EE 569 Discussion 5

Yao Zhu

02/12/2021

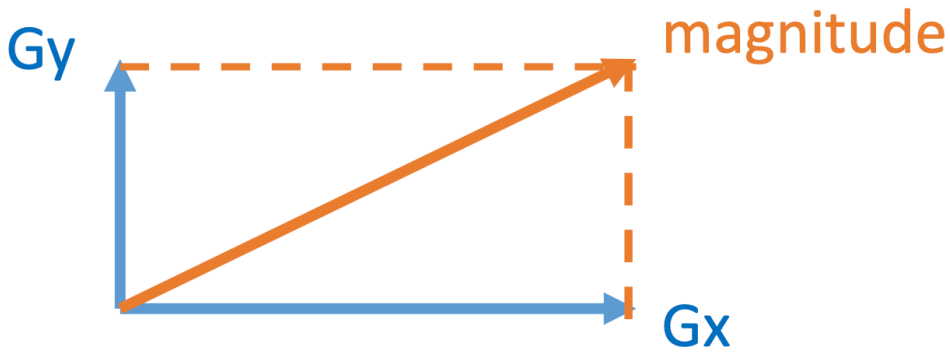
Contents

- HW2-P1 Edge Detection
  - Commonly asked questions
- HW2-P2 Digital Half-toning
  - Dithering
  - Error diffusion
- HW2-P3 Color halftoning
  - Separable error diffusion
  - MBVQ

HW2-P1 Edge Detection FAQ

1. Normalization for Sobel

- Normalize Gx, Gy just for visualization
- Calculate gradient magnitude M using the non-normalized Gx, Gy
- Normalize M
- Set different threshold



HW2-P1 Edge Detection FAQ

2. Installation of SE

- Recommend MATLAB version of 2016b or older: https://www.mathworks.com/downloads/web_downloads/download_release?release=R2016b
- Do not compile (skip step (c) in edges installation guide)
- Refer to Piazza for frequently installation question

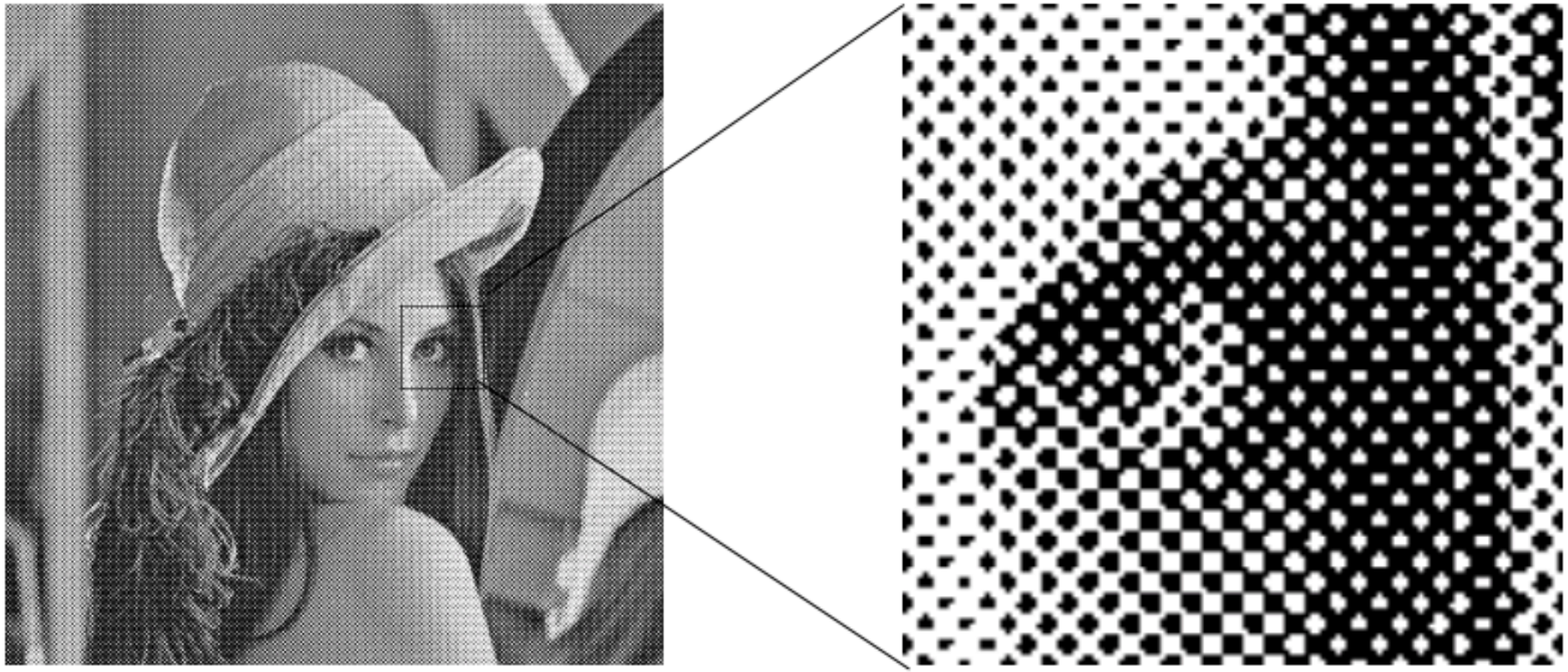
Contents

- HW2-P1 Edge Detection
  - Commonly asked questions
- HW2-P2 Digital Half-toning
  - Dithering
  - Error diffusion
- HW2-P3 Color halftoning
  - Separable error diffusion
  - MBVQ

HW2-P2 Digital Half-toning

Half-toning:

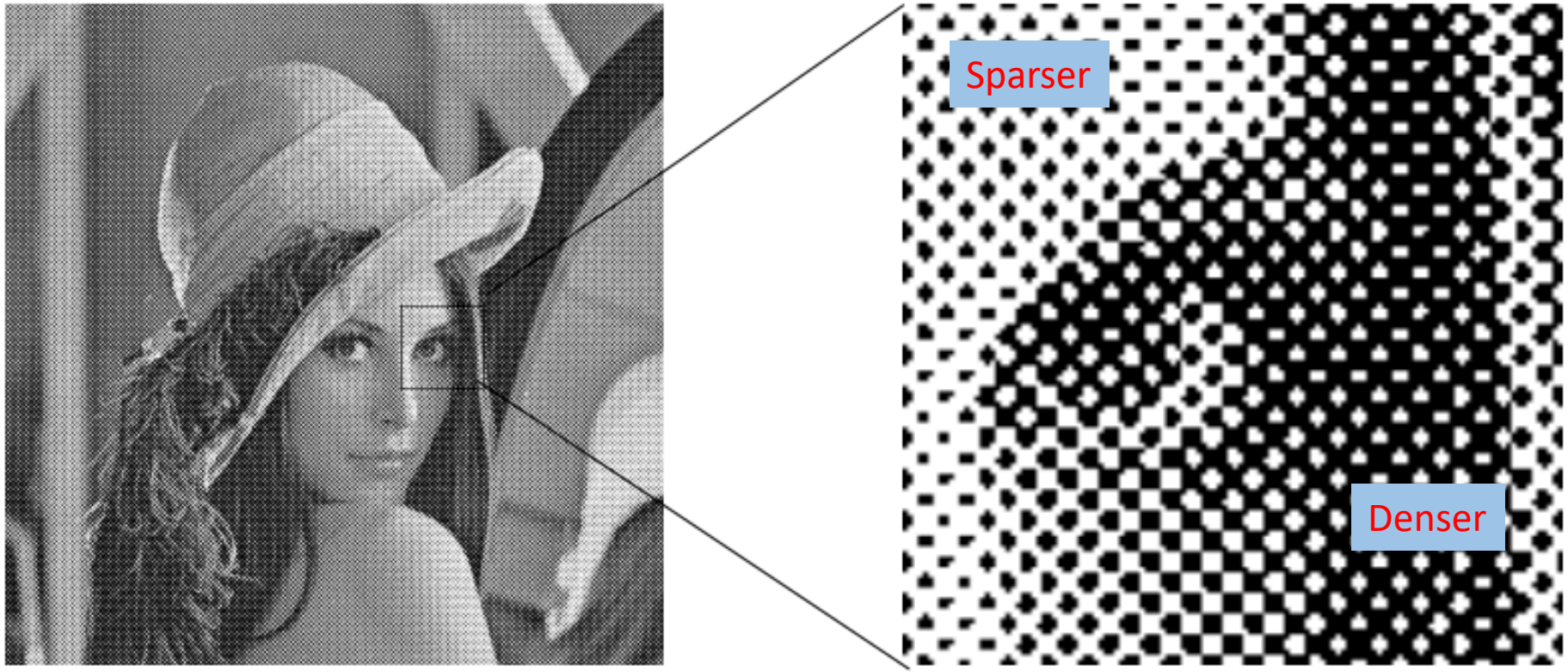
A method for creating the illusion of continuous tone output with a binary printing device.



HW2-P2 Digital Half-toning

Digital Half-toning:

To decide how to place black dots by using computer algorithms.



HW2-P2 Digital Half-toning

Why necessary?

Grayscale images: 8 bits → 256 tones

RGB Color images: 24 bits → $256^3 \approx 16.7$ million colors

But most printing devices are not able to reproduce different shadows of gray or so many different colors

So the original digital image must be transformed into an image containing only white (**no ink**) and black (**one dot of ink**)

Contents

- HW2-P1 Edge Detection
  - Frequently asked questions
- HW2-P2 Digital Half-toning
  - Dithering
  - Error diffusion
- HW2-P3 Color halftoning
  - Separable error diffusion
  - MBVQ

HW2-P2(a) Dithering

Binarize the grayscale using thresholds

$$output(i,j) = \begin{cases} 0 & 0 \le input(i,j) < \boxed{thre(i,j)} \\ 255 & thre(i,j) \le input(i,j) < 256 \end{cases}$$

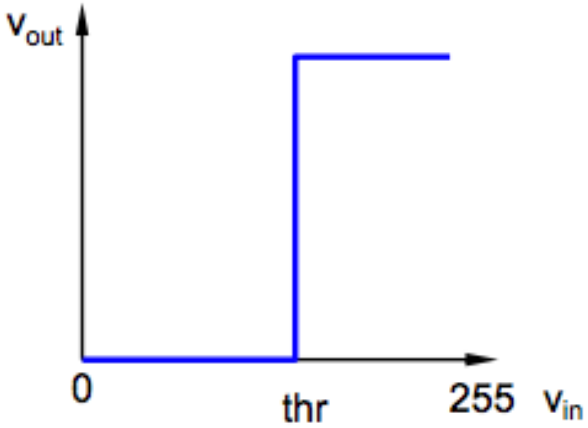
How to choose the thresholds?

HW2-P2(a) Dithering

(1) Constant threshold

$$G(i,j) = \begin{cases} 0 & \text{if } 0 \le F(i,j) < T \\ 255 & \text{if } T \le F(i,j) < 256 \end{cases}$$

8 bpp (256 levels)



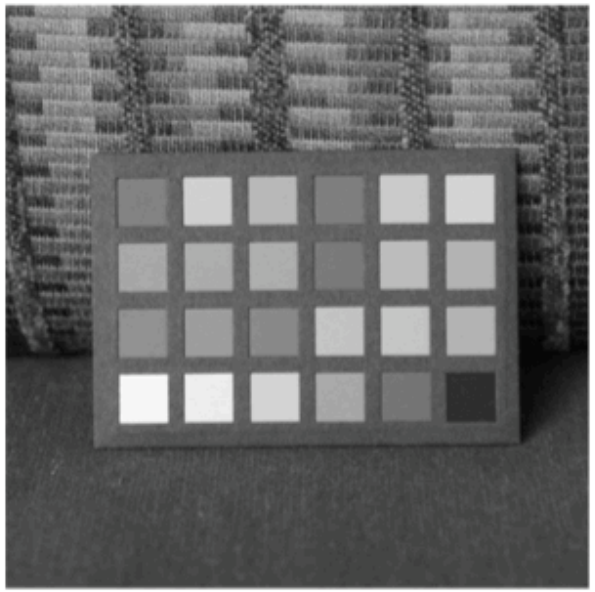
1 bpp (two-level)

Poor quality rendering of continuous tone image

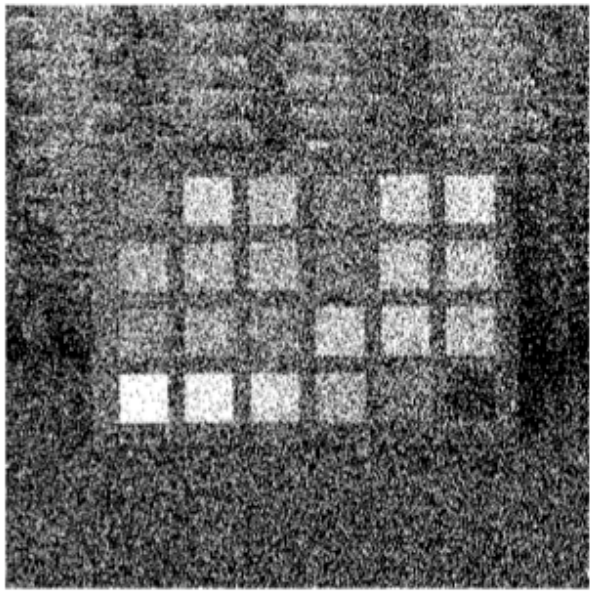
HW2-P2(a) Dithering

(2) Random thresholds

$$G(i,j) = \begin{cases} 0 & if\ 0 \le F(i,j) < rand(i,j) \\ 255 & if\ rand(i,j) \le F(i,j) < 256 \end{cases}$$



origin



random threshold dithering

HW2-P2(a) Dithering

(3) Dithering matrix

Dithering Matrix $I_{2n} = \begin{bmatrix} 4 \times I_n + 1 & 4 \times I_n + 2 \\ 4 \times I_n + 3 & 4 \times I_n \end{bmatrix}$

the initial one: $I_2 = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$

4 tones within range [0,4)

16 tones within range [0,16)

$I_2 = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} \Rightarrow I_4 = \begin{bmatrix} 5 & 9 & 6 & 10 \\ 13 & 1 & 14 & 2 \\ 7 & 11 & 4 & 8 \\ 15 & 3 & 12 & 0 \end{bmatrix} \Rightarrow I_8 \Rightarrow I_{16} \quad \ldots$

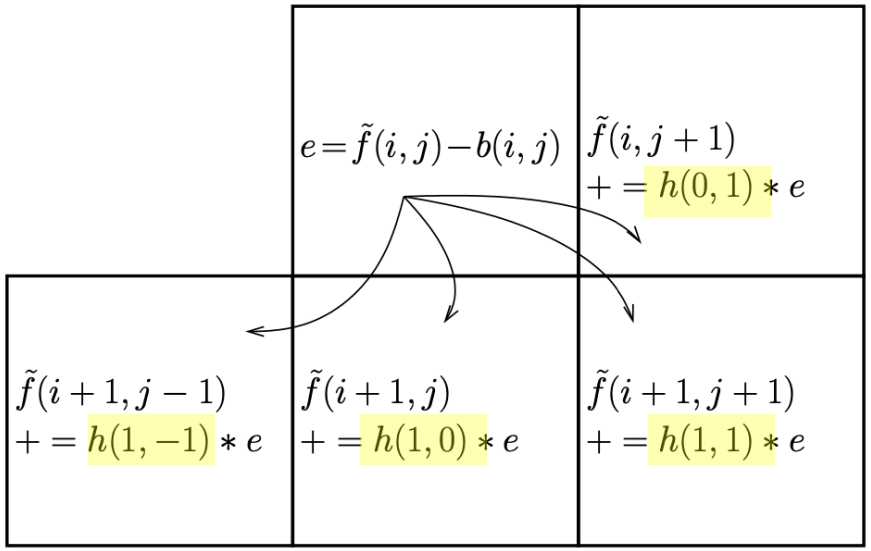
threshold matrix $T(x,y) = \dfrac{I_N(x,y) + 0.5}{N^2} \times 255$

Each value in threshold matrix is the threshold for that location

output: $G(i,j) = \begin{cases} 0 & if\ F(i,j) \le T(i\ mod\ N, j\ mod\ N) \\ 255 & otherwise \end{cases}$

HW2-P2(b) Error Diffusion

- Initialize $\tilde{f}(i,j) \leftarrow f(i,j)$
- For each pixel:
  - Binarize $$b(i,j) = \begin{cases} 255 & \text{if } \tilde{f}(i,j) > T \\ 0 & \text{otherwise} \end{cases}$$ Fixed threshold here
  - Diffuse error: Diffuse error forward using the following scheme "In-place" change



Choose *h* to have summation = 1

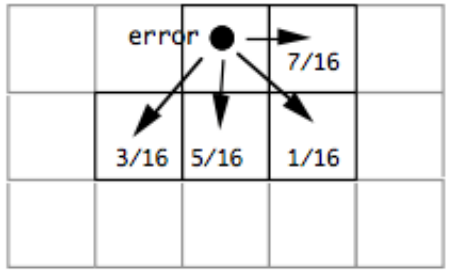
https://engineering.purdue.edu/~bouman/ece637/notes/pdf/Halftoning.pdf

HW2-P2(b) Error Diffusion

Different choices for *h* -- Error diffusion matrix

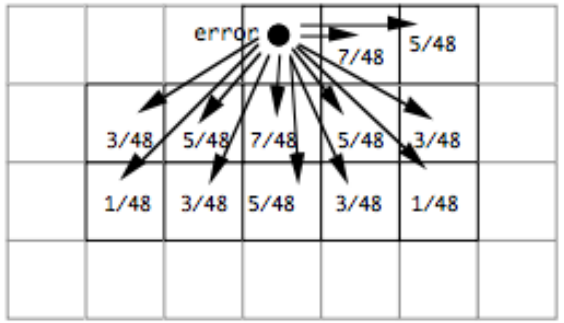
- **Floyd-Steinberg**

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$



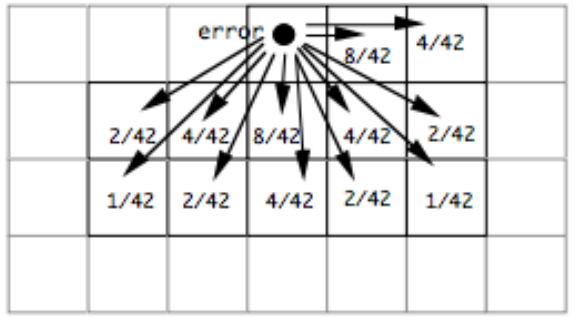
- **JJN**

$$\frac{1}{48}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$



- **Stucki**

$$\frac{1}{42}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$$

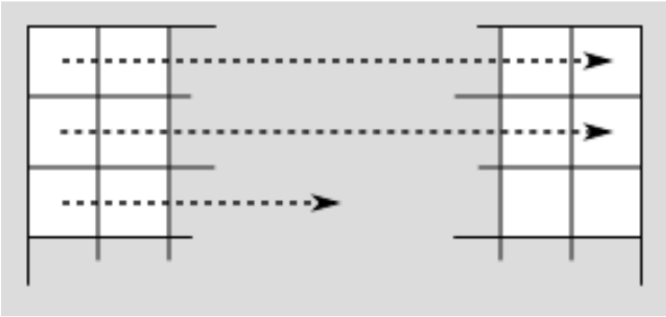


HW2-P2(b) Error Diffusion

Scanning order (crucial to results)

Use this for HW2

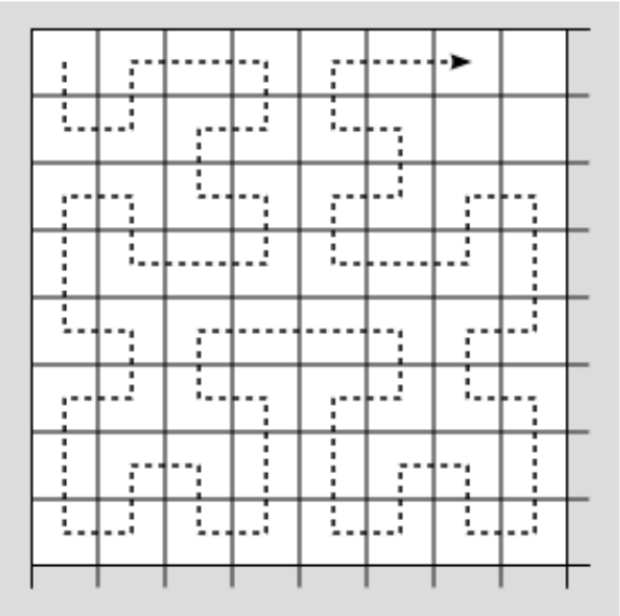
- **Raster parsing**



```
for i = 0:Height
    for j = 0:Width
```

- **Serpentine parsing**

- **Hilbert curve**



Mirror the diffusion matrix when going left

Can ignore the portion of error that is diffused out of the image boundary

Contents

- HW2-P1 Edge Detection
  - Frequently asked questions
- HW2-P2 Digital Half-toning
  - Dithering
  - Error diffusion
- HW2-P3 Color halftoning
  - Separable error diffusion
  - MBVQ

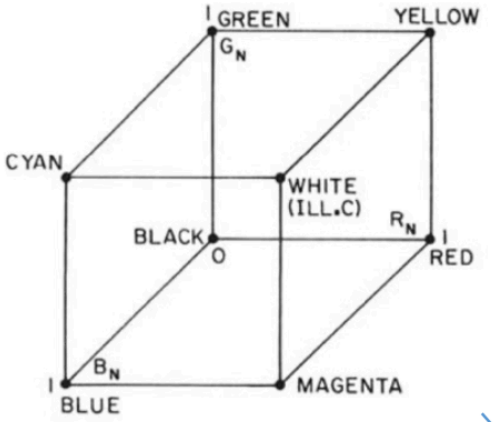
HW2-P3 Color Half-toning with Error Diffusion

Grayscale images: quantize 256 gray levels into 2 colors (black & white)

Color images: quantize $256^3 \approx 16.7$ million colors into $2^3 = 8$ colors
(binary in each channel)

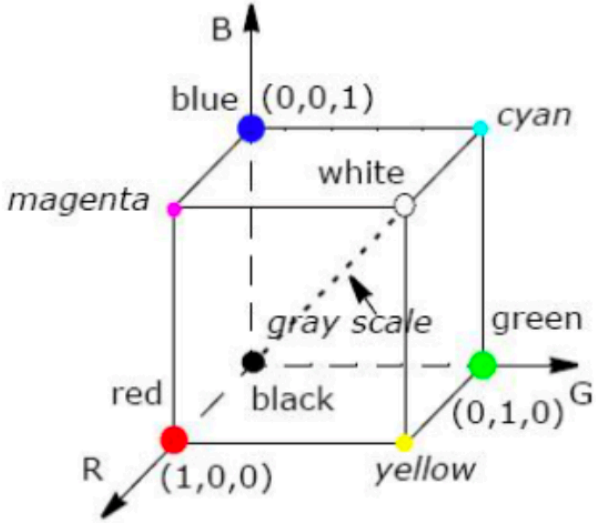
HW2-P3 Color Half-toning with Error Diffusion

- Color Space



(W, K), (Y, B), (C, R), (M, G)
are complementary color pairs

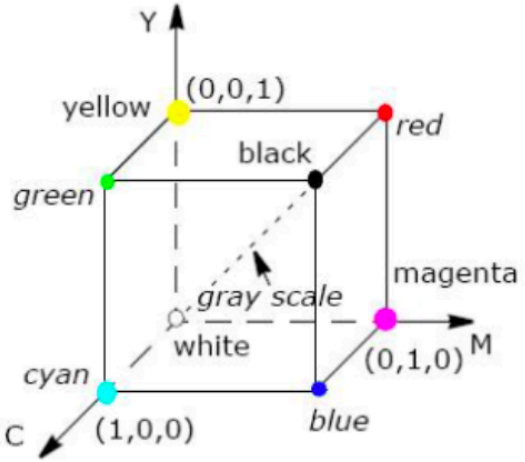
RGB Cube Representation



K = (0,0,0)
B = (0,0,1)
G = (0,1,0)
R = (1,0,0)
C = (0,1,1)
M = (1,0,1)
Y = (1,1,0)
W = (1,1,1)

CMY Cube Representation

Use this in half-toning



W = (0,0,0)
Y = (0,0,1)
C = (0,1,0)
M = (1,0,0)
G = (0,1,1)
R = (1,0,1)
B = (1,1,0)
K = (1,1,1)

HW2-P3 Color Half-toning with Error Diffusion

1. Separable Error Diffusion

Do halftoning for each of the CMY channels **independently**.

- Separate an image into CMY three channels
- Apply the Floyd-Steinberg error diffusion algorithm to quantize each channel separately

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Here assume CMY are in range [0,1]

HW2-P3 Color Half-toning with Error Diffusion

2. MBVQ-based Error Diffusion[1]

Quantize the color but preserve the energy (brightness)

Figure 1: The brightness scale of the eight basic colors rendered using *hp* inks.

[1] *D. Shaked, N. Arad, A. Fitzhugh, I. Sobel, "Color Diffusion: Error-Diffusion for Color Halftones", HP Labs Technical Report, HPL-96-128R1, 1996.*

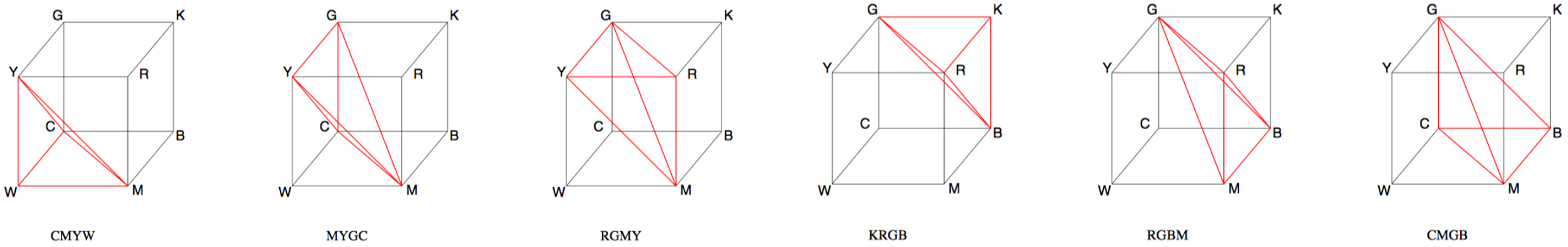
HW2-P3 Color Half-toning with Error Diffusion

2. MBVQ-based Error Diffusion[1]

The method proceeds by separating the RGB color space into minimum brightness variation quadrants (MBVQs)
→ render every input color using one of six complementary quadruples: RGBK, WCMY, MYGC, RGMY, RGBM, or CMGB, each with obviously minimal brightness variation.

Refer to [1] for more details.

MBVQ (Minimal Brightness Variation Quadruples)



[1] *D. Shaked, N. Arad, A. Fitzhugh, I. Sobel, "Color Diffusion: Error-Diffusion for Color Halftones", HP Labs Technical Report, HPL-96-128R1, 1996.*

HW2-P3 Color Half-toning with Error Diffusion

2. MBVQ-based Error Diffusion[1]

For each pixel (i, j) in the image do:

1. Determine MBVQ$(RGB(i, j))$. → limit the pixel into one quadrant
2. Find the vertex $v \in$ MBVQ which is closest to $RGB(i, j) + e(i, j)$. → further limit the pixel to the closest vertex
3. Compute the quantization error $RGB(i, j) + e(i, j) - v$. (Note: Error diffusion used in the paper is FS)
4. Distribute the error to "future" pixels.

[1] *D. Shaked, N. Arad, A. Fitzhugh, I. Sobel, "Color Diffusion: Error-Diffusion for Color Halftones", HP Labs Technical Report, HPL-96-128R1, 1996.*

HW2-P3 Color Half-toning with Error Diffusion

2. MBVQ-based Error Diffusion[1]

- Step 1: Determine MBVQ

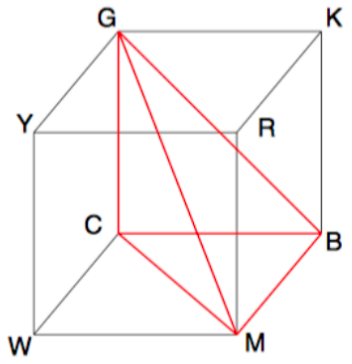
```
pyramid MBVQ(BYTE R, BYTE G, BYTE B)
{
  if((R+G) > 255)
    if((G+B) > 255)
      if((R+G+B) > 510)     return CMYW;
      else                  return MYGC;
    else                    return RGMY;
  else
    if(!((G+B) > 255))
      if(!((R+G+B) > 255)) return KRGB;
      else                  return RGBM;
    else                    return CMGB;
}
```

[1] *D. Shaked, N. Arad, A. Fitzhugh, I. Sobel, "Color Diffusion: Error-Diffusion for Color Halftones", HP Labs Technical Report, HPL-96-128R1, 1996.*

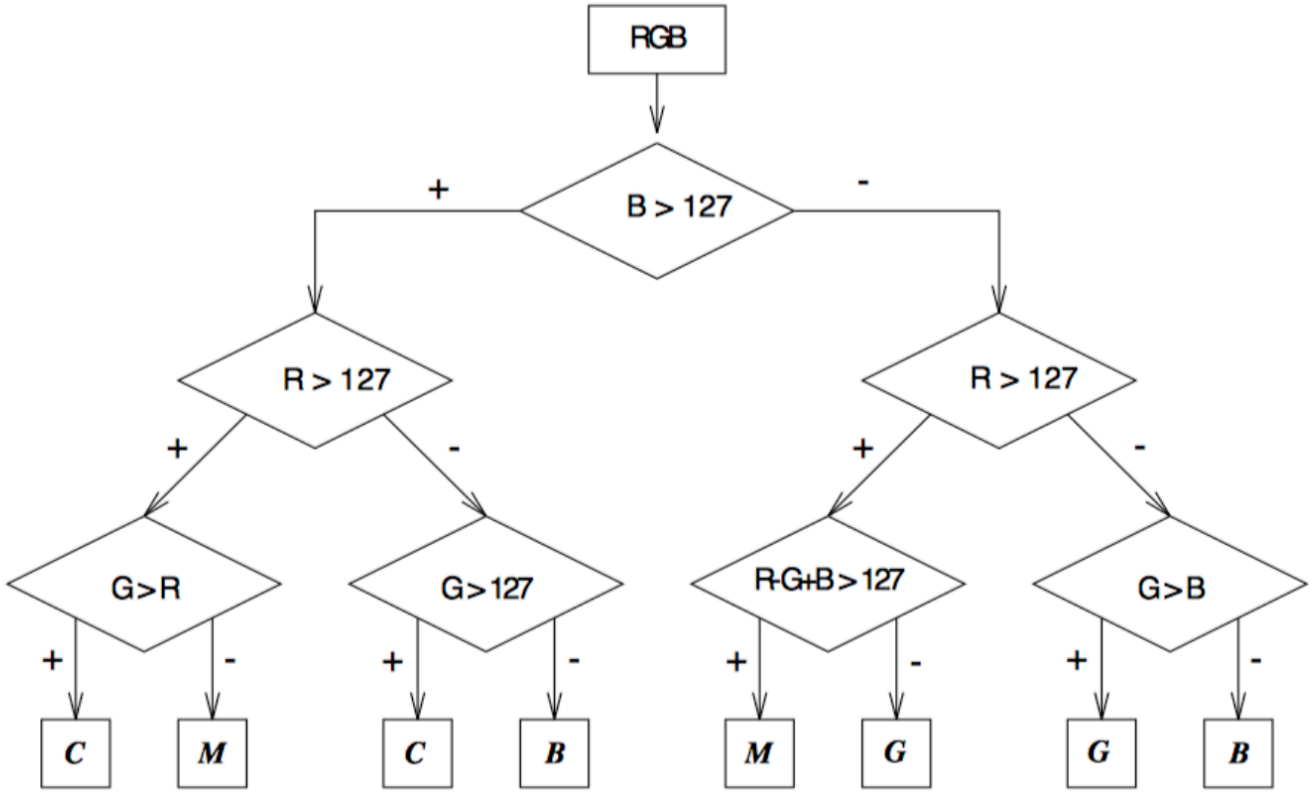
HW2-P3 Color Half-toning with Error Diffusion

2. MBVQ-based Error Diffusion[1]

- Step 2: To determine the closet vertex
  - (e.g. for CMGB Quadruple)



CMGB



[1] *D. Shaked, N. Arad, A. Fitzhugh, I. Sobel, "Color Diffusion: Error-Diffusion for Color Halftones", HP Labs Technical Report, HPL-96-128R1, 1996.*

Q
A

References:

- Yijing Yang, EE569 2020 spring discussion 4