CMPUT 411/511

Assignment 3

Herb Yang

October 12, 2018

Due date **Oct. 18, 2018** at 23:55
Total marks: 25

Submit your assignment electronically to eClass. Organize your files properly into folders and zip up all the folders that correspond to a particular question into one single zip file for submission.

Important notes

- A late penalty of 10% per day applies to all late submissions. The maximum number of days late permitted is 1. After 1 day late, your assignment will be given a mark of **ZERO**.
- You are expected to complete the assignment on your own **without** collaboration with others. Discussions at the conceptual level but not at the coding level are permitted.
- Looking at other's code or letting other to see your code is not permitted.
- Do not wait until the last minute to work on your assignment. Start as soon as possible. Debugging programs and understanding of the materials take time.
- During the lab, the TA is instructed to give directions but **not** solutions. As well, debugging is not part of the duty of the TA.
- For the programming part,
  - Your code should have sufficient comments to make it readable.
  - Include a README file to document features or bugs, if any, of your program.
- If there are questions, please ask the TA in the lab or via email.

(25 marks) The problem will help you understand one particular class of halftoning techniques, namely, Floyd-Steinberg, and dot diffusion. In the provided code, there is a function to dither the image to show you how to access different channels of an image and how to output the result. In particular, the input is in `image` and the output in `output`. There are two image files provided, namely, `house`, and `rainbow`. After you compile and run the supplied skeleton code, you can open and view the supplied images.

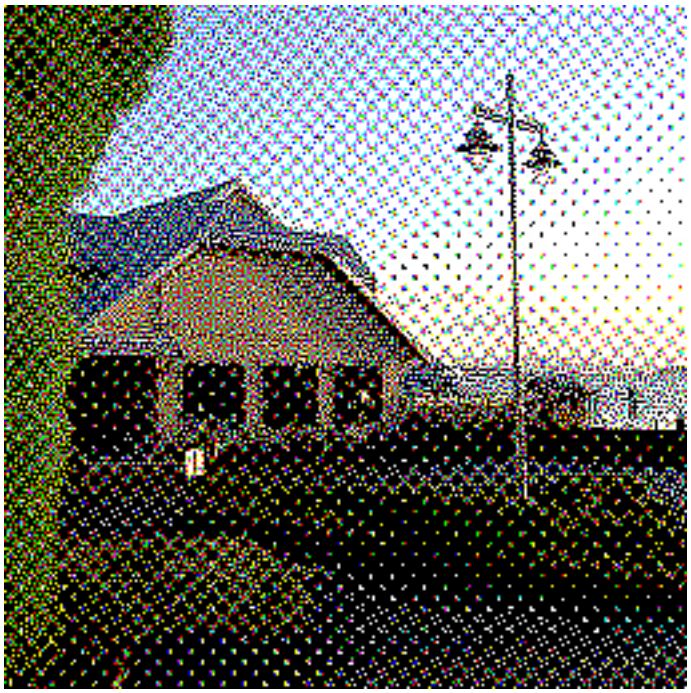
1. (10 marks) Use the skeleton code provided, implement the Floyd-Steinberg error diffusion algorithm as discussed in class. In particular, add the code in the function `floyd()`. If your implementation is correct, you should see the following result:

Figure 1. Halftoning using Floyd-Steinberg error diffusion algorithm with a threshold of 128.

Note, to obtain a similar result, you need to scan the image from left to right and then from top to bottom.

2. (15 marks) Using the skeleton code provided, implement the Dot Diffusion algorithm as discussed in class. Please see the original paper for more details (Digital halftones by dot diffusion). (If your implementation is correct, you should see the following:

(a)



(b)

Figure 2. Halftoning using Dot Diffusion using (a) M1 and (b) M2 with a threshold of 128.

The functions that you are required to implement are described in the following:

(a) (1 mark) `weight(int x, int y)` - returns the weight of the neighbour at x and y. See text for the definition.

(b) (1 mark) `isValidNeighbour(int x, int y)` - a neighbour is not valid if it is outside of the range of the image; otherwise it is valid.

(c) (2 marks) `classRank(int x, int y)` - returns the rank of a pixel. The rank depends on which choice is selected. In this part, there are two possible choices, either the `M1` matrix is used or the `M2`. The choice is set by setting the variable `matrixChoice`, which is input by the user.

(d) (11 marks) `dotDiffusion`. This is the main function to implement for this part of the assignment. It should call the above functions.

Note: In this assignment, you may need to implement additional functions.