Lab 4 Problem

Objective: Simulation of Printing with and without Error Diffusion

Let P1 and P2 be two bi-level printers. Bi-level printers are the printers which will print only two shades(black and white) even when the input image is colour or gray scale image.

The bi-level printer P1 prints the colour image as binary image using the following transformation :

1) Colour image is converted to gray scale using the formula : f '(x,y) = (0.299r(x,y)+0.587g(x,y)+0.114b(x,y)), where the input image f(x,y)=(r(x,y), g(x,y), b(x,y))
2) The gray scale image f'(x,y) is converted into binary image using thresholding, where the threshold is 127 for the image with pixel range 0 to 255(ie. B1(x,y) =0 if f'(x,y)<=127; B1(x,y)=255 otherwise)
3) The resultant binary image B1(x,y) will be printed

Whereas the bi-level printer P2 prints the colour image as binary image using the following transformation :

1) Colour image is converted to gray scale using the formula : f '(x,y) = (0.299r(x,y)+0.587g(x,y)+0.114b(x,y)), where the input image f(x,y)=(r(x,y), g(x,y), b(x,y))
2) Apply Floyd –Steinberg error diffusion algorithm on f'(x,y), say the resultant image is B2(x,y), which is a binary image
    a. The gray scale image f'(x,y) is converted into binary image using thresholding after distributing error, where the threshold is 127 for the image with pixel range 0 to 255
3) The resultant binary image B2(x,y) will be printed (Note that B2(x,y) will be 0 or 255 for all (x,y))

a) Simulate process of printing by printers P1 and P2 and display the final output of P1 and P2 in any language of your choice(Python, MATLAB, OpelGL) for the colour input lena image given below.
b) Display f'(x,y) along with B1(x,y) and B2(x,y), and find out whether B1 is close to f' or B2 is close to f'
c) Print the local average absolute error between f'(x,y) and B1(x,y) and also between f'(x,y) and B2(x,y). The local average absolute error for f' and B1 is defined at each pixel (x,y) as the average of absolute difference f' and B1 in 3x3 window centred at (x,y). Similarly for f' and B2.
d) It is more complex to use OpenGL, hence you will be awarded extra credit if OpenGL is used

In Floyd- Steinberg algorithm, You are expected to use three neighbours: for (x,y), consider (x+1,y), (x+1, y+1), (x,y+1) with weights 3/8, 2/8,3/8 respectively. And also consider top-left corner as origin.