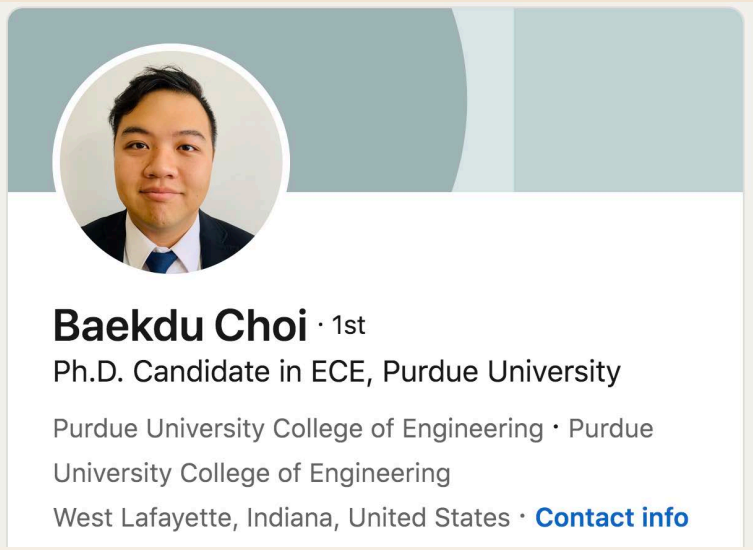
Developing an inkjet printer II :

CMY ink amounts to multibit CMY halftones

Baekdu Choi[1], Daulet Kenzhebalin[1], Sige Hu[1], George Chiu[1], Davi He[2], Zillion Lin[2], and Jan Allebach[1]

[1]Purdue University and [2]Sunvalleytek International Inc. (United States)

30 January 2020



SUNVALLEY

PURDUE UNIVERSITY

Overview

- **Motivation**
- 2-drop error diffusion
- Dot-off-dot 2-drop error diffusion
- DBS screen blending-in
- Summary & Conclusions
- References

B. Choi, D. Kenzhebalin, S. Hu, G. Chiu, Z. Lin, D. He, and J. Allebach, “Developing an inkjet printer II: CMY ink amounts to multibit CMY halftones,” Color Imaging XXV: Displaying, Processing, Hardcopy, and Applications, (Part of IS&T Electronic Imaging 2020), R. Eschbach, G. Marcu, and A. Rizzi, Eds., Burlingame, CA, 26 January - 30 January 2020.

Motivation

- When printing images on an object, prints with high saturation are generally preferred
- Multi-drop printing gives us a possibility of exploring a wider printer gamut including printable colors with more saturation, compared to typical single-drop printing
- Our goal is to develop a halftoning algorithm for 2-drop printing that generates good quality halftone images without visible artifacts

Overview

- Motivation
- **2-drop error diffusion**
- Dot-off-dot 2-drop error diffusion
- DBS screen blending-in
- Summary & Conclusions
- References

2-drop error diffusion

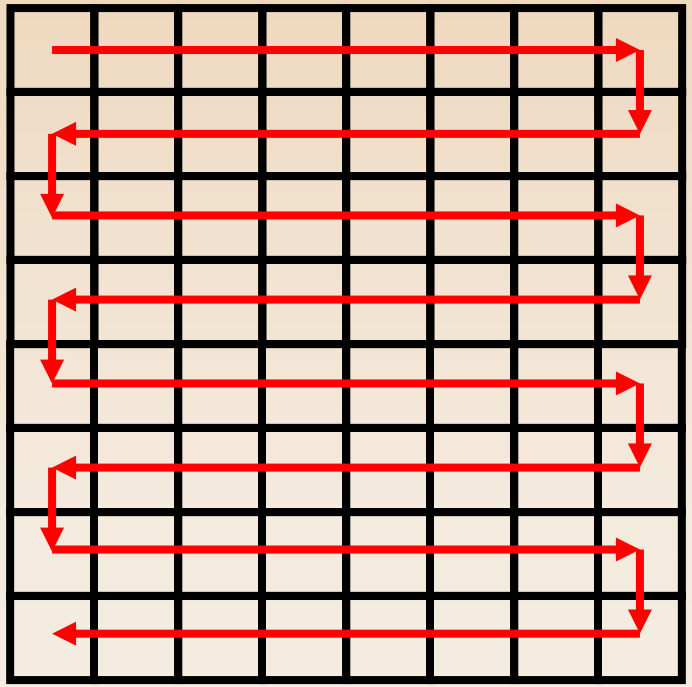
- The 2-drop error diffusion is based on the Floyd-Steinberg error diffusion algorithm, first introduced in [1]
- We keep updating an image called *error diffused image*, which is initialized as a copy of the original input image
- In the halftone image, we represent 2 drops of colorant with value 1, 1 drop of colorant with value 1/2, and no colorant with value 0
- For each pixel in the error diffused image, if each color plane's absorptance value at the pixel is over 1/3, we place 1 drop, and if it is over 2/3, we place 2 drop of the colorant
- Then we diffuse the error to nearby pixel of the error diffused image using the Floyd-Steinberg error diffusion filter
- After that, we move to the next pixel and iterate the procedure

	X	7/16
3/16	5/16	1/16

Floyd-Steinberg error diffusion filter (X : current pixel)

Serpentine raster scan

- Serpentine raster scan is used as an attempt to reduce halftone artifacts ([2], pp. 266)
- When performing a serpentine raster scan, an image is first scanned in the first row from the left to the right, and as we proceed to the next row, the scan direction is alternated



Serpentine raster scan direction

Error diffusion : illustration

- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located

Error diffused image

0.4	0.8	0.6
0.3	0	0.3
0.7	0.1	0.4

Error diffusion : illustration

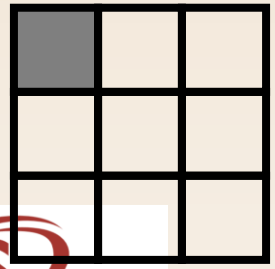
- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located

Error diffused image

0.4	0.8	0.6
0.3	0	0.3
0.7	0.1	0.4

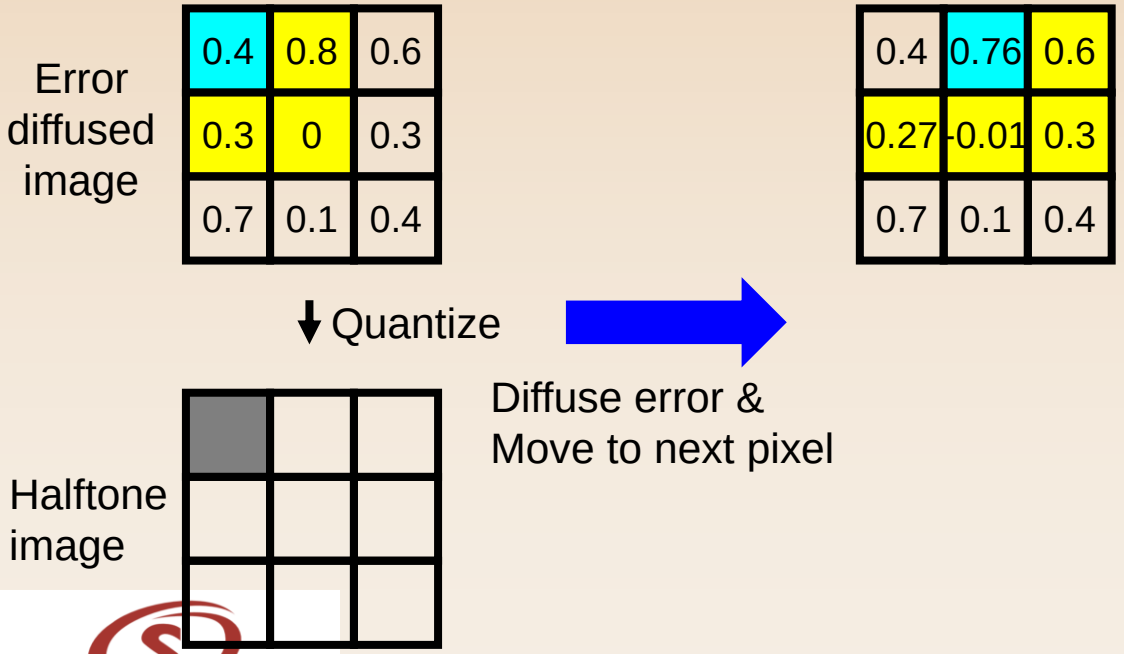
↓ Quantize

Halftone image



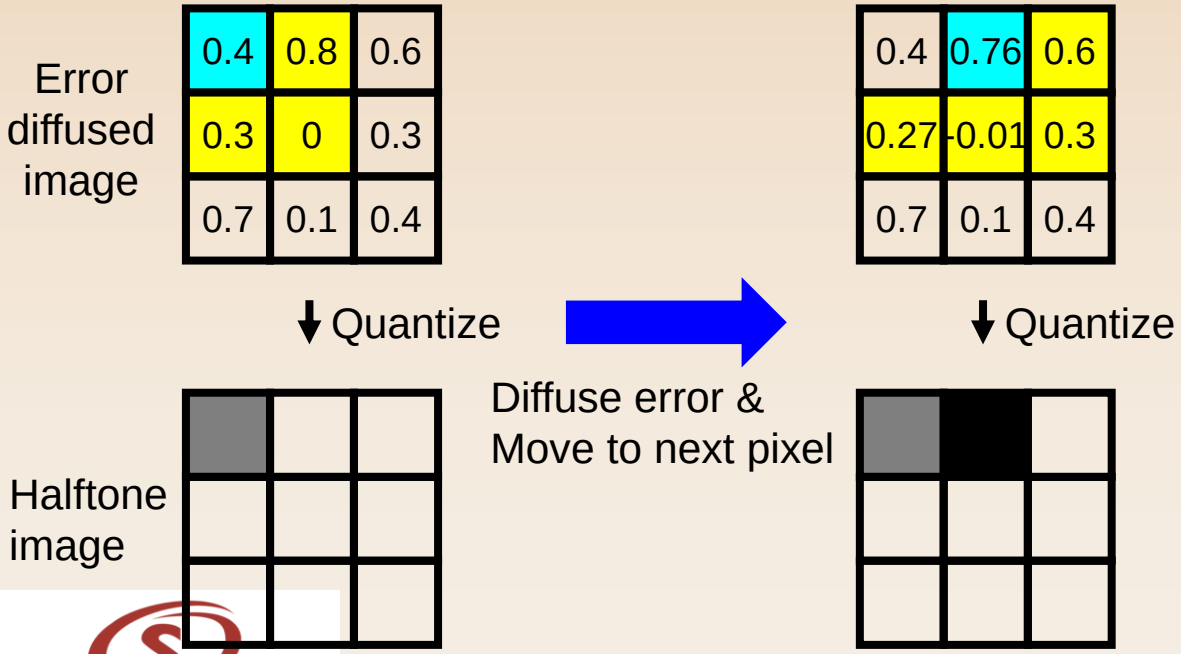
Error diffusion : illustration

- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located



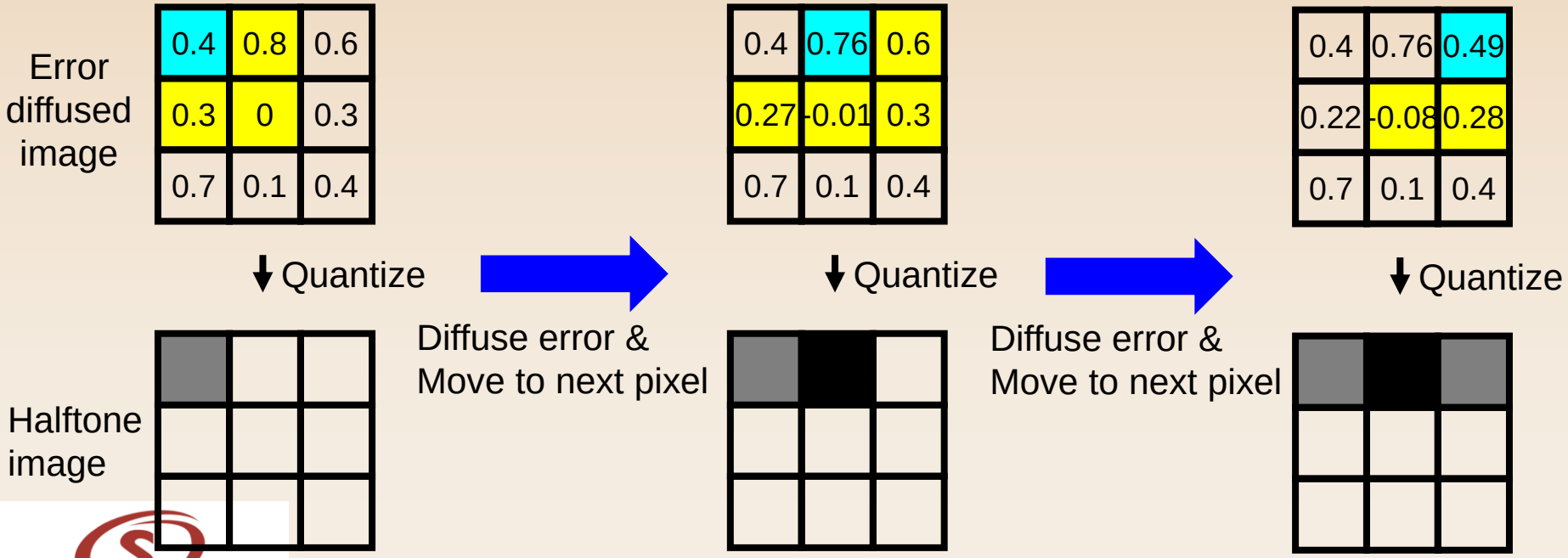
Error diffusion : illustration

- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located



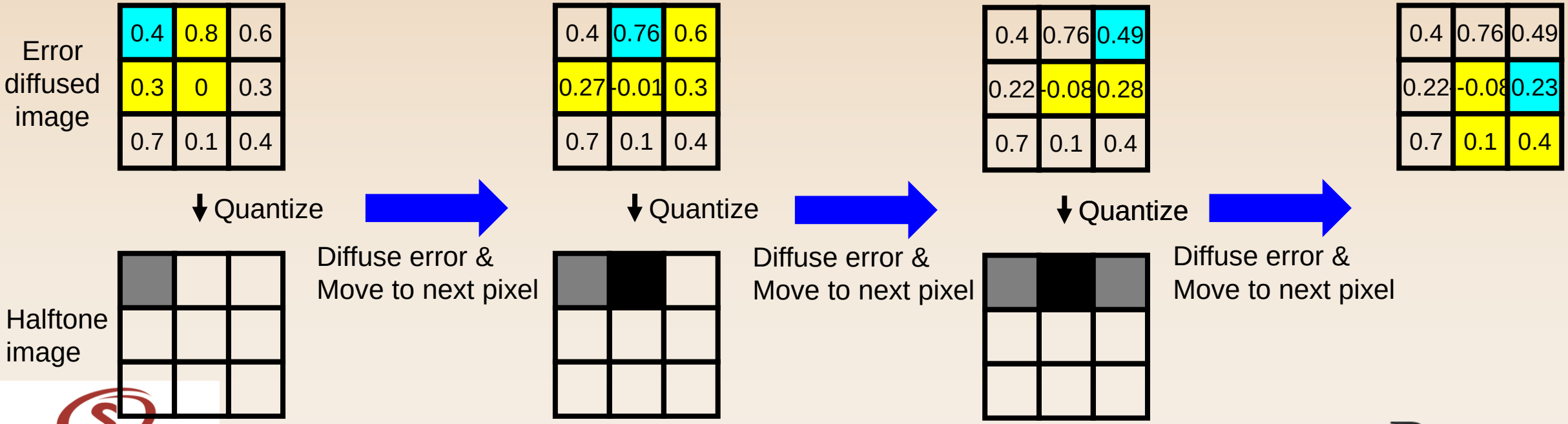
Error diffusion : illustration

- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located



Error diffusion : illustration

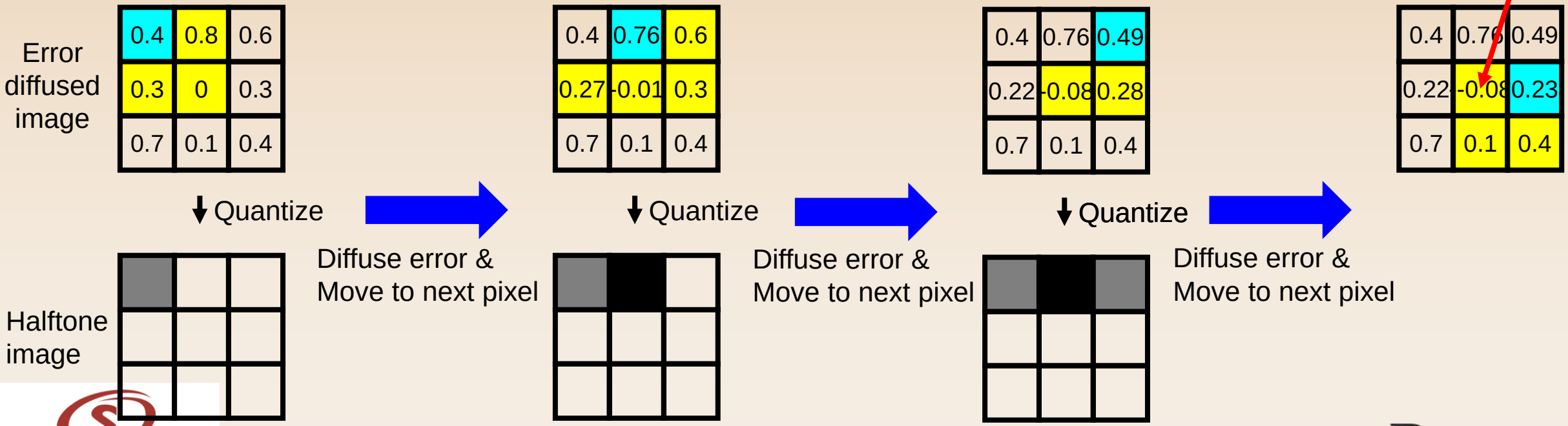
- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located



Error diffusion : illustration

- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located

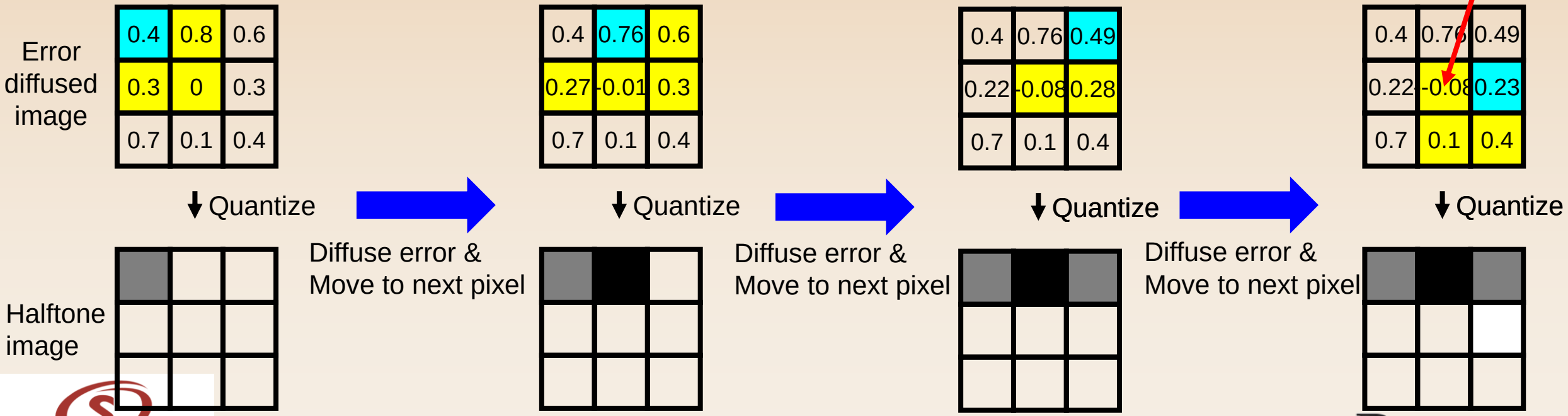
Note that due to the scan direction changing from right to left, the filter is also mirrored, thus this pixel will be updated



Error diffusion : illustration

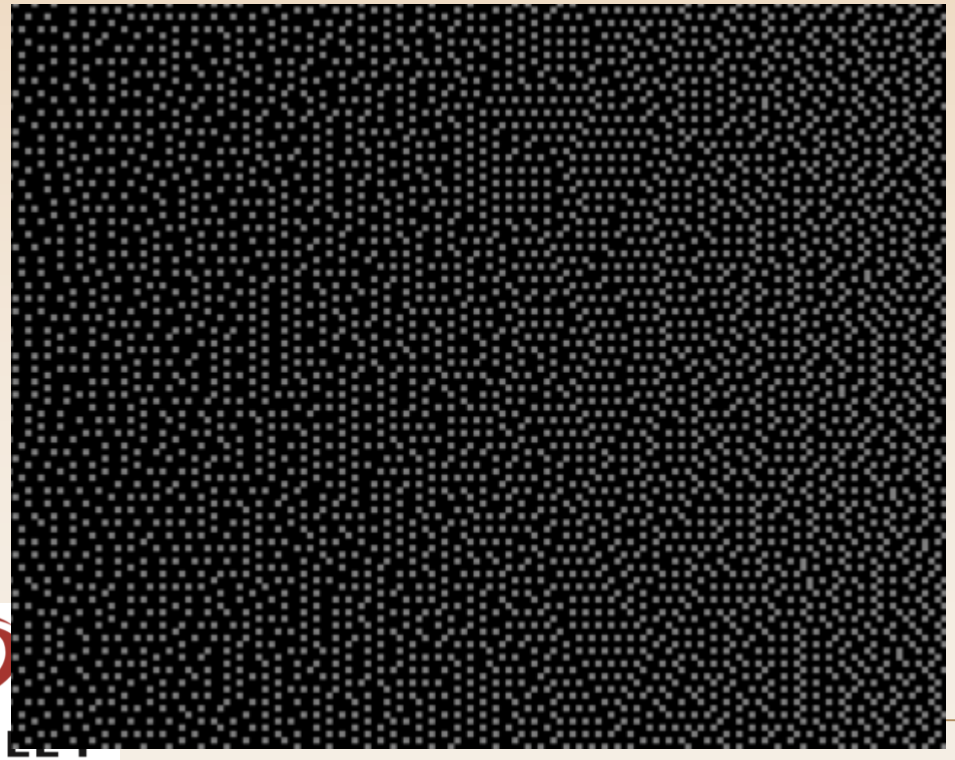
- Note that for this error diffusion algorithm, each color plane is processed independently
- In the error diffused image, cyan indicates the current pixel being processed and yellow indicates pixels where error will be diffused
- In the halftone image, white indicates no colorant, gray indicates 1 drop of the colorant, whereas black indicates 2 drops of the colorant has been located

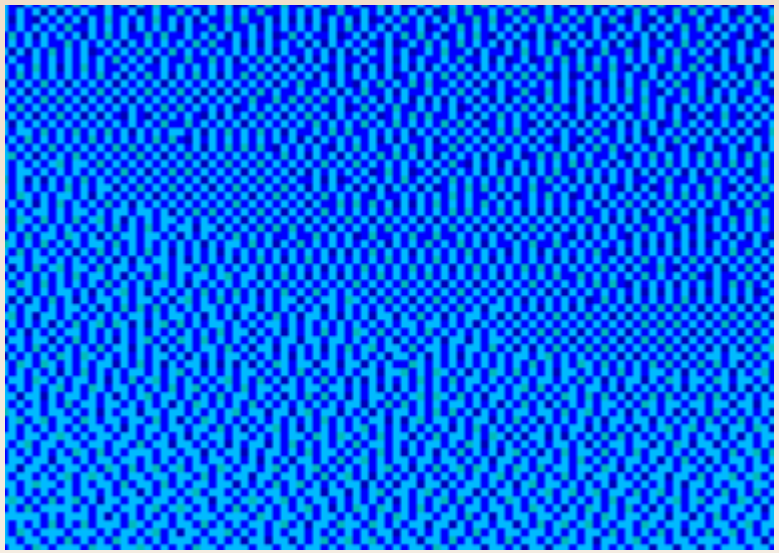
Note that due to the scan direction changing from right to left, the filter is also mirrored, thus this pixel will be updated



2-drop error diffusion : halftone artifacts

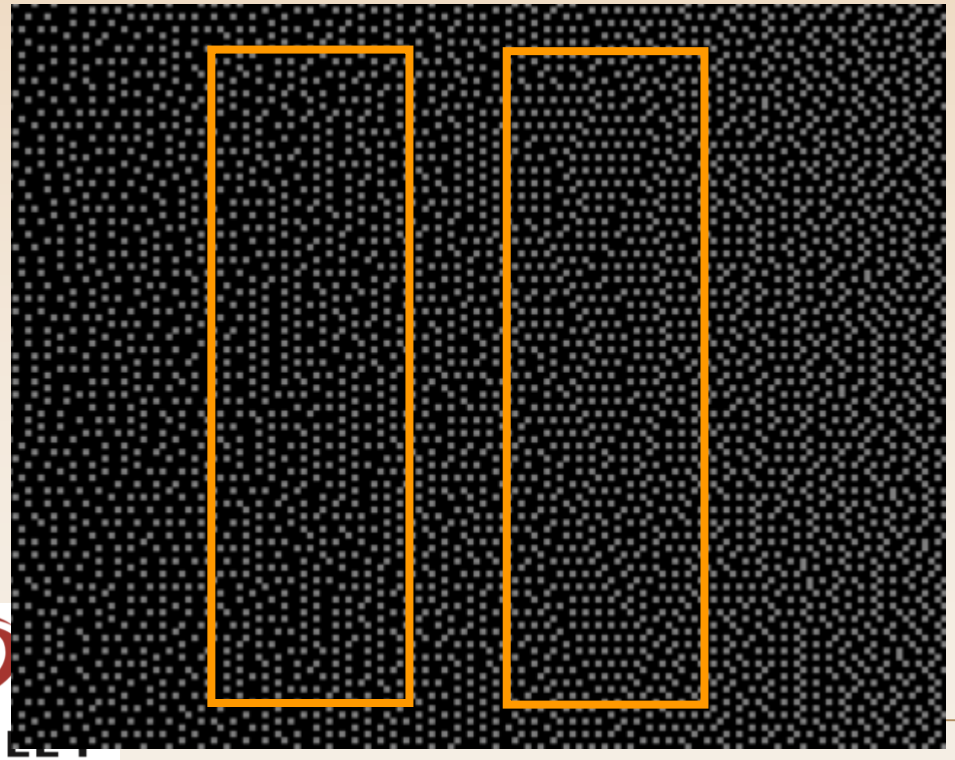
- Unfortunately since it is merely an extension of the traditional Floyd-Steinberg error diffusion, the 2-drop error diffusion algorithm delivers halftone images with severe artifact issues
- While using serpentine raster scan somewhat helps reducing artifacts, it is known to have a limit to its effect, especially with vertical artifacts ([3], pp. 374)
- Both images below are results of halftone images generated with the 2-drop error diffusion algorithm

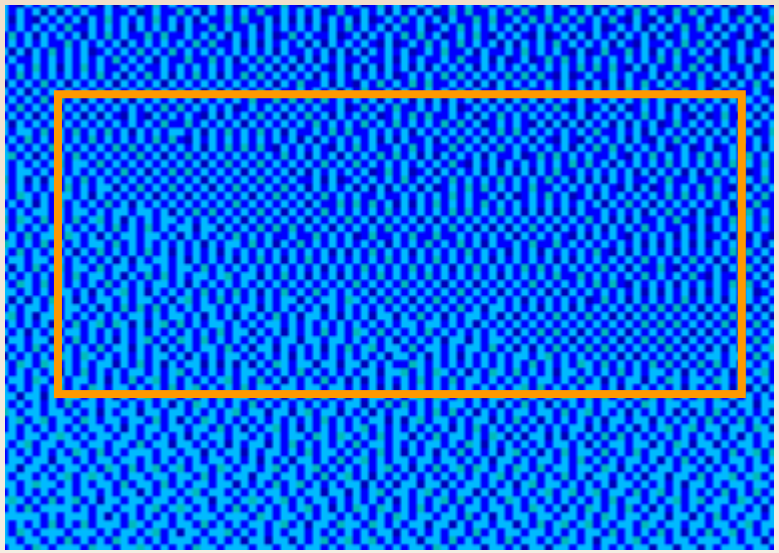




2-drop error diffusion : halftone artifacts

- Unfortunately since it is merely an extension of the traditional Floyd-Steinberg error diffusion, the 2-drop error diffusion algorithm delivers halftone images with severe artifact issues
- While using serpentine raster scan somewhat helps reducing artifacts, it is known to have a limit to its effect, especially with vertical artifacts ([3], pp. 374)
- Both images below are results of halftone images generated with the 2-drop error diffusion algorithm





Overview

- Motivation
- 2-drop error diffusion
- **Dot-off-dot 2-drop error diffusion**
- DBS screen blending-in
- Summary & Conclusions
- References

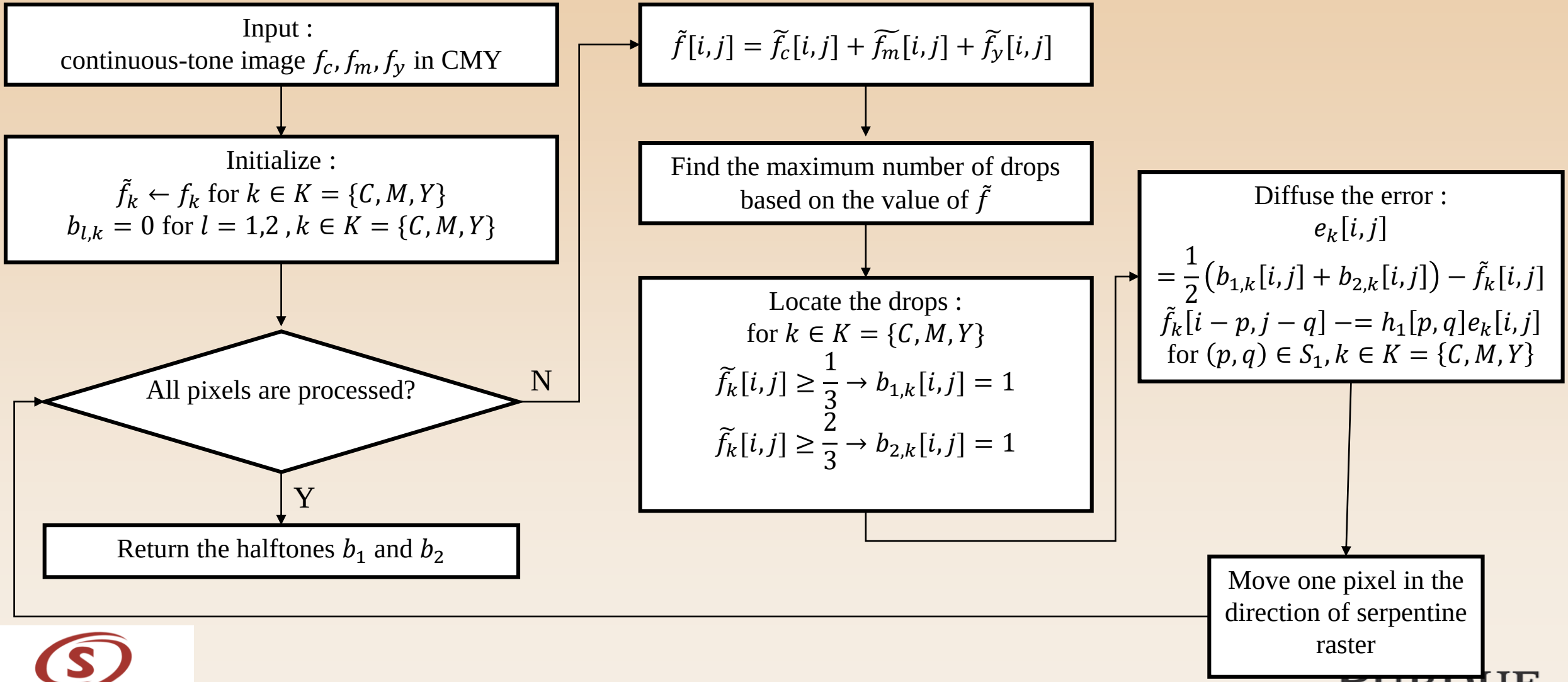
Dot-off-dot halftoning : motivation

- Previous slides' 2-drop error diffusion algorithm processes each color plane in CMY independently
- It is more desirable to locate the drops on different pixel locations than to locate on top of each other, but independent processing of color planes do not give us a measure to perform dot-off-dot halftoning
- Dot-off-dot could give us the following benefits :
  - It could result in using less ink compared to overlapping the colorant drops, since it reduces interaction between colorant drops. Similar effect is discussed in [4]
  - It could result in halftone images with smoother texture, since we reduce overlapping dots

Dot-off-dot halftoning : implementation

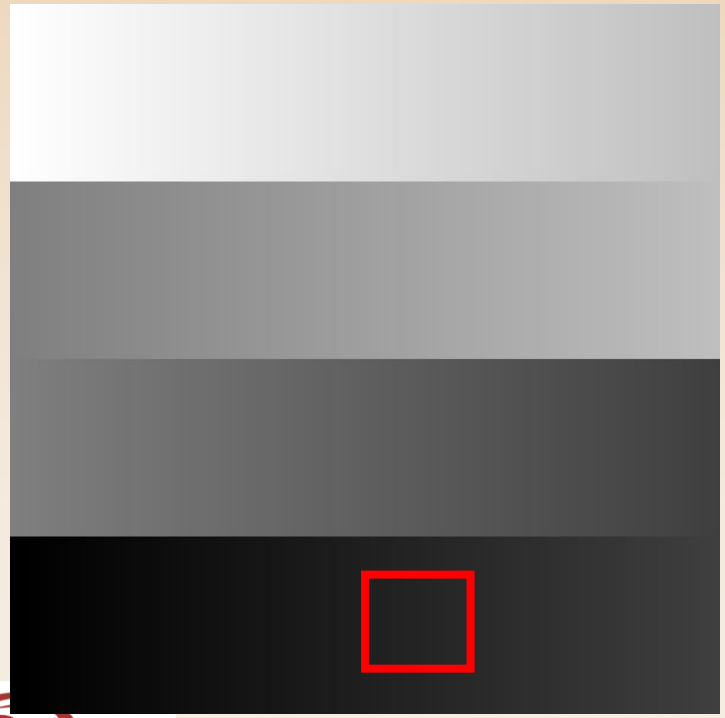
- We propose a way to perform dot-off-dot 2-drop halftoning by limiting the number of all colorant drops possible for each pixel
- First the algorithm looks at the sum of the color planes at the current pixel, and then based on the sum it determines the maximum number of all colorant drops to locate at the pixel
- After that, each color plane is quantized prioritizing the color plane with the largest absorptance value at the current pixel location of the error diffused image
- After locating the colorant drops for each colorant at the current pixel, error diffusion is performed in the same way as with the previous 2-drop halftoning, but limiting the number of drops results in dot-off-dot halftoning

Dot-off-dot halftoning : block diagram

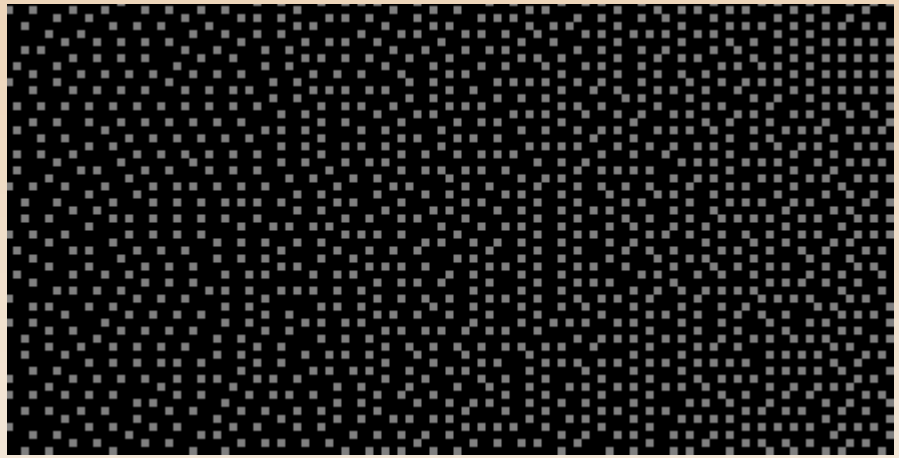


Dot-off-dot halftoning : results & limits

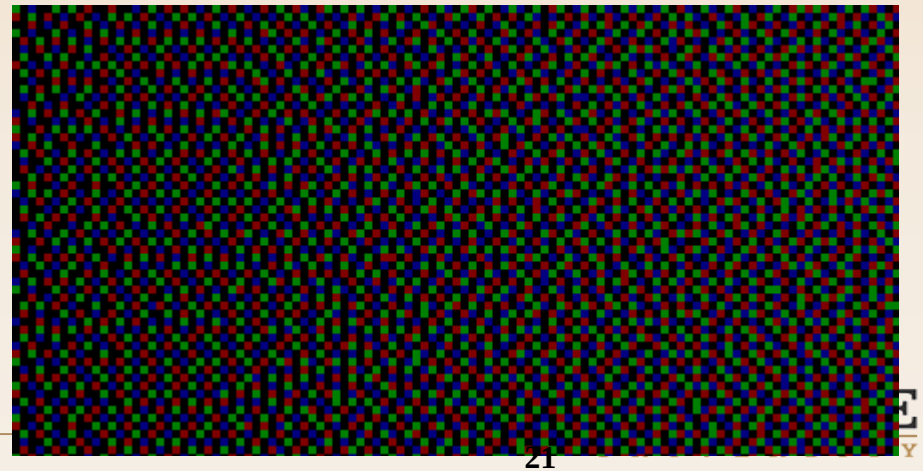
- Comparing the two halftone images below, we can see that dot-off-dot error diffusion tends to generate better quality halftone image
- Specifically, the artifacts observed from the simple multidrop error diffusion halftone images are somewhat alleviated
- We can also see that dot-off-dot halftoning cannot remove all halftone artifacts



Halftoned using simple 2-drop error diffusion, magnified



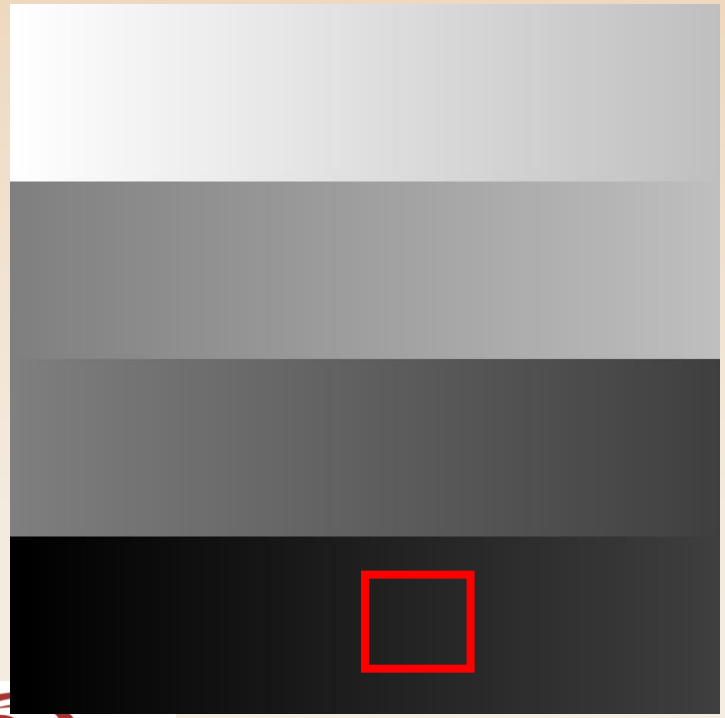
Halftoned using dot-off-dot error diffusion, magnified



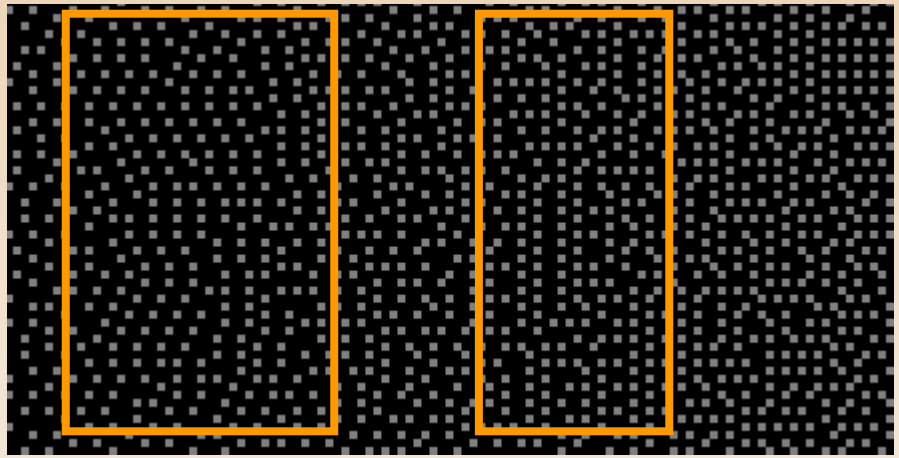
SUNVALLEY

Dot-off-dot halftoning : results & limits

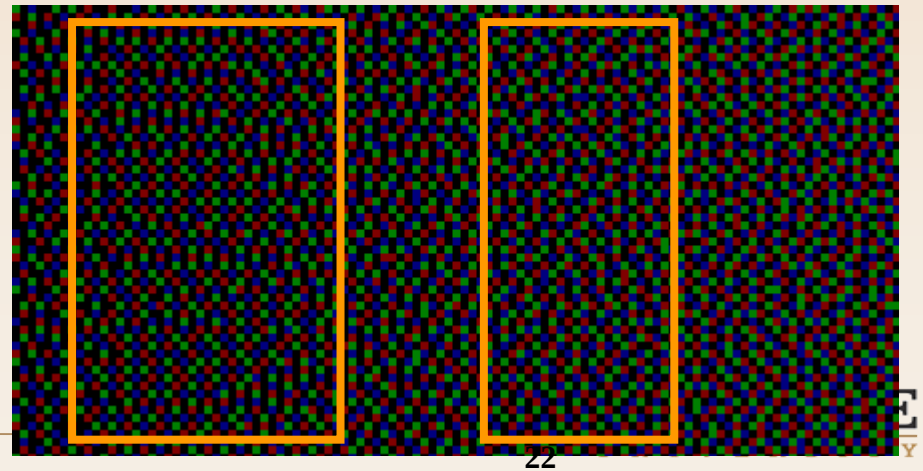
- Comparing the two halftone images below, we can see that dot-off-dot error diffusion tends to generate better quality halftone image
- Specifically, the artifacts observed from the simple multidrop error diffusion halftone images are somewhat alleviated
- We can also see that dot-off-dot halftoning cannot remove all halftone artifacts



Halftoned using simple 2-drop error diffusion, magnified



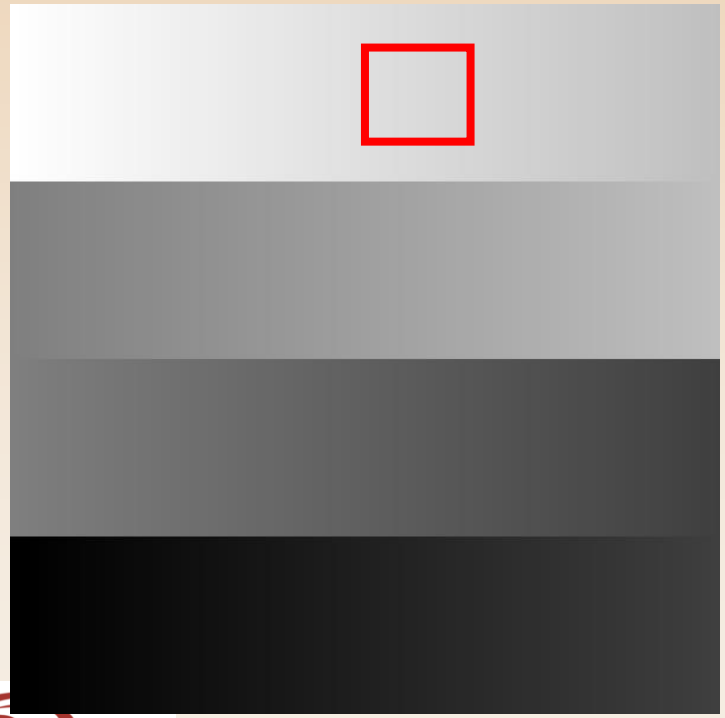
Halftoned using dot-off-dot error diffusion, magnified



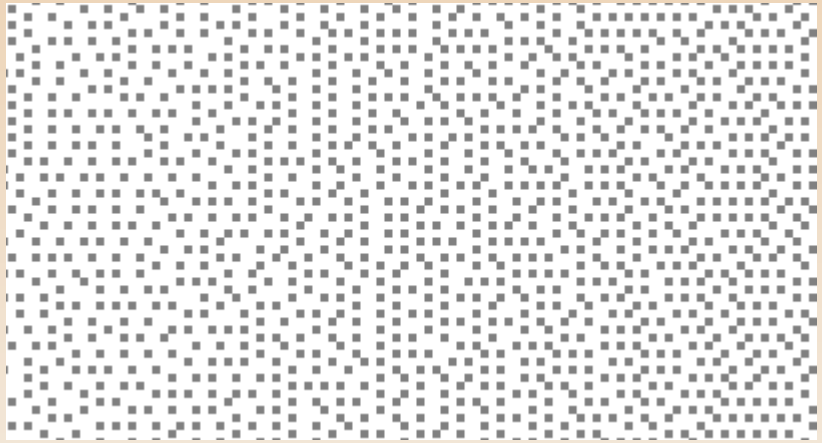
SUNVALLEY

Dot-off-dot halftoning : results & limits

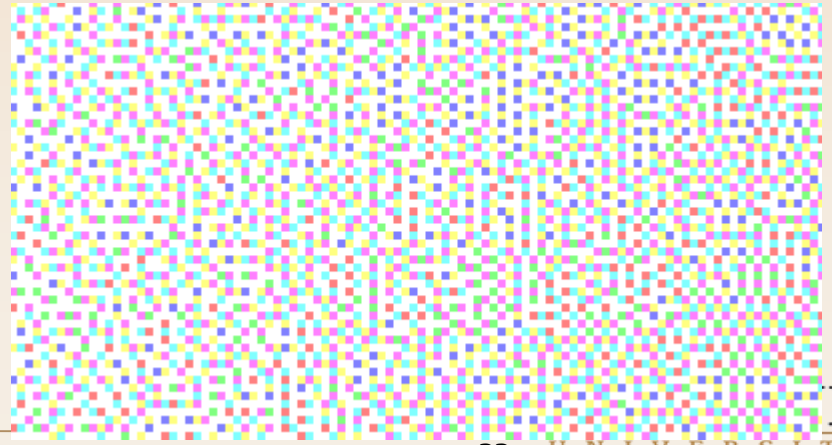
- Comparing the two halftone images below, we can see that dot-off-dot error diffusion tends to generate better quality halftone image
- Specifically, the artifacts observed from the simple multidrop error diffusion halftone images are somewhat alleviated
- We can also see that dot-off-dot halftoning cannot remove all halftone artifacts



Halftoned using simple 2-drop error diffusion, magnified

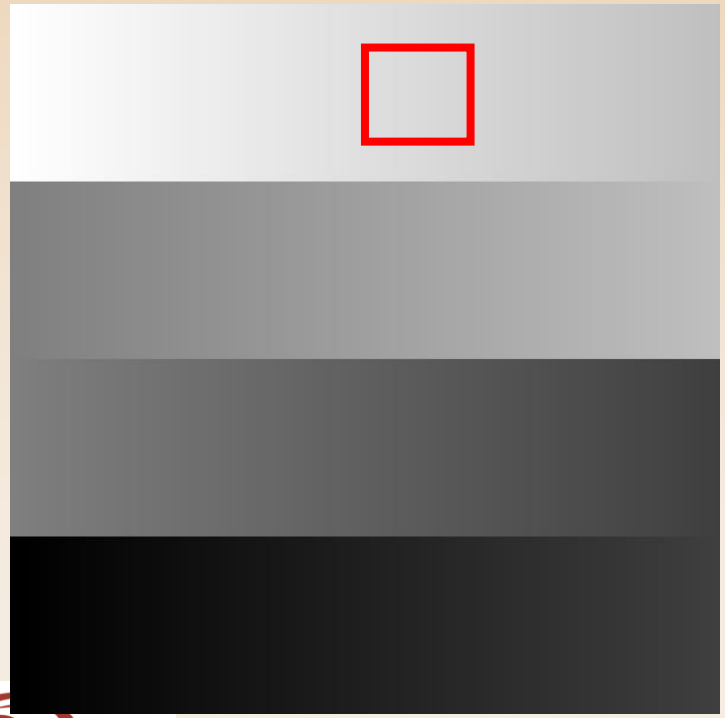


Halftoned using dot-off-dot error diffusion, magnified

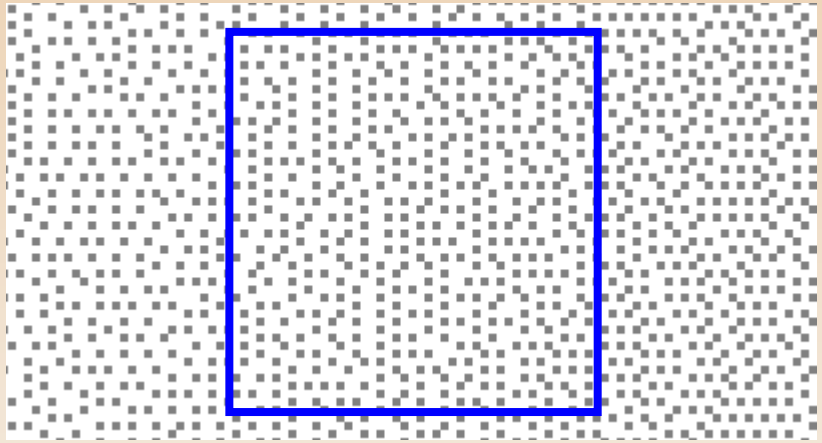


Dot-off-dot halftoning : results & limits

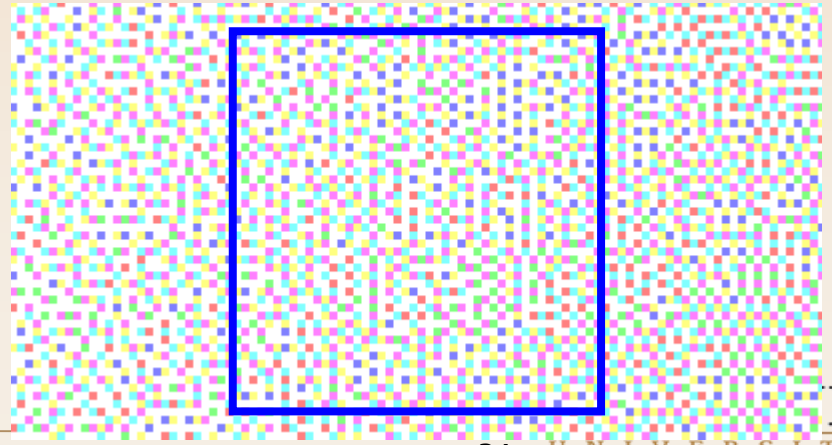
- Comparing the two halftone images below, we can see that dot-off-dot error diffusion tends to generate better quality halftone image
- Specifically, the artifacts observed from the simple multidrop error diffusion halftone images are somewhat alleviated
- We can also see that dot-off-dot halftoning cannot remove all halftone artifacts



Halftoned using simple 2-drop error diffusion, magnified



Halftoned using dot-off-dot error diffusion, magnified



Overview

- Motivation
- 2-drop error diffusion
- Dot-off-dot 2-drop error diffusion
- **DBS screen blending-in**
- Summary & Conclusions
- References

DBS screen blending-in : motivation

- Direct binary search (DBS) algorithm is known to generate high-quality halftone images [5]
- Using a DBS-generated screen to halftone a continuous-tone image eliminates the artifacts commonly associated with error diffusion. But the resulting halftone images are considerably noisier than those resulting from error diffusion.
- DBS screen blending-in was implemented with the goal that we can find a compromise between noisiness of screened halftone images and the artifacts associated with error diffused halftone images
- The DBS screen used is created based on the method explained in [6]

DBS screen blending-in : implementation

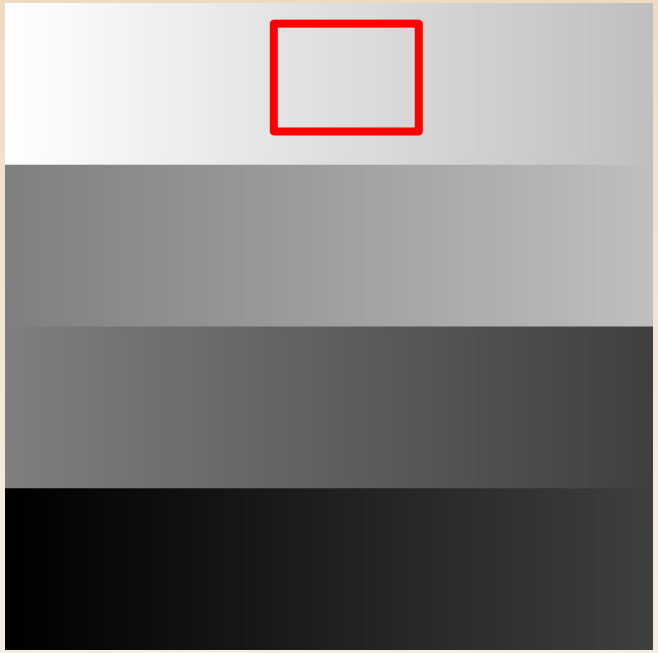
- To reduce the artifacts from error diffusion, we feed to the dot-off-dot error diffusion module the result of blending-in the original continuous-tone image with the tri-level DBS screened image
- For each color plane CMY of the original continuous-tone image f_c, f_m, f_y, we apply the tri-level DBS screen to obtain halftone images g_c, g_m, g_y
- For the three color planes CMY, we obtain mixtures $\tilde{f}_c, \tilde{f}_m, \tilde{f}_y$ of the trilevel screened image and the original continuous-tone image
- $\widetilde{f_k}[i,j] = 0.8 * f_k[i,j] + 0.2 * g_k[i,j], \quad k \in \{c,m,y\}$
- Then we perform dot-off-dot 2-drop halftoning on $\widetilde{f_k}, k \in \{c,m,y\}$

Tri-level screening

- To use DBS screen for tri-level screening, we first start with a set of three different bilevel DBS screens generated by following the method explained in [6]
- To create a trilevel halftoned image using bilevel DBS screen, the following method is used. We assume the screen and the image are normalized to be in [0,1] :
  - 1. Divide the screen elements by 2 and use the resulting scaled screen to halftone the image. This bilevel halftone will include the first drops of colorants.
  - 2. Subtract 0.5 from all pixels, and use the same scaled screen again to halftone the image. This bilevel halftone will include the second drops of colorants.
  - 3. Combine the two halftone images from step 1 and 2 to get the final tri-level halftone image.
- Three different screens are used for each color plane of cyan, magenta and yellow, which will have an effect similar to dot-off-dot halftoning.

Tri-level DBS screen blending-in

- We will show the digital halftones of the monochromatic ramp image below
- The red-squared area will be zoomed in to show the differences between the simple 2-drop error diffusion, the 2-drop dot-off-dot error diffusion without DBS screen blending-in and with DBS screen blending-in

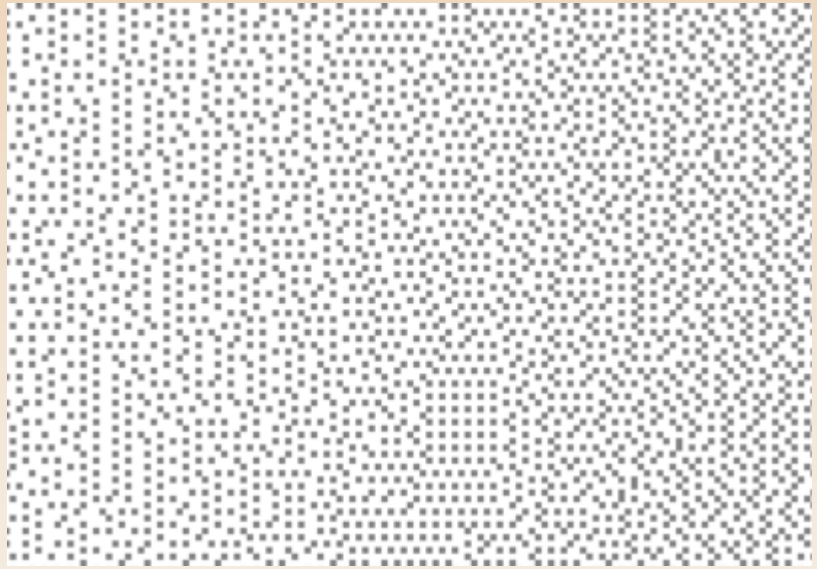


SUNVALLEY

PURDUE UNIVERSITY

Tri-level DBS screen blending-in

- Left : simple 2-drop, middle : dot-off-dot 2-drop, right : dot-off-dot 2drop & DBS screen blending-in



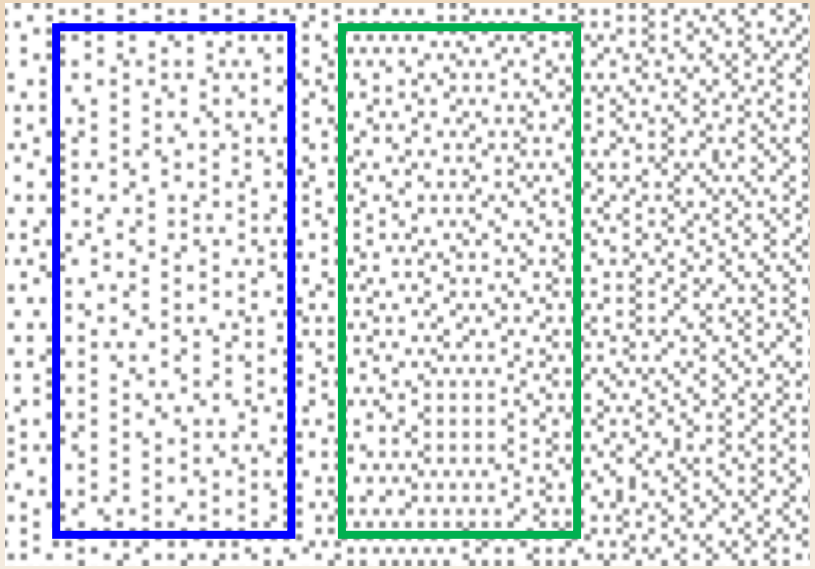
2-drop error diffusion

Dot-off-dot error diffusion

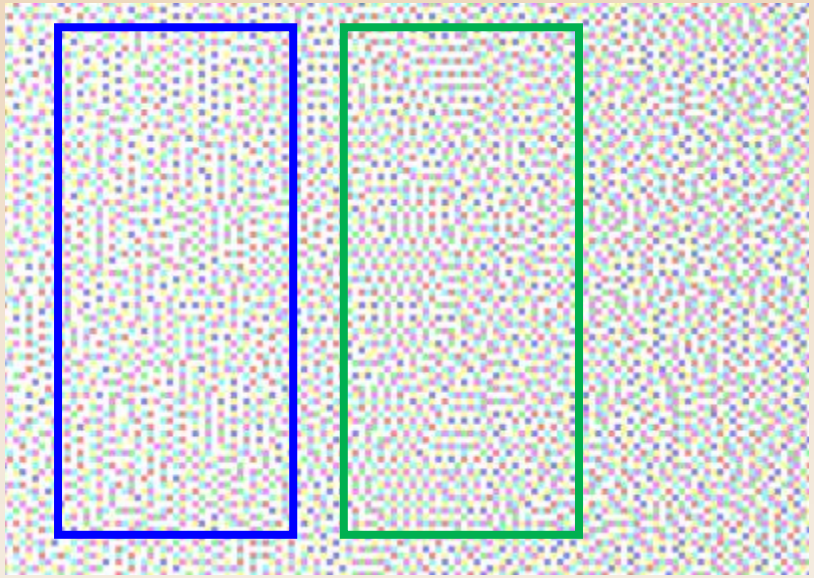
Dot-off-dot & DBS screen blending-in

Tri-level DBS screen blending-in

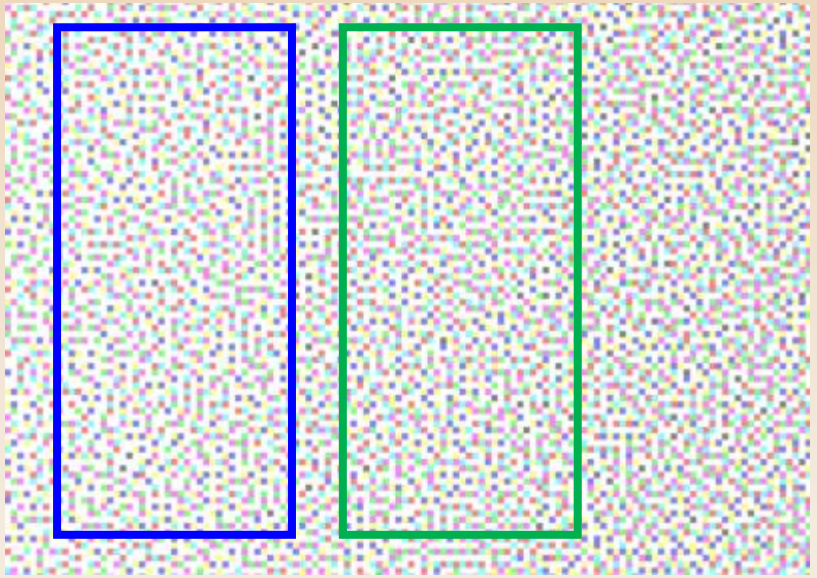
- Left : simple 2-drop, middle : dot-off-dot 2-drop, right : dot-off-dot 2drop & DBS screen blending-in



2-drop error diffusion



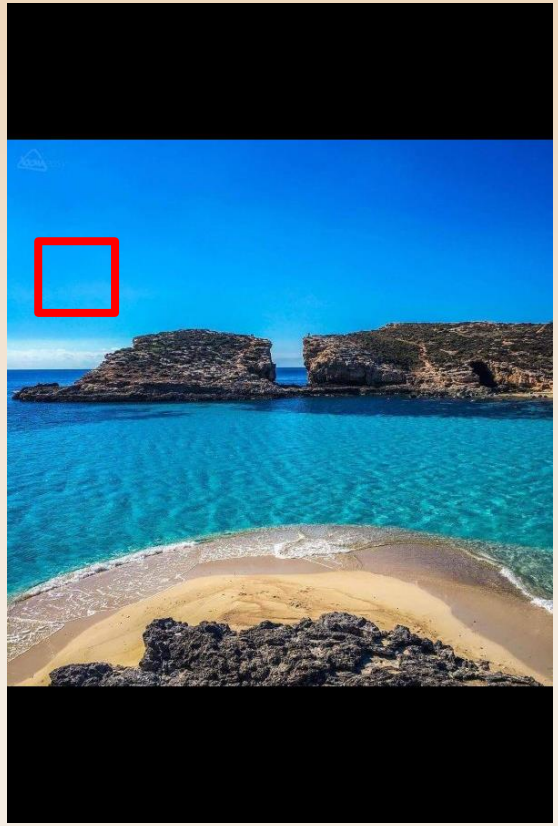
Dot-off-dot error diffusion



Dot-off-dot & DBS screen blending-in

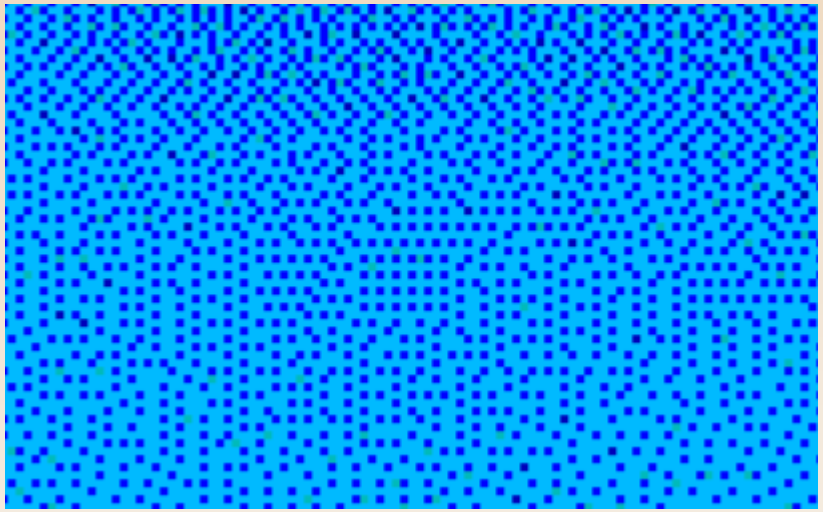
Tri-level DBS screen blending-in

- We will show the digital halftones of the sea image below
- The red-squared area will be zoomed in to show the differences between the simple 2-drop error diffusion, the 2-drop dot-off-dot error diffusion without DBS screen blending-in and with DBS screen blending-in

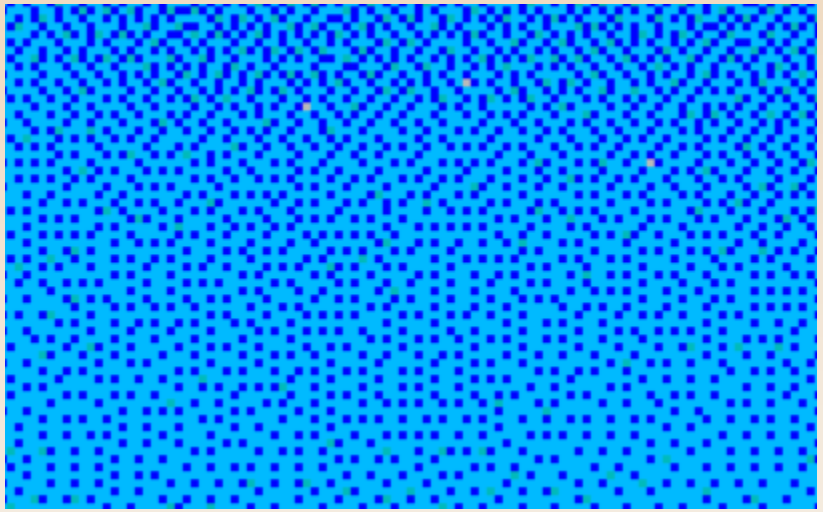


Tri-level DBS screen blending-in

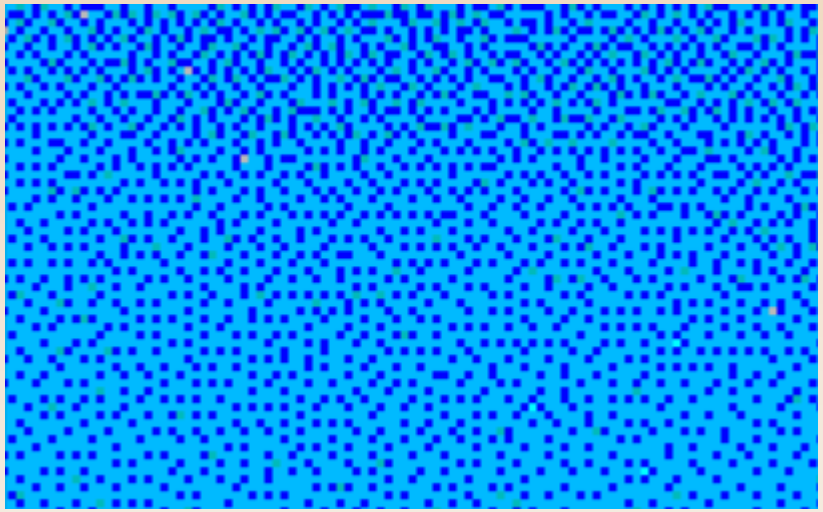
- Left : simple 2-drop, middle : dot-off-dot 2-drop, right : dot-off-dot 2drop & DBS screen blending-in



2-drop error diffusion



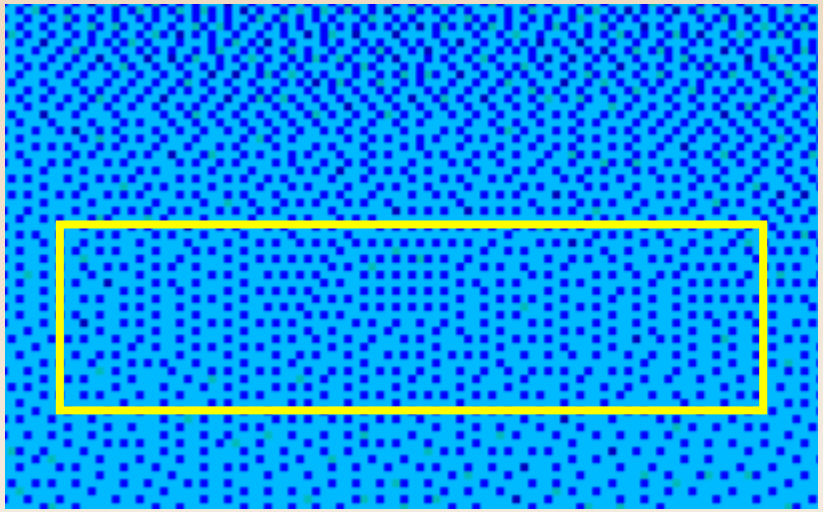
Dot-off-dot error diffusion



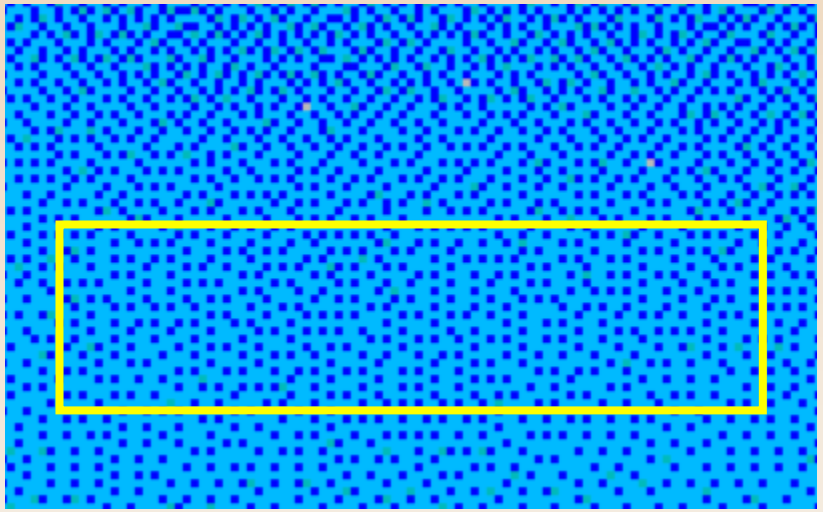
Dot-off-dot & DBS screen blending-in

Tri-level DBS screen blending-in

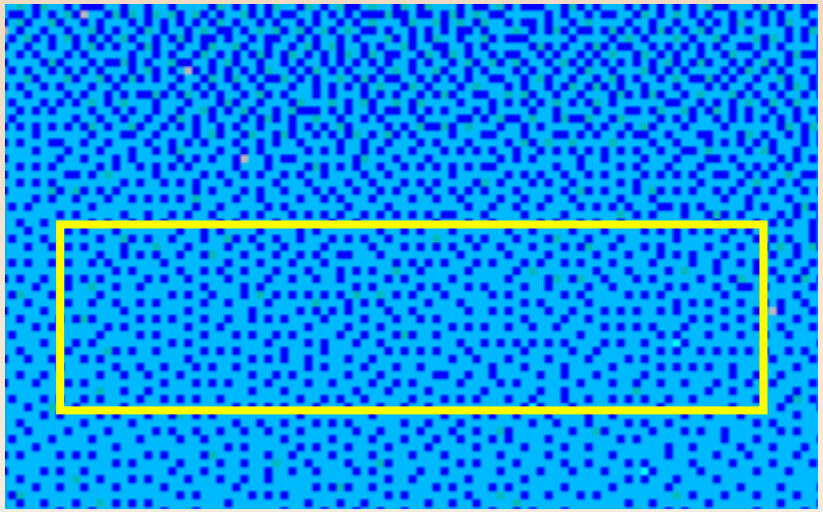
- Left : simple 2-drop, middle : dot-off-dot 2-drop, right : dot-off-dot 2drop & DBS screen blending-in



2-drop error diffusion



Dot-off-dot error diffusion



Dot-off-dot & DBS screen blending-in

Overview

- Motivation
- 2-drop error diffusion
- Dot-off-dot 2-drop error diffusion
- DBS screen blending-in
- **Summary & Conclusions**
- References

Summary & Conclusions

- We started with a 2-drop error diffusion algorithm, which is a simple extension of Floyd-Steinberg error diffusion
- Halftone images generated with the simple 2-drop error diffusion algorithm tends to show artifacts associated with error diffusion
- Dot-off-dot error diffusion halftoning is proposed with the goal of ink saving and smoother halftone textures
- Dot-off-dot error diffusion somewhat alleviates the artifacts but the effect is rather limited
- Blending-in the DBS screened image for dot-off-dot error diffusion results in halftone images without the artifacts associated with error diffusion and smoothness comparable to those created with error diffusion

References

- [1] R. W. Floyd and L. Steinberg, “An adaptive algorithm for spatial greyscale”, *Journal of the Society for Information Display,* vol. 17, n. 2, pp. 75-77, 1976.
- [2] R. Ulichney, *Digital Halftoning*, Cambridge, MA : The MIT Press, 1987.
- [3] H. R. Kang, *Digital Color Halftoning,* Bellingham, WA : SPIE Press, 1999.
- [4] W. Jiang et. al., “Novel color halftoning algorithm for ink savings”, in *Proc. of IS & T Symposium of Electronic Imaging, Color Imaging XXIII*, 2018, pp. 429-1-429-7 (7).
- [5] D. J. Lieberman and J. P. Allebach, “A dual interpretation for direct binary search and its implications for tone reproduction and texture quality”, in *IEEE Transactions on Image Processing*, vol. 9, n. 11, 2000, pp. 1950-1963.
- [6] J. P. Allebach and Q. Lin, “FM screen design using DBS algorithm”, in *Proc. IEEE International Conference on Image Processing*, 1996, pp. 549-52.