CMPT365 Project 1

Junchen Li

301385486

2021/3/12

Introduction:

We need to write a simple decoder to read tif. format image files and display the original image, grayscale image, ordered dithering image and dynamic range adjustment image in user interface. Test the dithering matrices and thresholds for different values to find the best way to display ordered dithering image and dynamic range adjustment image.

Experiment environment:

I decided to use Java in IntelliJ. Most libraries which I used in this report is about JavaFx. JavaFx library is used for setting the UI and this library contains a lot of UI elements. Like setting ActionEvent and EventHandler for menu button, make a file chooser, scene, border pane, stage. There are two libraries for the file are:
"java.io.File" When user choose a tiff file and we set a variable which its data type is File. And we can decode this File variable.
"java.io.IOExpection" When we select a tif. File and if it is not existing, we need to throw an exception, leave an error message and terminate the program.
There are two external libraries that using for displaying interpreted image:
"javafx.scene.image.ImageView" for set some details of image view and display the interpreted image.
"java.awt.*" is used for set the Color variable for displaying the each pixel in the image. The color variable will contain the data of R,G,B.

Designing the project:

1) Extract and display original image

Using the "FileChooser" and a tif filter for users to choose a tif. Image. Let the "selected file" be a File data type variable. I made a class object called TIFF which for decoding the image and storing all information of image. Following the TIFF version 6, add some members which contains the weight of image, height of image, number of pixels, resolution information, a Byte array for including all bytes of image, and four array lists: bits per sample, strip offset, counts for strip byte and sample format. According to the online resource to parse the meaning of each element, and using function "GetDEValue" to extract information from each element. The function "DecodeStrips" used for decoding the RGB information of each pixel in the Byte array and set them into the image. The original image could just use data from the above processes. Using the image View to display the original image.

2) Grayscale Image

Since we already get the RGB information of each pixel. We can use them directly for converting to grayscale image. We can use each pixel information of RGB and follow the property. For a gray color, R = G = B, the luminance y equal to that gray, since 0.299 + 0.587 + 0.114 = 1.0. We can use 0.299(R) + 0.587(G) + 0.114 (B) to convert each pixel to gray level pixel. Then we can get the Grayscale Image.

3) Dithered image

At the beginning of experiment, I use "Random" to random pick up the value of index element in dithering matrix. Start with 2 * 2 matrix. Due to the different range between image value and dithering value. We use formula:
image value/ (256 / dithering matrix size) to scale reduce. Then comparing this proportional matrix each value with the dithering value. If bigger, then print nothing; if smaller, then print dot. (More information will discuss later).

4) dynamic range adjusted image

The principle of dynamic range adjusted image is setting darker parts of image more and more brighter and setting brighter parts of image more and more darker. Due to the range of [0~255]. I decided to find the max value between RGB values at first. And if the max value is bigger and equal to 100 then, we reduce that R or G or B by fifteen and rest of two I reduce by ten.

Discussion: How we can choose a nice dithering matrix? This part is based on online knowledge research. I will leave the reference at the end of report. As the above, I generated a random matrix for testing, however, the result is not that we expected. According to the YuXiao said: A pixel is divided into N *N parts, and the brightness of each part is f(x). When N is large enough, the pixel brightness is approximately expressed as: $I = \sum_i^N f(x_i)$. What we hope more is that a result of the original image is only represented by an indication of the output result. We can sample multiple times and the result will get closer and closer to the real result. This is why if you use a random matrix to perform operations, I will get a result that is not relatively satisfactory (Due to the variance is larger). Now we have a clear goal, if I want to find a good matrix, I have to make sure that every two numbers are quite different from each other. Follow the next formula, it will help us to find a nice dithering matrix.

$$M(2n) = \begin{vmatrix} 4 * M(n) + 0 & 4 * M(n) + 2 \\ 4 * M(n) + 3 & 4 * M(n) + 1 \end{vmatrix}$$

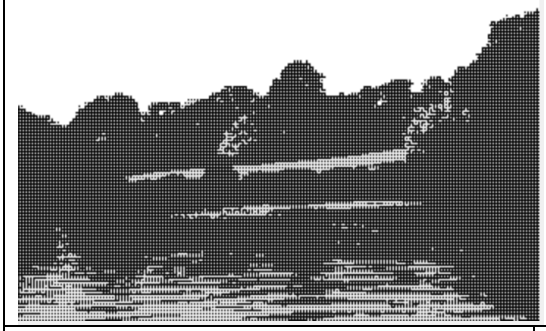
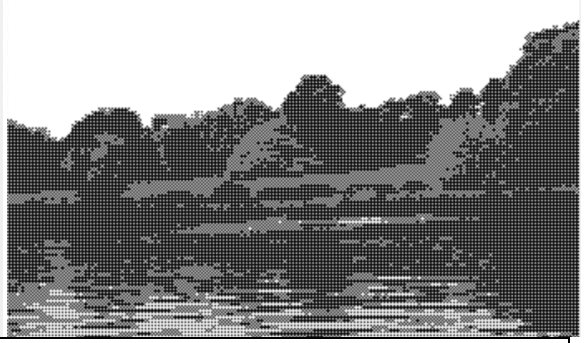
According to the `ordered_dithering.txt` we can have this order to print above number. I use a 4 * 4 Matrix as an example:

The same color block will correspond to each element in the M matrix. And follow the 1~4 order write element into each four blocks.

Theoretically, the bigger N * N size matrix could get, the better result, but for this experiment I'm only going to work with 4 by 4 matrices. So the final dithering matrix which have the best performance as the follow:

$$Dithering\ matrix = \begin{vmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{vmatrix}$$

Following comparison diagram will show difference between the two more visually

The random dithering matrix	Calculated dithering matrix
	
	

How we can choose a nice threshold for dynamic range adjustment?

The whole process is that I choose the max value between R,G,B value because of this value will decide what the dominant color is. I can't change the color to another color, like change bright blue to red. What I want is change bright blue to dark blue. Then I choose the threshold is the half of the 255 which means if the value over 127, this color of pixel will consider to dark color. And vice versa. If we need change one color to brighter one (> 127), I change the main color (the max value of RGB) lower by 15 and rest of another two will reduce by 10. Following comparison diagram will show difference between the two more visually. (I use the same image as the above showing)

The original image	Dynamic Range image (127)
	
Dynamic Range image (100)	Dynamic Range image (56)
	
Dynamic Range image (156)	Dynamic Range image (200)
	

The value in the bracket is represented by threshold for dynamic range adjustment.

In my opinion, the threshold below the 56 will cause an amorphous of adjustment image.

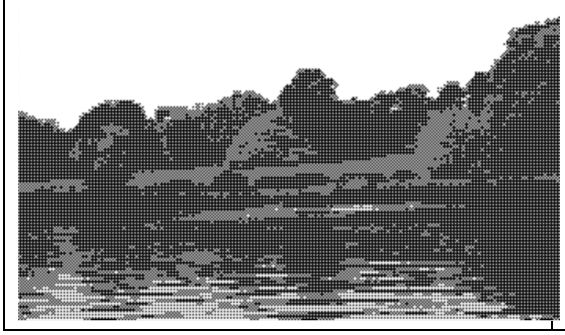
If we change the increase(decrease) range of secondary color, the only cause the whole image become more red.

The original image	Change secondary color
	

In conclude, I set the number 127 to the threshold value and the main color will increase (decrease) by value of 15, the secondary values will increase (decrease) by value of 10.

Report Experiment Result:

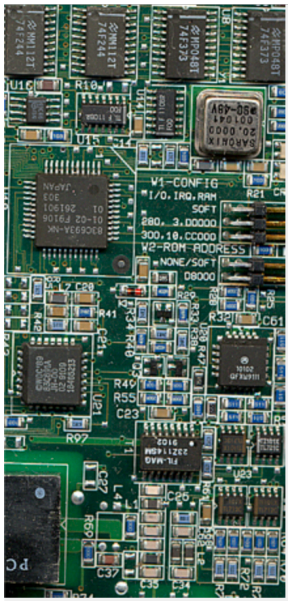
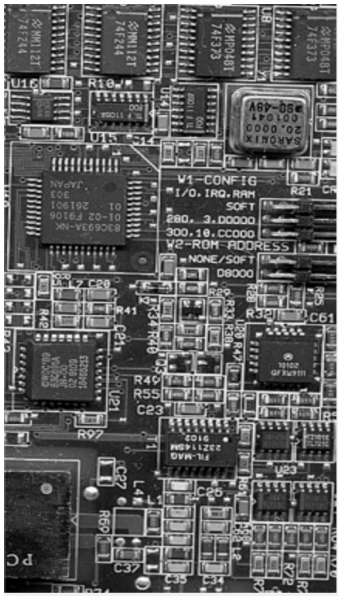
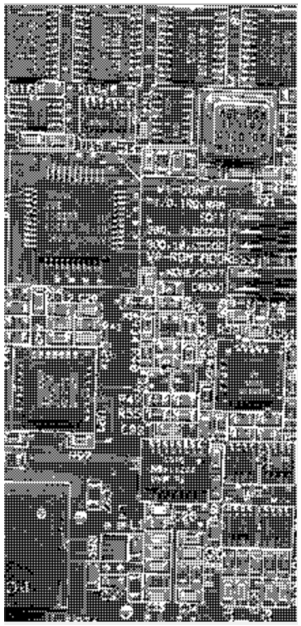
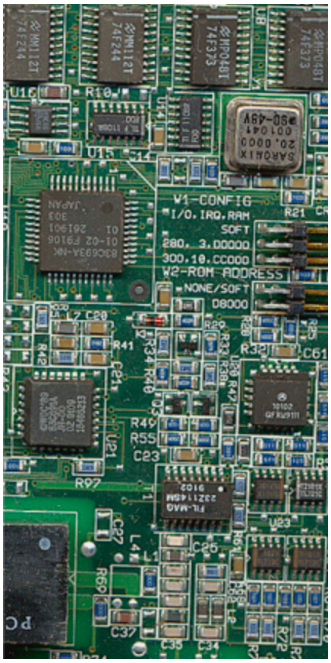
The autumn.tif

The original color image	Grayscale image
	
Ordered dithering image	**Dynamic Range image**
	

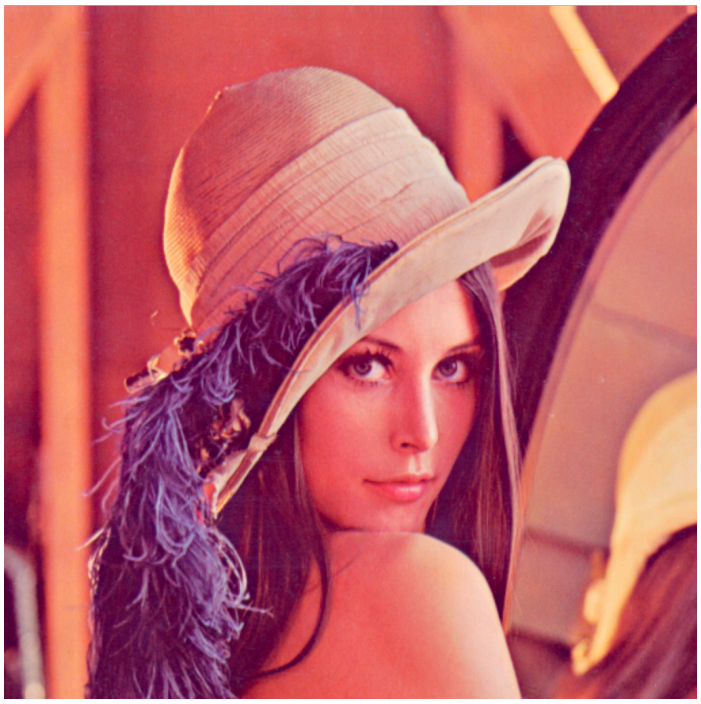
The balloons.tif

The original color image	Grayscale image
	
Ordered dithering image	**Dynamic Range image**
	

The board.tif

The original color image	Grayscale image
	
Ordered dithering image	Dynamic Range image
	

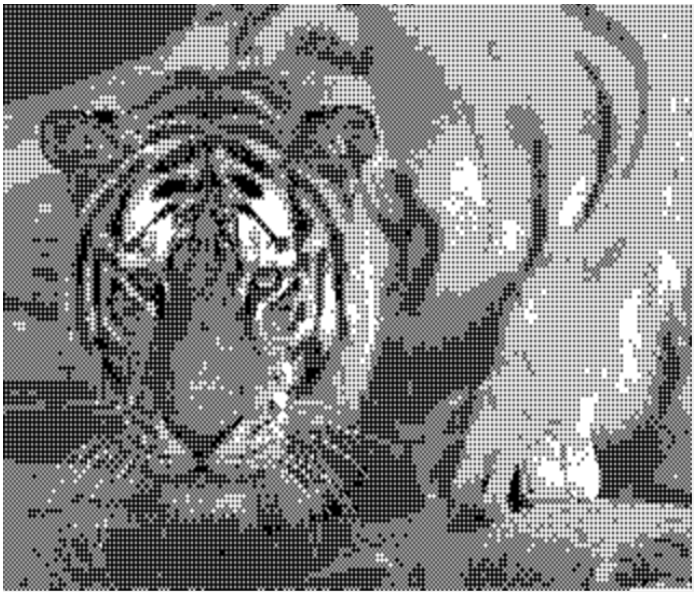
The lena.tif

The original color image	Grayscale image
	
Ordered dithering image	**Dynamic Range image**
	

The parrots.tif

The original color image	Grayscale image
	
Ordered dithering image	**Dynamic Range image**
	

The tiger.tif

The original color image	Grayscale image
	
Ordered dithering image	Dynamic Range image
	

Reference

Note: These references are about all images that additional adding, some ideas about choosing dithering and threshold and some source code for decoding image part.

https://forums.adafruit.com/viewtopic.php?f=47&t=35042
https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node170.html#24bitimage
https://zhuanlan.zhihu.com/p/110104674
https://www.imaging.org/site/PDFS/Papers/1999/RP-0-93/1786.pdf
https://gist.github.com/MehdiNS/bd41bbc6db780c9409157d35d331ac80#file-ordered_dithering-txt
https://blog.csdn.net/qq_16555407/article/details/88365151?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522161542620316780271573087%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=161542620316780271573087&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-1-88365151.pc_search_result_hbase_insert&utm_term=
https://docs.oracle.com/javase/8/docs/api/java/lang/Byte.html
https://www.jianshu.com/p/ff32eb09ed3d