CPS 843 Assignment 1

Xiaoxin Zhou
Computer Science
Ryerson university
Toronto, CA
Xiaoxin.zhou@ryerson.ca

Abstract: This work is for Assignment 1.

Introduction:

First, we will compare the power-law transformation on the intensity image with γ=0.3 and γ=3. We will write the equation for log, inverse log, and power-law transformations equations. To use eight-bit-plane slicing results to reconstruct an image from the highest 2 and 4 bit-planes. We compute the histogram of the three images from power-law transformation. Describe the process of histogram matching. And compute the histogram of the image and manually perform histogram equalization. Second, we do the padding and shearing example in MATLAB Image Processing Toolbox.

Problem 1:

The testing image is:

The gray image is:

Constant = 1 and γ = 0.3:

Constant = 1 and γ = 3.0:

γ means the intensity, if γ = 1, which is default, the mapping is linear. If the γ < 1, the mapping is weighted toward higher (brighter) output values. If γ > 1, the mapping is weighted toward lower(darker)output value [1].

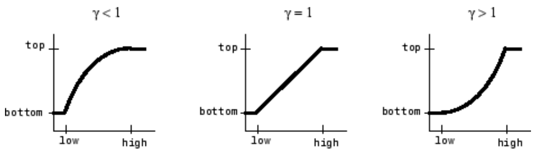


Figure 1: Three different Gamma correction Settings [1].

Equation of log, inverse log, and power-law transformations:

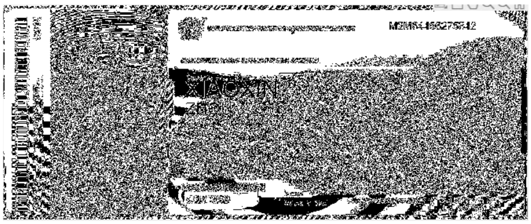
log_transformations = constant * log(1+IMAGE)

inverse_log_transformations = exp (IMAGE / constant)-1

power_law_transforamtions1 = constant * (IMAGE. ^ γ)

Problem 2:

Bit plane – 0



Bit plane – 1:

Bit plane – 2:

Bit plane – 3:

Bit plane – 4:

Bit plane – 5:

Bit plane – 6:

Bit plane – 7:

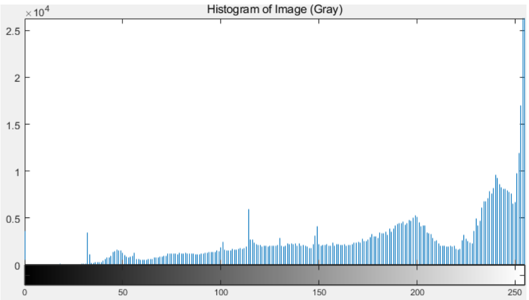
Highest 2 – bit reconstructed image:

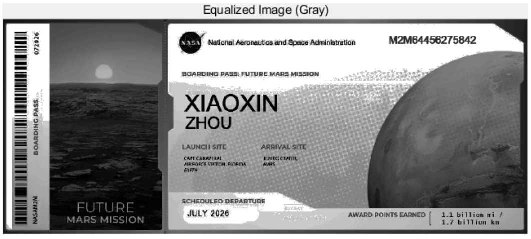
Highest 4 – bit reconstructed image:

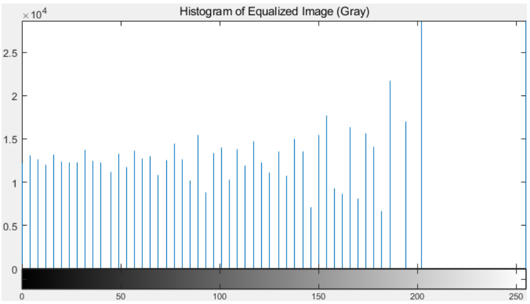
The lowest four-bit plane can be removed from the layer since the highest two or four is enough to perform the images. The different between highest two and four layer is the highest 4 layer can provide a strong contrast and intensity.

Problem 3:

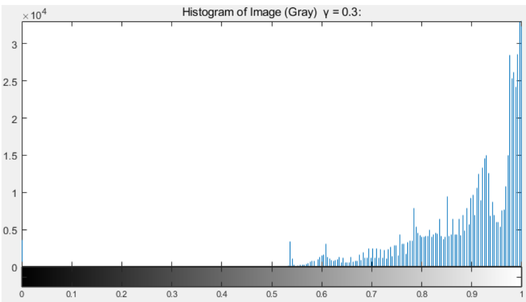
For original gray image:

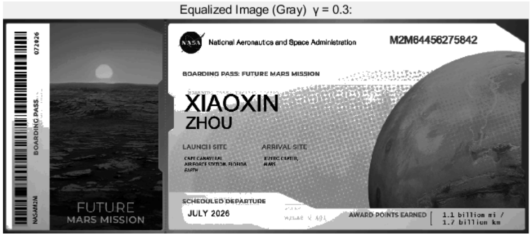


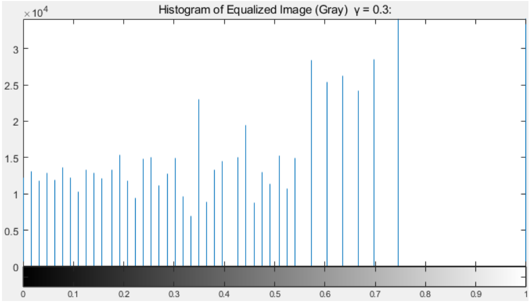




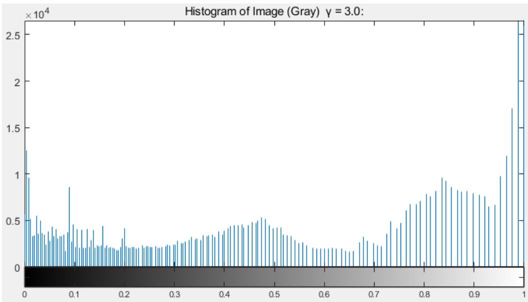
For Constant = 1 and γ = 0.3:

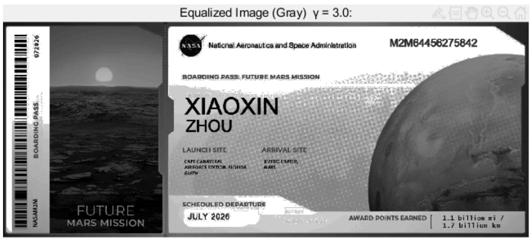


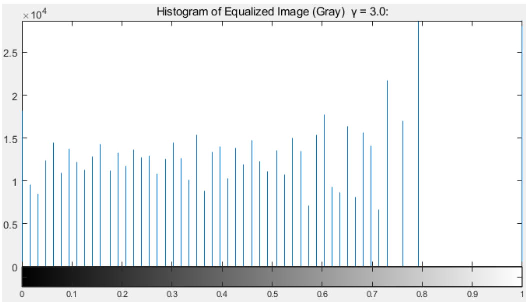




For Constant = 1 and γ = 3.0:







The histogram equalization makes the image have more contrast. By looking at the Log transformation 0.3, 3.0 and original equalization, the result of the three images is close, and the histogram is almost the same.

Problem 4:

We compute the histograms by using the cumulative distribution functions. One way to approach converts the discrete-valued image into a continuous-valued image and adds small random values to each pixel. So, the output image must contain the noise. There may have holes or open spots in the output matched histogram. The original histogram is different from the matched one.

Problem 5:

Original $\longrightarrow$ equalization

$$
\begin{matrix} 1 & 2 & 4 & 7 & 3 \\ 2 & 4 & 7 & 3 & 1 \\ 5 & 6 & 2 & 1 & 1 \\ 4 & 7 & 7 & 1 & 1 \end{matrix}
\qquad
\boxed{\begin{matrix} 3 & 4 & 5 & 7 & 4 \\ 4 & 5 & 7 & 4 & 3 \\ 6 & 6 & 4 & 3 & 3 \\ 5 & 7 & 3 & 3 & 3 \end{matrix}}
$$

	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Pixel intensity	1	2	3	4	5	6	7	8
No of pixels	7	3	2	3	1	1	3	0
Probability	0.35	0.15	0.1	0.15	0.05	0.05	0.15	0
Cumulative probability	0.35	0.5	0.6	0.75	0.8	0.85	1	1

we have 7 Pixel.

$S_0 = T(r_0) = 7\sum_{j=0}^{0} P_r(r_j) = 7P_r(r_0) = \cancel{2.45}$

$S_1 = T(r_1) = 7\sum_{j=0}^{1} P_r(r_j) = 7P_r(r_0) + 7P_r(r_1) = \cancel{2.45 + (0.15)7 = 3.5 \approx 4}$

$S_2 = T(r_2) = 7\sum_{j=0}^{2} P_r(r_j) = \cancel{3.5 + (0.1)7 = 8.4} \approx 8$

$S_3 = T(r_3) = 7\sum_{j=0}^{3} P_r(r_j) = \cancel{8.4 + (0.15)7 = 9.45} \approx 10$

$S_4 = T(r_4) = 7\sum_{j=0}^{4} P_r(r_j) = \cancel{9.45 + (0.05)7 = 9.8} \approx 10$

$S_5 = T(r_5) = 7\sum_{j=0}^{5} P_r(r_j) = \cancel{9.8 + (0.05)7 = 10.15 \approx 10}$

$S_6 = T(r_6) = 7\sum_{j=0}^{6} P_r(r_j) = \cancel{10.15 + (0.15)7 = 11.2 \approx} 11$

$S_0 = 0.35 \times 7 = 2.45 \approx 3$

$S_1 = 0.5 \times 7 = 3.5 \approx 4$

$S_2 = 0.6 \times 7 = 4.2 \approx 4$

$S_3 = 0.75 \times 7 = 5.25 \approx 5$

$S_4 = 0.8 \times 7 = 5.6 \approx 6$

$S_5 = 0.85 \times 7 = 5.95 \approx 6$

$S_6 = 1 \times 7 = 7 \approx 7$

Part 2 is provided in code.

Resource:

[1] MATLAB. Gamma Correction – MATLAB & Simulink. 2021, from https://www.mathworks.com/help/images/gamma-correction.html