it can overcome the distortion problem mentioned in [9]



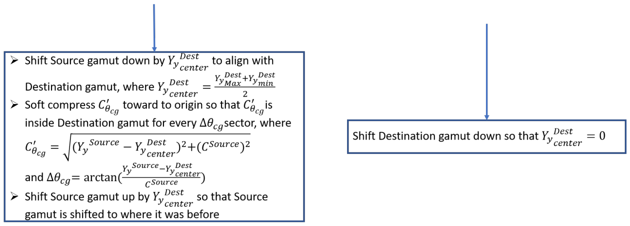
(a) Part 1



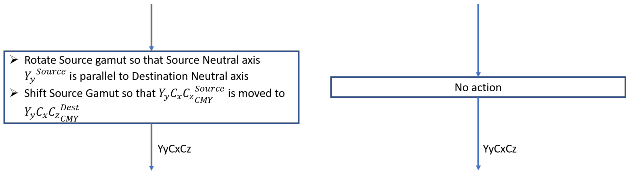
(b) Part 2



(c) Part 3



(d) Part 4



(e) Part 5

Figure 2: Process of gamut mapping. Left hand side is the action for source gamut and the right hand side action is for destination gamut. (a) Part 1 (b) Part 2 (c) Part 3 (d) Part 4 (e) Part 5

## Printer Output Space

We assume a CMY printer with 16 levels per output channel, i.e., 12 bits/pixel. We simulate the printer output space by uniformly interpolating the space defined by the measured 8 Neugebauer Primaries (NPs) which are: W, Y, C, CY, M, MY, CM, CMY. The following steps describe how we simulate our printer output space: First, we choose our target output device, which is the Indigo 7000 press. Second, we print a constant-tone patch for each of the 8 NPs. Third, we measure their CIE XYZ value by using hardware X-rite DTP70 and then convert the obtained XYZ value into YyCxCz value. Next, we uniformly sample 16 × 16 × 16 points along each edge of the CMY color cube as shown in Figure 3. Finally, we use tetrahedral interpolation to find their corresponding YyCxCz values as shown in Figure 4.

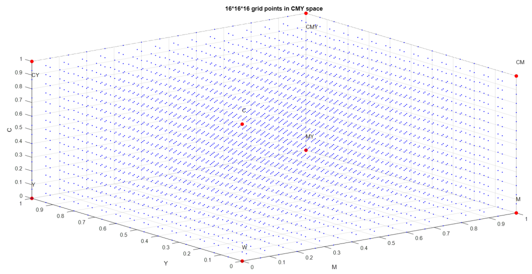


Figure 3: Sampled points in CMY color space

## Tone-Dependent Fast Error Diffusion

Figure 5 illustrate the block diagram of the TDFED system. In this system, $\vec{f}[m,n]$ is the input image vector, $\tilde{\vec{f}}[m,n]$ is the

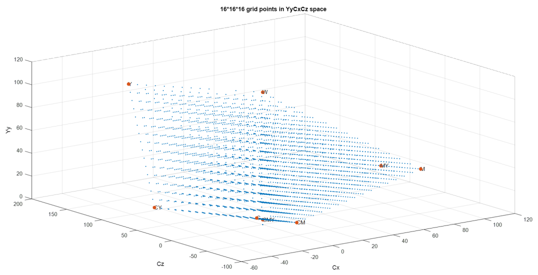


Figure 4: Sampled points in YyCxCz color space

modified input, and $\vec{g}[m,n]$ is the YyCxCz image vector quantized according to Equation (4) where the set the $C$ is the $16 \times 16 \times 16$ set of Neugebauer Primaries for the output device. This means that the current pixel will be replaced by the one of the sampled Neugebauer Primary with the shortest between them. After obtaining $\vec{g}[m,n]$, we can calculate the error vector according to Equation (5). Here, the weight function $W(Y_y)$ depends on the position of the pixel to which the error is being diffused relative to that of the pixel being processed and the ratio of the Yy value of the current pixel to the maxmimum Yy value. Plots of the weight function are shown in Figure 6, where the star sign represents the current pixel. Moreover, the processing order of TDFED is according to a serpentine scan.

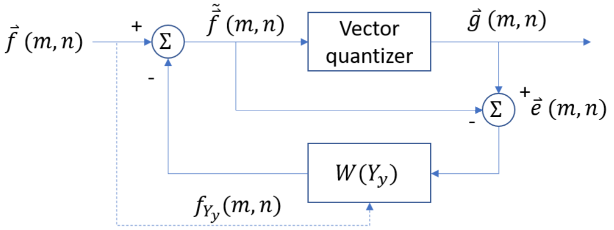


Figure 5: Tone-dependent fast error diffusion system

$$\vec{g}[m,n] = \arg\min_{c\in C}\{\|c - \tilde{\tilde{\vec{f}}}[m,n]\|\} \tag{4}$$

$$\vec{e}[m,n] = \vec{g}[m,n] - \tilde{\tilde{\vec{f}}}[m,n] \tag{5}$$

### *Comparison with Floyd-Steinberg Error Diffusion*

Figure 7 illustrates the block diagram of the FSED system. The overall system is similar with TDFED but there are two differences. First, the weight filter $H$ in FSED is the fixed set of numbers where w1 is equal to $\frac{7}{16}$, w2 is equal to $\frac{3}{16}$, w3 is equal to $\frac{5}{16}$, and w4 is equal to $\frac{1}{16}$. In the TDFED system, the weight filter $W(Y_y)$ is a function. Second, the processing order of FSED is according to raster scan instead of the serpentine scan in the TDFED system. In the experimental result, the comparisons are between FSED with raster scan and TDFED with serpentine scan.

## 4. Experimental Result

Figure 8 shows the comparison of the original image, the gamut mapped image, the TDFED image, and the FSED image. For the gamut mapped image, it is a little bit brighter than the original image. But overall, it looks good. The color rendering results of the TDFED image and the FSED image look very similar at this scale. So we need to zoom in to see the detail of the halftone images. Figure 9 shows part of the figure in larger scale to have the big picture of what is generated by halftoning. Figures 10 to 12 shows the image detail in pixel scale. In FSED image, there are obvious checkerboard and texture cliques artifact. But in some case as shown in Figure 12, FSED performs better than TDFED. However, the overall image quality generated by TDFED is better than that of FSED.

| | | |
|---|---|---|
| | * | w1 |
| w2 | w3 | w4 |

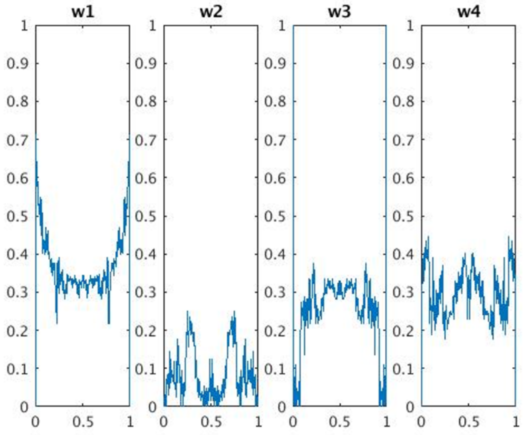


Figure 6: weight function for TDFED

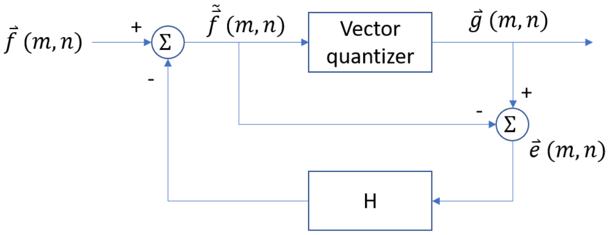
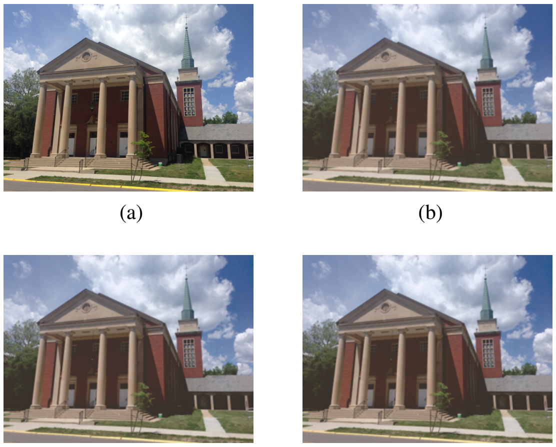


Figure 7: Floyd-Steinberg error diffusion system

## 5.Conclusion

We have developed a halftoning solution for a simulated multilevel CMY printer with 16 levels per channel. Our proposed imaging pipeline consists of: First, transform input sRGB to YyCxCz linearized uniform color space. Next, we do the gamut mapping to fit the source gamut into the printer gamut also in the YyCxCz color space. Finally, we do vector error diffusion that maps modified continuous-tone image values to the nearest output color on a $16 \times 16 \times 16$ grid. It yields good quality output images but the improvement over FSED provided by TDFED is not as significant as it is for a binary output device.



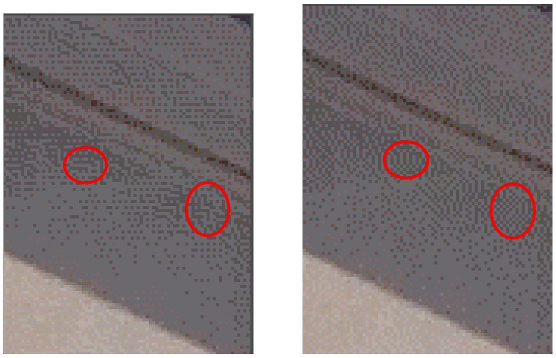
(a) (b)

(c) (d)

Figure 8: Comparison of resulting image at three stages in the system pipeline. (a) Original image (b) Gamut mapped image (c) TDFED image (d) FSED image

(a) (b)

Figure 9: The result zoom in for Figure 8 (c) and (d) respectively: (a) TDFED image (b) FSED image.



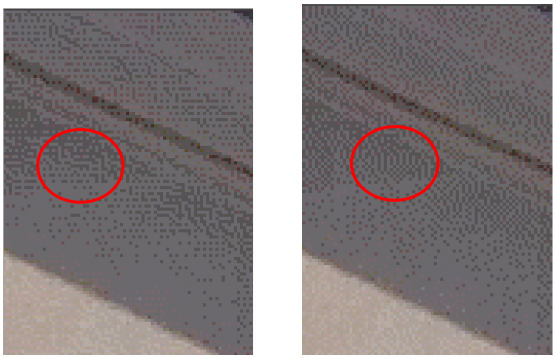
(a) (b)

Figure 10: Comparison between (a) TDFED image and (b) FSED image. We can see the obvious checkerboard artifact in FSED.

## References

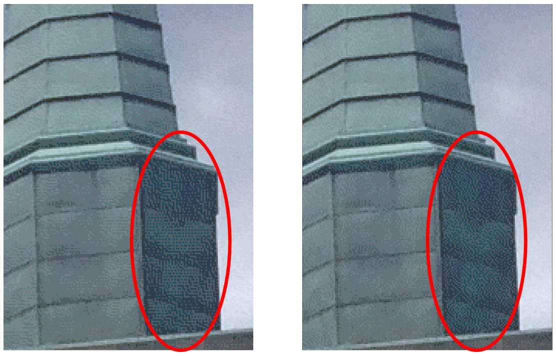
[1] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," Proc. Soc. Image Display, vol. 17, 1976.

[2] P. Li and J. P. Allebach, "Tone-dependent error diffusion," in IEEE Transactions on Image Processing, vol. 13, no. 2, pp. 201-215, Feb. 2004.



(a) (b)

Figure 11: Comparison between (a) TDFED image and (b) FSED image. We can see the texture cliques artifact in FSED.



(a) (b)

Figure 12: Comparison between (a) TDFED image and (b) FSED image. We can see that in this case, FSED performs better than TDFED.

[3] V. Monga, N. Damera-Venkata and B. L. Evans, "Design of Tone-Dependent Color-Error Diffusion Halftoning Systems," in IEEE Transactions on Image Processing, vol. 16, no. 1, pp. 198-211, Jan. 2007.

[4] G. Sarailidis and I. Katsavounidis, "A Multiscale Error Diffusion Technique for Digital Multitoning," in IEEE Transactions on Image Processing, vol. 21, no. 5, pp. 2693-2705, May 2012.

[5] J. Guo, J. Chang, Y. Liu, G. Lai and J. Lee, "Tone-Replacement Error Diffusion for Multitoning," in IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 4312-4321, Nov. 2015.

[6] H. Yu, K. Inoue, K. Hara, and K. Urahama, "Color Error Diffusion Based on Neugebauer Model," in IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E99.A, no. 9, pp. 1758-1761, Sept. 2016.

[7] M. Wolski, J. P. Allebach, and C. A. Bouman, "Gamut Mapping Squeezing the Most Out of Your Color System", in Color and Imaging Conference, vol. 1994, no. 1, pp. 89-92, 1994.

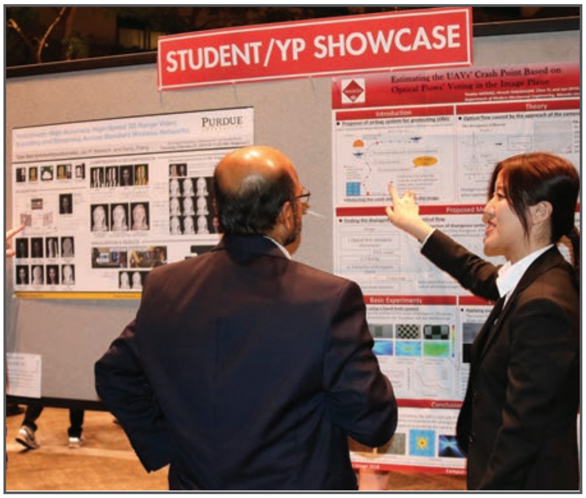
[8] R. S. Gentile, J. P. Allebach, and E. Walowit, "A Comparison of Techniques for Color Gamut Mismatch Compensation," in Human Vision, Visual Processing, and Digital Display, vol. 1077, pp. 342-355, 1989.

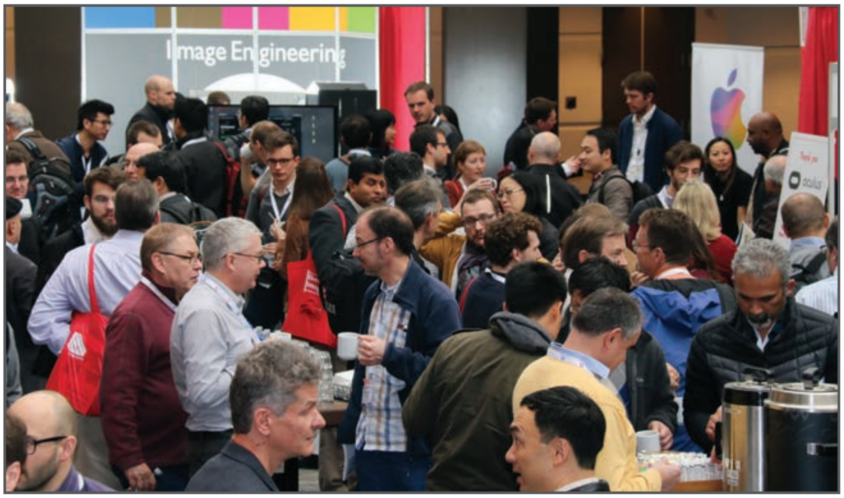
[9] T. J. Flohr, B. W. Kolpatzik, R. Balasubramanian, D. A. Carrara, C. A. Bounam, and J. P. Allebach, "Model-Based Color Image Quantization," in Proc.SPIE, vol. 1913, 1993.

[10] G. Sharma, Digital Color Imaging Handbook. Boca Raton, FL: CRC, 2002.

## Author Biography

*Chin-Ning Chen received both her BS and MS in electrical and computer engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2014 and 2016, respectively. Currently, she is a Ph.D. student in Purdue University, West Lafayette, IN, where she has been since 2016. Her research interests include color management and halftoning.*





AR is a fundamental and transformative interface and platform.

20+ years from my initial working demo to today's AR displays

Optimistic that it won't take another 20 to reach ubiquitous consumer AR!