Multimedia Processing and Communications **Fall-2020**

Homework # 2 **Due: Nov. 26th, 2020**

Programming exercises

Please submit all the plots and your code. Put title and labels on your plots; include comments throughout your code.

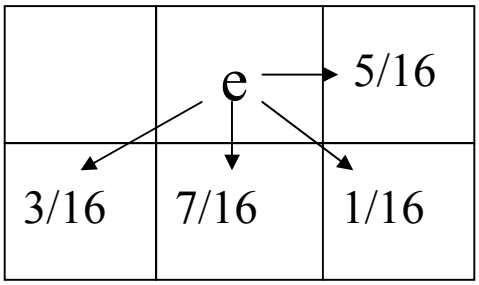
Floyd-Steinberg dithering

The Floyd-Steinberg algorithm achieves dithering by diffusing the quantization error of a pixel to its neighboring pixels. One of the strengths of this algorithm is that it minimizes visual artifacts through an error-diffusion process; the Floyd-Steinberg algorithm typically produces images that more closely represent the original than simpler dithering algorithms.

The steps of dithering procedure are given below. Implement these steps in your code:

1. Download **lena512** image from Teams. Load the image; display it on the screen.
2. Scale the pixel values such that all gray levels are in the interval [0,1].
3. Start at one corner and work through image pixel by pixel (Scan top to bottom row by row in a raster pattern).
    a. Threshold the pixel: If the intensity r(x,y)< 0.5, replace with black (t(x,y)=0) , else replace with white (t(x,y)=1).
    b. Compute the error at that pixel: The difference between what should be there and what you did put there (e(x,y)=r(x,y)-t(x,y)). Propagate error to neighbors by adding some proportion of the error to each unprocessed neighbor. More specifically, 5/16 of the error is added to the pixel to its right, 3/16 of it to the pixel to its lower left, 7/16 to the pixel below, and 1/16 to the pixel to its lower right. That is;

$$r(x+1, y) = r(x+1, y) + \frac{5}{16} e(x, y)\,, \quad \text{and so forth.}$$



4. Repeat steps 3.a and 3.b until all pixels are visited.
5. Plot the resulting dithered image (t(x,y)).

Comment on the quality of the dithering procedure. What would happen if there were no error propagation (Perform thresholding without dithering and plot the resulting monochrome image)? How does error propagation help improve the quality of the dithered image?