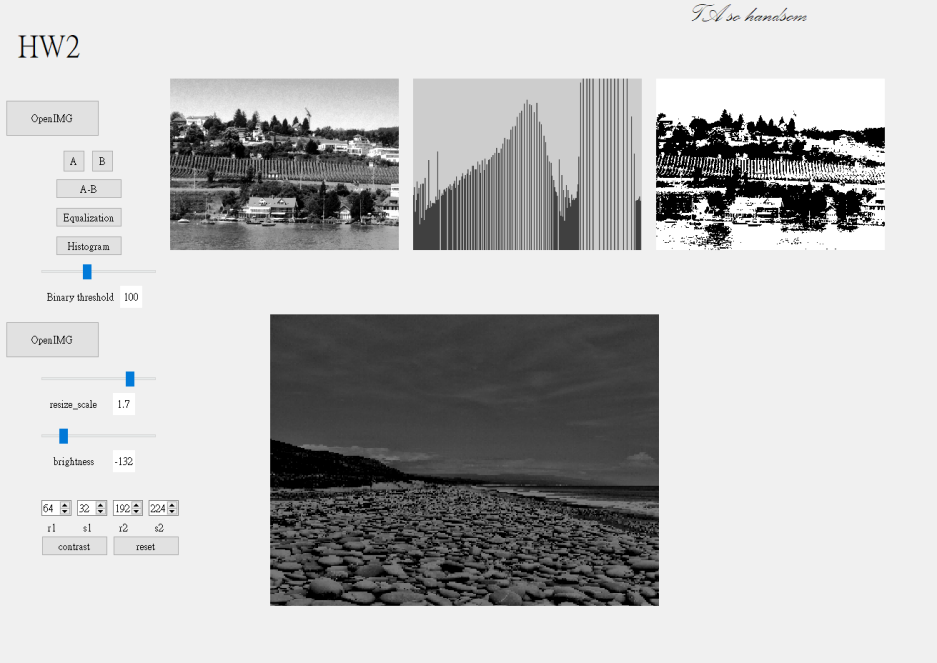
Image processing HW2

Name:武敬祥

ID:b06611032

GUI



Part 1

2.5

You are preparing a report and have to insert in it an image of size 2048 × 2048 pixels.

(a) $\frac{2048\ pixels}{50\ mm} = 40.96\ {}^{pixels}/_{mm}$

(b) $\frac{2048\ pixels}{2\ inches} = 1024\text{dpi}$

2.12

Assume for simplicity that the reflectance of the area is constant and equal to 1.0, and let *K* = 255. If the intensity of the resulting image is quantized using *k* bits, and the eye can detect an abrupt change of eight intensity levels between adjacent pixels, what is the highest value of *k* that will cause visible false contouring?

$$i(x,y) = Ke^{-[(x-x_0)^2 + (y-y_0)^2]}$$

(x_0, y_0) is the center of this image. $\mathrm{i}(\mathrm{x}, \mathrm{y})$ is distribution of light intensity

The K=255 so that the range of I is in 0 to 255. If we want to quantize the intensity with k bits, which means that there will be 2 to the kth power (2^k) gray levels. When a gray level contains 8 intensity levels, the eye will detect the contour. As a result, we want to make a gray level contains less than 8 intensity levels.

$$\frac{256}{2^k} < 8$$

$$32 < 2^k$$

when k is 6, it won't cause a visible false contouring. As a result, the maximize value of k to cause a false contouring is **5**.

2.18

Find the shortest length of 4- 8- m- path in the case of V={0,1} and V={1,2}. If a particular path doesn't exist, then explain why.

3	1	2	1 (q)
2	2	0	2
1	2	1	1
(p) 1	0	1	2

V={0,1}

4-path : not exist

8-path : 4

m-path : 5

V={1,2}

4-path : 6

8-path : 4

m-path : 6

the reason that 4-path cannot be found in case V={0,1} is a lack of 0 or 1 pixel value in $N_4(q)$

2.36

provide single, composite transformation functions for performing the following operations:

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to −1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

(a)

$$\begin{bmatrix} C_x & 0 & f_x \\ 0 & C_y & f_y \\ 0 & 0 & 1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} C_x cos\theta & -C_y sin\theta & f_x cos\theta - f_y sin\theta \\ C_x sin\theta & C_y cos\theta & f_x sin\theta - f_y cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

(c)

$$\begin{bmatrix} C_x cos\theta & C_x S_v cos\theta - C_y sin\theta & f_x cos\theta - f_y sin\theta \\ C_x sin\theta & C_y cos\theta + C_x S_v sin\theta & f_x sin\theta - f_y cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

(d)

Yes, it does if we change the order of multiplication on case (a). The result becomes

$$\begin{bmatrix} C_x & 0 & C_x f_x \\ 0 & C_y & C_y f_y \\ 0 & 0 & 1 \end{bmatrix}$$

3.12

3.21

$$w = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad f = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) Give the convolution of the two

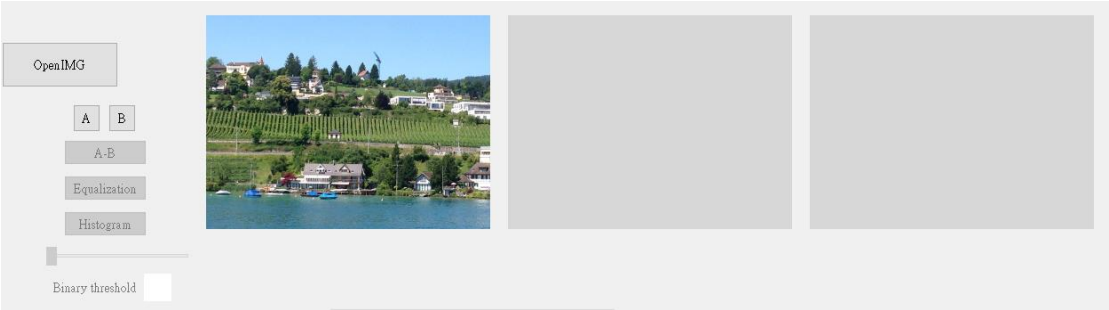
$$\begin{bmatrix} 9 & 12 & 12 & 12 & 9 \\ 12 & 16 & 16 & 16 & 12 \\ 12 & 16 & 16 & 16 & 12 \\ 12 & 16 & 16 & 16 & 12 \\ 9 & 12 & 12 & 12 & 9 \end{bmatrix}$$

(b) Does your result have a bias?

Yes, it does.

Part 2: Image File Reading, Display and Basic Processing

1. Read a color BMP or JPEG image file and display it on the screen. You may use the functions provided by Qt, OpenCV, or MATLAB to read and display an image.

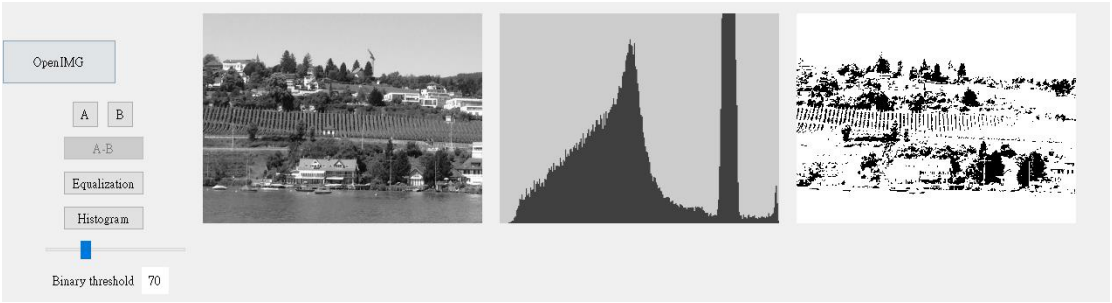


2. Convert a color image into a grayscale image using the following equations and compare it:

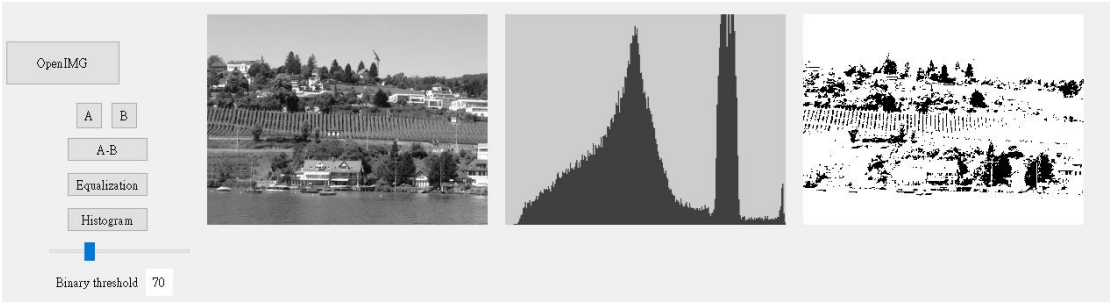
   A. GRAY = (R+G+B)/3.0

   B. GRAY = 0.299*R + 0.587*G + 0.114*B

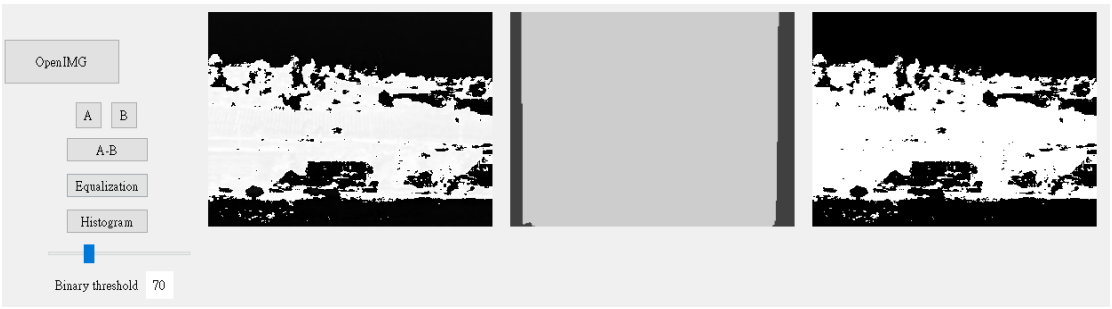
A.



B.



A-B:compare



3. Determine and display the histogram of a grayscale image.

   Please see the middle image of second question.

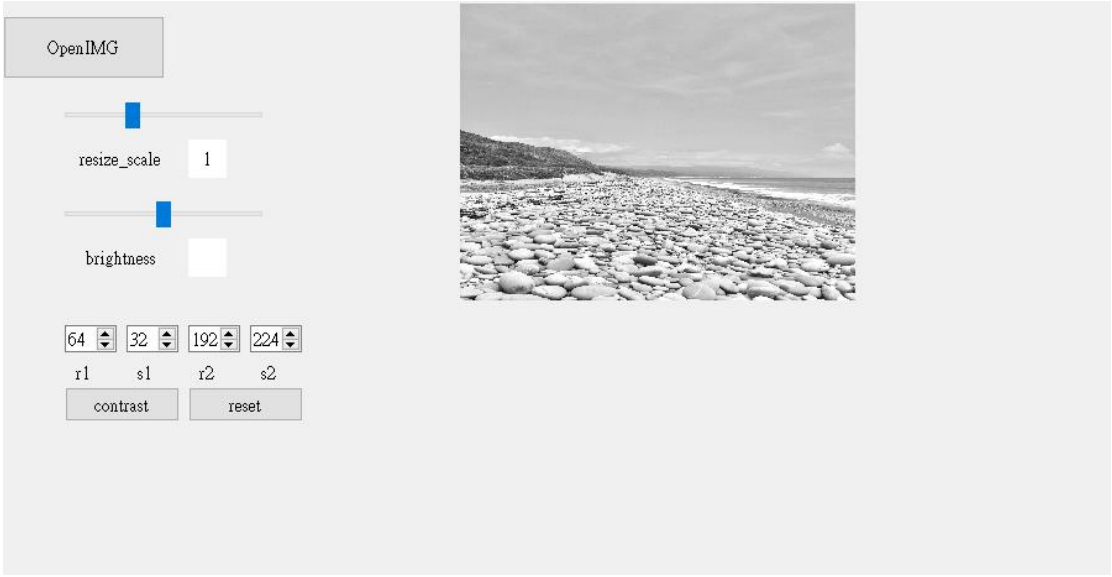
4. Implement a manual threshold function to convert a grayscale image into a binary image.

   Please see the right image of second question.

5. Implement a function to adjust the spatial resolution (enlarge or shrink) and grayscale levels of an image. Use interpolation on enlarging the

image.

Magnification = 1



Magnification = 2

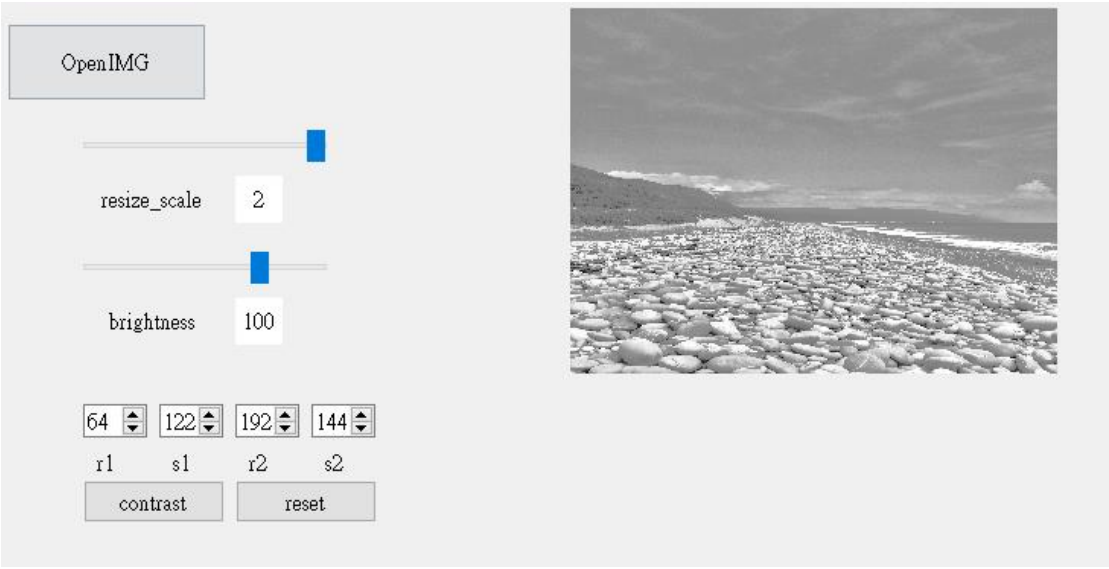
Drag the resize_scale slider to change the magnification of the image. The range of magnification is between 0.5 and 2. In this demonstration, I choose to implement the “nearest neighbor interpolation” method mentioned in the textbook

6. Implement a function to adjust the brightness and contrast of an image.

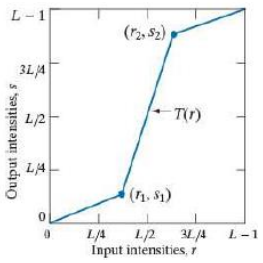
brightness

Drag the brightness slider to change the value of brightness. This is a demonstration of an image after adding 100 to every pixel.

Contrast



According to the following contrast equation on the textbook. We can arbitrary adjust the position of (r_1, s_1) and (r_2, s_2) by change the value showed in the spinBox object. After we set our value done, pressing the "contrast" button to show the result. Additionally, if we want to initialize our setting, just pressing "reset" button. The initial value is that $(r_1, s_1) = (64,32)$ and $(r_2, s_2) = (192,224)$ (please ignore the value of brightness….)



7. Implement a histogram equalization function for automatic contrast adjustment.

Before histogram equalization

After histogram equalization