

CMPT 365 Project 1

Yi Hui Chen

May 9, 2021

1 Libraries used

1.1 java.awt.*

- java.awt.BorderLayout
- java.awt.Dimension
- java.awt.Image
- java.awt.Toolkit
- java.awt.event.ActionEvent
- java.awt.event.ActionListener
- java.awt.event.KeyEvent
- java.awt.image.BufferedImage
- java.awt.Color
- java.awt.Graphics
- java.awt.Graphics2D

Items above used for capturing user actions, image display and layout.

1.2 java.swing.*

- javax.swing.BorderFactory
- javax.swing.JLabel
- javax.swing.JPanel
- javax.swing.JOptionPane
- javax.swing.ImageIcon
- javax.swing.JButton
- javax.swing.JFileChooser
- javax.swing.JFrame

- javax.swing.JMenu
- javax.swing.JMenuBar
- javax.swing.JMenuItem

Items above used for application GUI.

1.3 java.nio.file.*

- java.nio.file.FileSystem
- java.nio.file.Files
- java.nio.file.Path
- java.nio.file.Paths

Items above used for directory/file access.

1.4 java.io.*

- java.io.File
- java.io.FileInputStream
- java.io.FileOutputStream
- java.io.IOException
- java.io.InputStream

Items above used for saving/opening image files.

1.5 javax.imageio.ImageIO

Used for opening images that are not in TIFF format.

1.6 Self written classes

- DEntry Used for reading and storing entries of tiff image.
- Header Used for reading and storing file header information.
- ImageLabel is an extension of JLabel, it is use to display image.
- ImagePanel is an extension of JPanel, it is use to arrange the layout.
- MyFrame is an extension of JFrame, it is the main window frame of the application.
- TIFFimage TIFFimage is a tiff file management class that uses DEntry to keep tiff file information and offers functionality such as saving iamges in tiff format, reading tiff RGB values and store the RGB values in the BufferedImage class.

2

2.1 Methods used to read and present the color image

- Read the whole file and store every byte as an integer in an int array.
- Read the first 2 byte to define the byte ordering of file.
- According to "II" or "MM" endian, read and confirm byte 2-3 is in correct byte ordering and contain value 42.
- Read byte: 4-7 to get the offset of the IFD. In this project, images are assume to have only one IFD.
- Go to the first IFD and read the entry count.
- For each entry, record the tag, data type, count and offset/value.
- Determine if offset/value field is offset or value. ($count \times (dataSize) > 4$) implies offset, offset/value field is value otherwise.
- If offset/value field is value, value is store in an array of length 1. Else application go to the offset location and retrieve ($count \times dataSize$) bytes, convert and store every offset value in an integer array of length $count$.
- For each stripOffset value, go to the stripOffset location and retrieve the according strip-ByteCount number of bytes and store store every three byte as a RGB value in the BufferedImage.
- Location of pixels is calculated as horizontally, $x = (i^{th} \text{ group of 3 bytes}) \% \text{ ImageWidth}$, and vertically, $y = \text{floor}((i^{th} \text{ group of 3 bytes}) \div \text{ ImageWidth})$.
- Image then displays by using graphic2d's draw function if no processing required.
- If image are to be processed displayed, the BufferedImage or the original file will be passed to visual effect functions and RGB values of every pixels are modified and then displayed on application.

2.2 Grayscale image

Grayscale image is generated by modification of rgb values in every pixel. For each pixel, covert the convert rgb to yuv, and set each value in rgb to y.

2.3 Ordered dithering image

Bayer matrix of size 8x8 is used for ordered dithering. To perform ordered dithering, the maximum and minimum range of the grayscale image's color is first calculated. Then the range is divided by the 64(8x8) possible values in the dithering matrix to get var . And then map dithering matrices to every sub 8x8 fields of the image. For every pixel, if the rbgValue $\div var$ is greater than the coordinate of the dithering matrix mapped to that pixel, the pixel's rgb is set to 255,255,255, pixel's rgb is set to 0,0,0 otherwise.

2.4 Dynamic range adjusted image

The mean of brightness Y is calculated among all of the pixels. For every pixel, if the y' of the pixel's rgb value is greater than the mean brightness of Y , the each value of rgb is reduced by 15% of its own brightness. If the y' is darker than the Y , then y' is increase by 15%.

3 Dither matrix

0	32	8	40	2	34	10	42
48	16	56	24	50	18	58	26
12	14	4	36	14	46	6	38
60	28	52	20	60	30	54	22
3	35	11	43	1	33	9	41
51	19	59	27	49	17	57	25
15	47	7	39	13	45	5	37
63	31	55	23	61	29	53	21

The values in this matrix are calculated by an algorithm to produce an overall evenly "colored" effect when it is "colored". I believe this would make the image look smoother after ordered dithering, and the size of this matrix can support a reasonably wide range of different "colors".

4 Dynamic adjustment experiment

4.1 Adjusting the luminosity at a bound

When only adjusting the 20% brightest or 20% darkest values of RGB of the image and reducing/increasing the y' by 50%, the image seems darker overall and looks warmer. And colors are not as sharp.



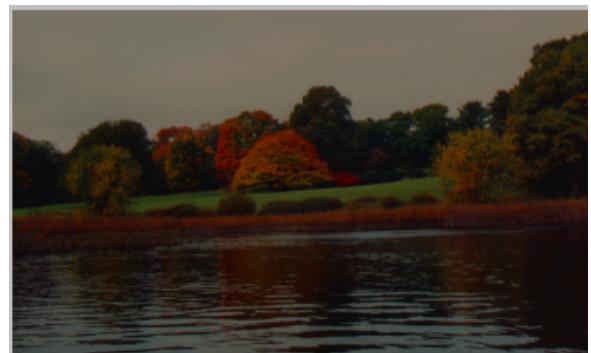
(a) Original



(b) Dynamic Range Adjusted



(a) Original



(b) Dynamic Range Adjusted

4.2 Adjusting the luminosity that does not cross the mean luminosity

When only adjusting the luminosity of the pixels by 20% and the adjustment does not make the pixel go from brighter than average to darker than average or from darker than average to brighter than average. (not crossing the mean luminosity after the adjustment.) The image looks a little odd for image that has greater range of luminosity level, but image with low luminosity levels looks just fine.



(a) Original



(b) Dynamic Range Adjusted



(a) Original



(b) Dynamic Range Adjusted

5 Result images

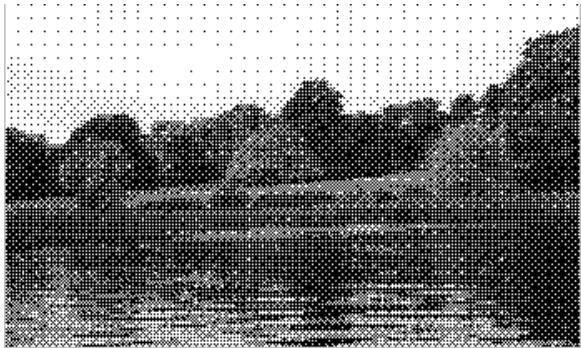
5.1 autumn.tif



(a) Original



(b) Grayscale



(c) Order Dithering



(d) Dynamic Range Adjusted

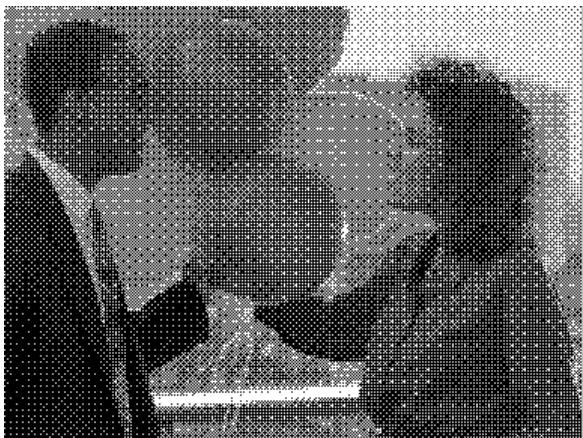
5.2 balloons.tif



(a) Original



(b) Grayscale

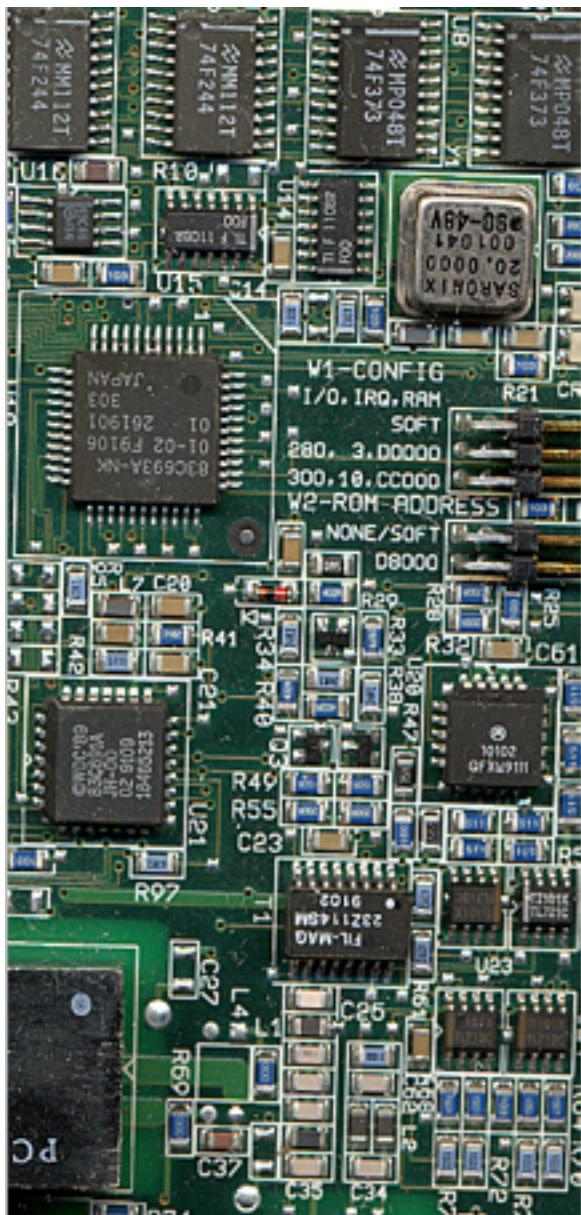


(c) Order Dithering

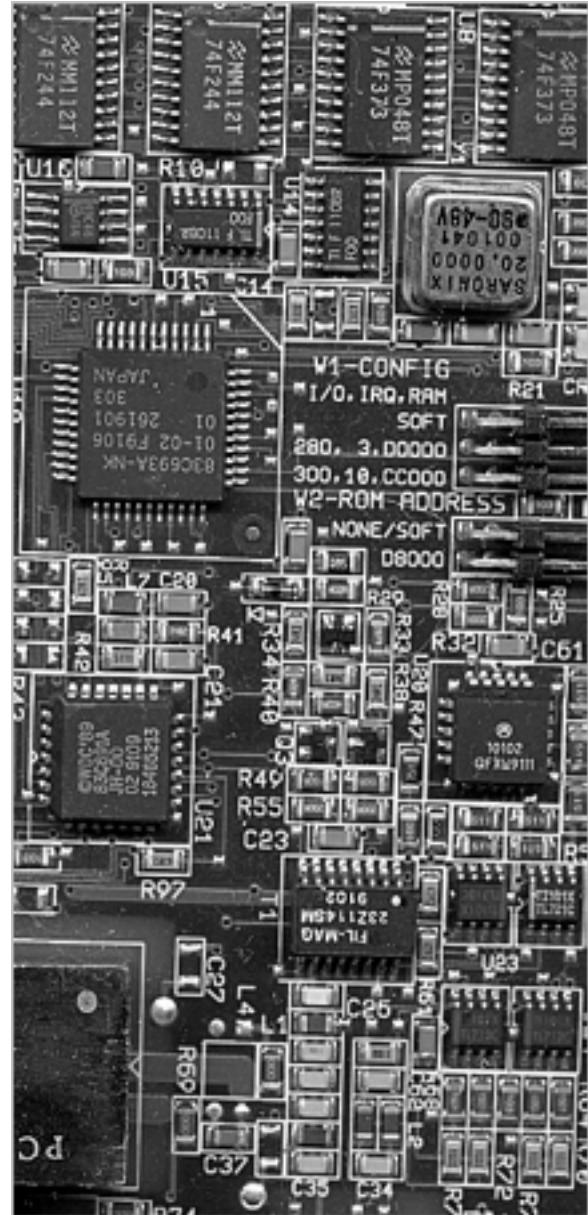


(d) Dynamic Range Adjusted

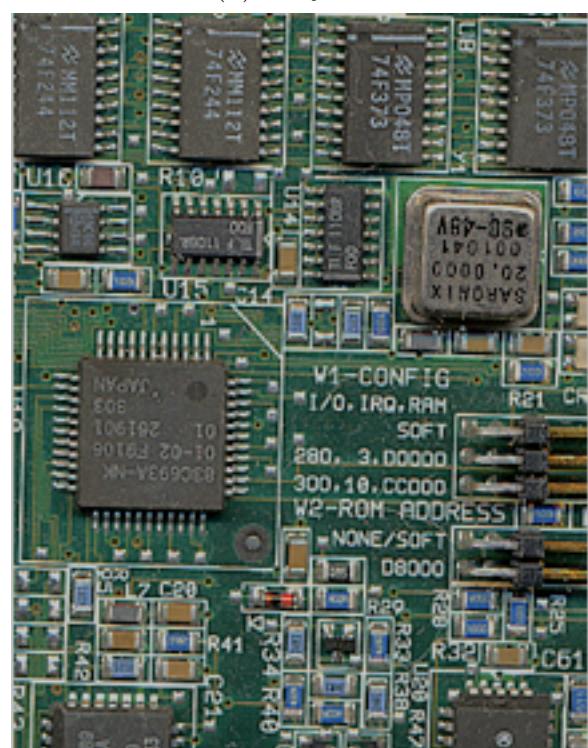
5.3 board.tif



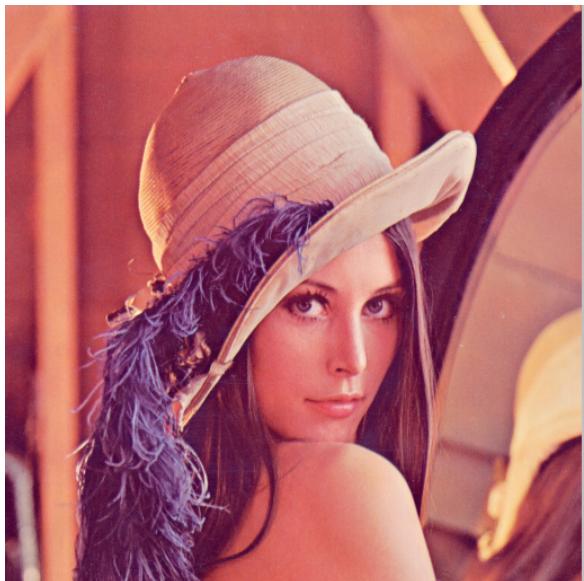
(a) Original



(b) Grayscale



5.4 lena.tif



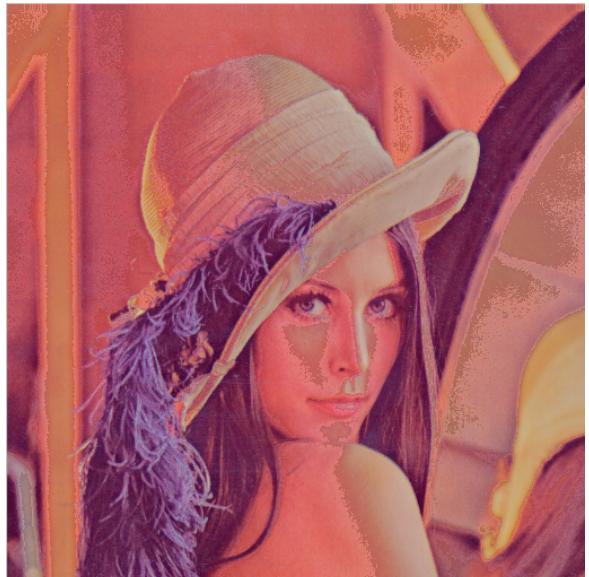
(a) Original



(b) Grayscale



(c) Order Dithering



(d) Dynamic Range Adjusted

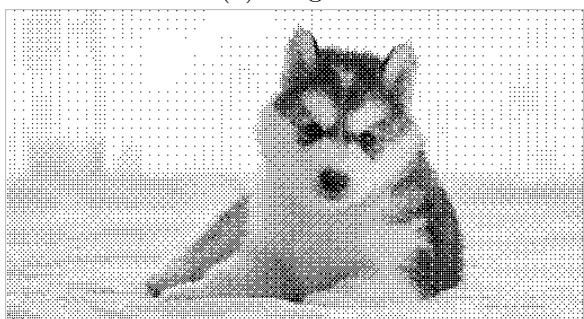
5.5 husky.tif



(a) Original



(b) Grayscale



(c) Order Dithering



(d) Dynamic Range Adjusted

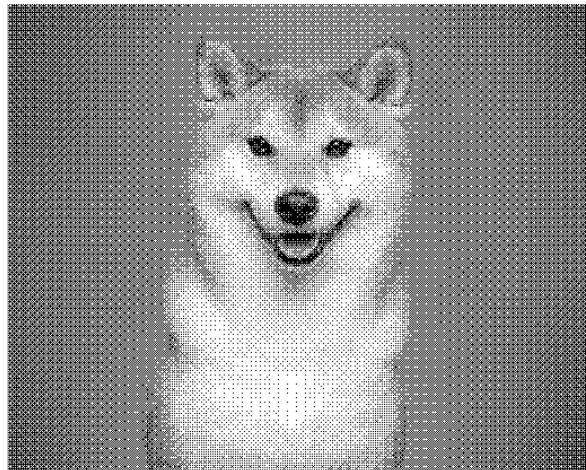
5.6 inu.tif



(a) Original



(b) Grayscale

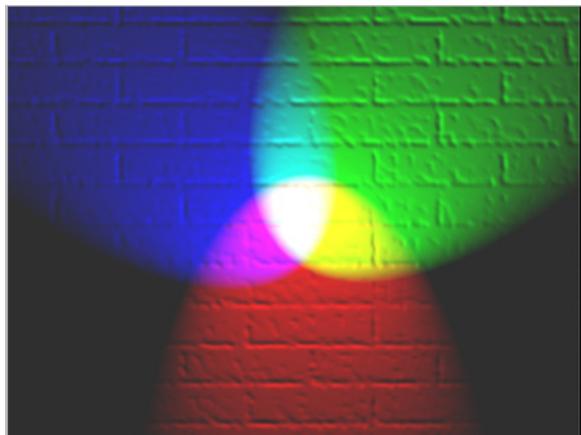


(c) Order Dithering

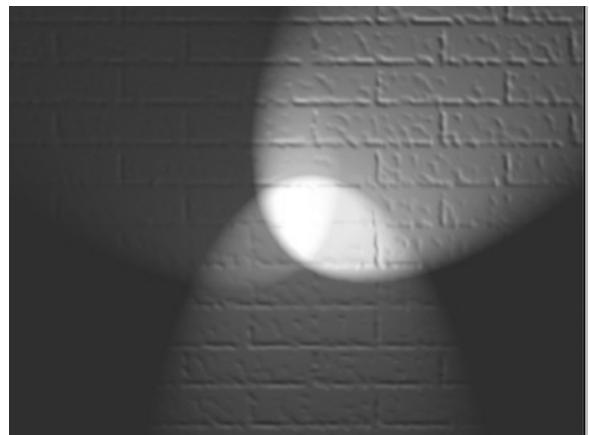


(d) Dynamic Range Adjusted

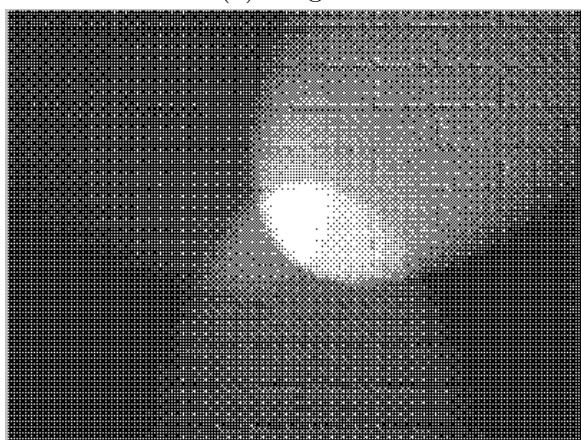
5.7 RGB.tif



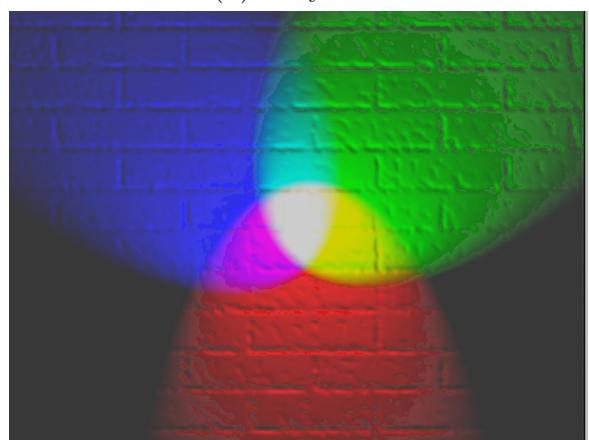
(a) Original



(b) Grayscale



(c) Order Dithering



(d) Dynamic Range Adjusted