Problem 2: Digital Halftoning (30%)

(a) Dithering (Basic: 6%)

Convert the 8-bit *Mandrill* image in Fig. 7 to a half-toned image using the dithering method.

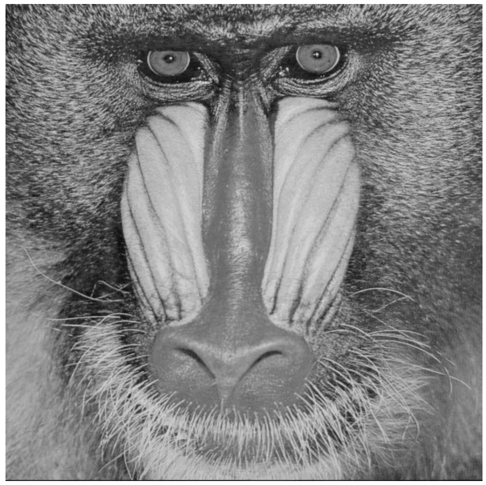

Figure 7: mandrill.raw

Dithering parameters are specified by an index matrix. The values in an index matrix indicate how likely a dot will be turned on. For example, an index matrix is given by

$$I_2(i,j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

where 0 indicates the pixel most likely to be turned on, and 3 is the least likely one. This index matrix is a special case of a family of dithering matrices first introduced by Bayer [1].

The Bayer index matrices are defined recursively using the formula:

$$I_{2n}(i,j) = \begin{bmatrix} 4 * I_n(x,y) + 1 & 4 * I_n(x,y) + 2 \\ 4 * I_n(x,y) + 3 & 4 * I_n(x,y) \end{bmatrix}$$

The index matrix can then be transformed into a threshold matrix T for an input gray-level image with normalized pixel values (*i.e.* with its dynamic range between 0 and 1) by the following formula:

$$T(x,y) = \frac{I(x,y) + 0.5}{N^2}$$

where N^2 denotes the number of pixels in the matrix. Since the image is usually much larger than the threshold matrix, the matrix is repeated periodically across the full image. This is done by using the following formula:

$$G(i,j) = \begin{cases} 1 & \text{if } F(i,j) > T(i\, modN, j\, modN) \\ 0 & otherwise \end{cases}$$

where *F(i,j)* and *G(i, j)* are the normalized input and output images. Answer the following questions.

1. Construct $I_4(i,j)$ and $I_8(i,j)$ Bayer index matrices and apply them to the *Mandrill* image.
2. If a screen can only display FOUR intensity levels, design a method to generate a display-ready *Mandrill* image. Show your best result in gray-scale with four gray-levels (0, 85, 170, 255) and explain your design idea and detailed algorithm.

(b) Error Diffusion (Basic: 6%)

Convert the 8-bit *Mandrill* image to a half-toned one using the error diffusion method. Show the outputs of the following three variations, and discuss these obtained results.

1. Floyd-Steinberg's error diffusion with the serpentine scanning, where the error diffusion matrix is

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

2. Error diffusion proposed by Jarvis, Judice, and Ninke (JJN), where the error diffusion matrix is

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

3. Error diffusion proposed by Stucki, where the error diffusion matrix is

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

Describe your own idea to get better results. There is no need to implement it if you do not have time. However, please explain why your proposed method will lead to better results.

(c) Scalar Color Halftoning (Basic: 6%)

A naive idea is to apply the monochrome halftoning method independently to the colorant [cyan, magenta, yellow, and black (CMYK)] planes. Implement this idea as follows: Separate an image into CMY three channels and apply the Floyd-Steinberg error diffusion algorithm to quantize each channel separately. Then, you will have one of the following 8 colors, which correspond to the 8 vertices of the CMY cube at each pixel:

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)

Note that (W, K), (Y, B), (C, R), (M, G) are complementary color pairs as illustrated in Figure 5. Please show and discuss the result of the half-toned color *Sailboat* image. What is the shortcoming of this approach?

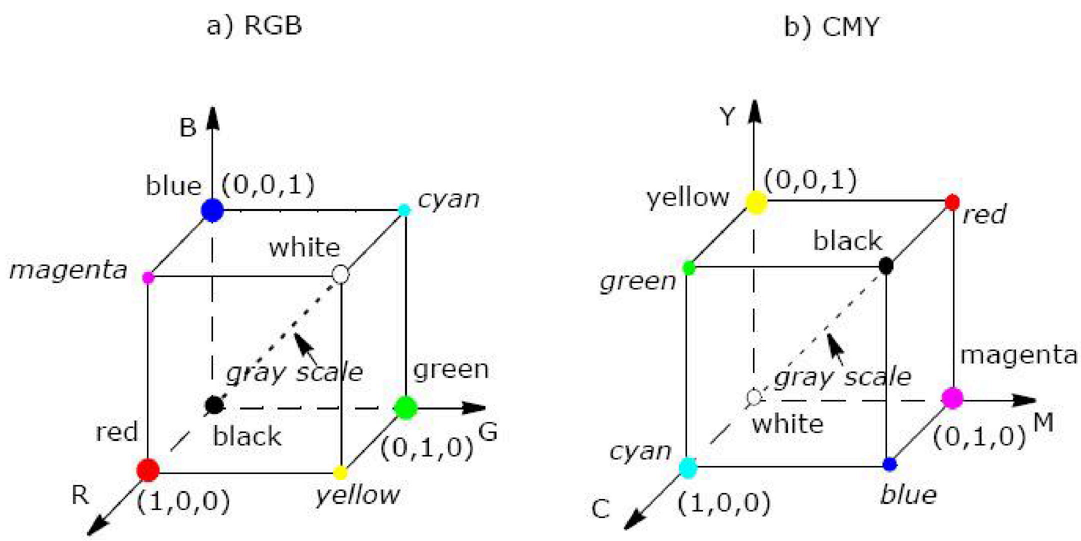

Figure 8: RGB and CMY Cube

(d) Vector Color Halftoning (Advanced: 12%)

We expect better results by conducting color halftoning in three channels jointly. It is called the "vector color halftoning" technique. Shakad *et al.* [2] proposed a vector color error diffusion method. They partition the CMY color space into 6 Minimum Brightness Variation Quadruples (MBVQ) as shown in Fig. 2 of [2]. Let the CMY value and its quantization error at pixel (x, y) be CMY(x, y) and e(x, y), respectively. The error diffusion can be conducted as follows:

Initialization: Set the quantization error at all pixels to zero.
MBVQ Quantization and Color Error Diffusion: Scan pixels in the input image with the serpentine order and repeat the following operations.

1) **MBVQ Quantization:** Find the vertex, V, of the MBVQ tetrahedron that is closest to CMY(x, y) + e(x, y) and output the value of V at location (x, y);
2) **Color Error Diffusion:** Compute the new quantization error using CMY(x, y) + e(x, y) – V, and distribute the quantized error to the error buckets e(x,y) of future pixels using a standard error diffusion process (e.g. the FS error diffusion).

You may refer to [2] for more details. Implement this algorithm and apply it to the *Sailboat* image in Figure 6. Compare the output with that obtained in 1. . Comment on the differences between these two methods.