

CPS843 Assignment 1

Udbhav Prasad
500909034

Abstract—First Assignment for Intro to Computer Vision (CPS843).

PART 1 - PROBLEM 1

Figure 1 contains the original gray scale image used in the various transformations.



Fig. 1. Original Gray scale Image

A. Log Transformation

Figure 2 contains the gray scale image after its been log transformed.

r is the input pixel, s is the output pixel, and c is a constant.

$$s = c \log(1 + r) \quad (1)$$



Fig. 2. Log Transformation Image

The log transformation brightens the dark areas of the image. It stretch's the low intensity areas and compresses the high intensity areas. Here we can see the dark areas under the tables are more visible.

B. Inverse Log Transformation

Figure 3 contains the gray scale image after its been inverse log transformed.

r is the input pixel, s is the output pixel, and c is a constant.

$$s = c \log^{-1}(r) \quad (2)$$



Fig. 3. Inverse Log Transformation Image

The inverse log transformation darkens the dark areas of the image. It stretch's the high intensity areas and compresses the low intensity areas. Here we can see the dark areas under the tables are even less visible.

C. Power Law Transformation

Figure 4 contains the gray scale image after its been power law transformed with gamma as 3.

Figure 5 contains the gray scale image after its been power law transformed with gamma as 0.3.

r is the input pixel, s is the output pixel, c and gamma is a constant.

$$s = cr^\gamma \quad (3)$$

With gamma as 3 we see the transformation work similar to inverse log transformation. With gamma as 0.3 we see the transformation work similar to the log transformation.

PART 1 - PROBLEM 2

In figure 6, the 8-bit place slicing is shown for the image.

The highest 4-bit and 2-bit planes reconstruction images are show in figure 7 and 8 respectively.

The 4-bit reconstruction shows more detail than the 2-bit reconstruction. Also, the 4-bit reconstruction is more discernible than the 2-bit reconstruction image.



Fig. 4. Power law Transformation Image (gamma is 3)



Fig. 5. Power law Transformation Image (gamma is 0.3)

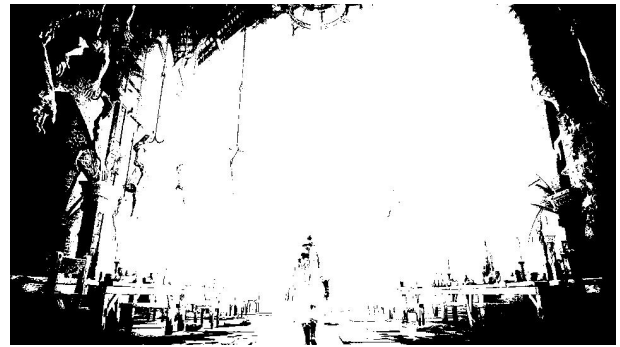


Fig. 7. Highest 4-bit Reconstruction Image



Fig. 8. Highest 2-bit Reconstruction Image

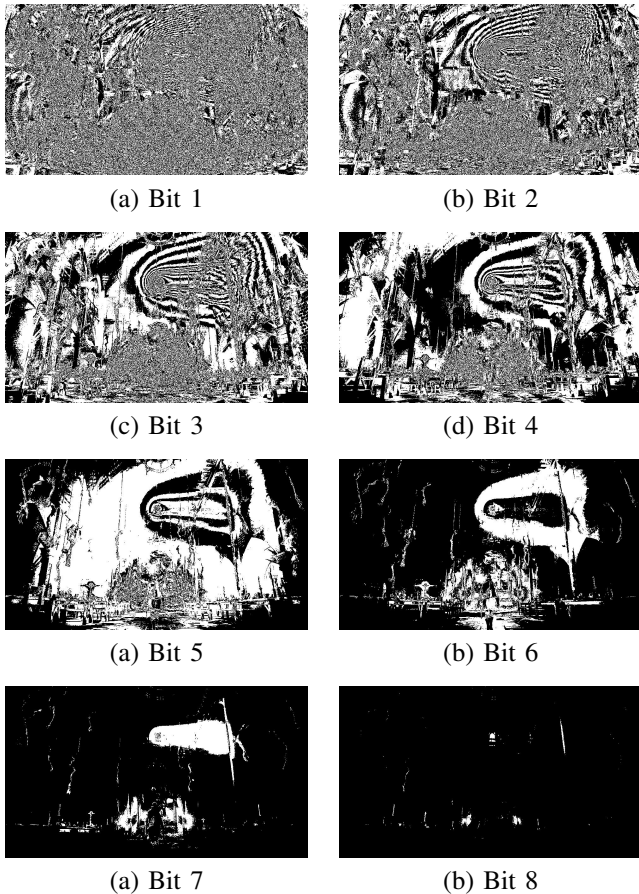


Fig. 6. 8-Bit Place Slicing

PART 1 - PROBLEM 3

Figure 9 shows the histograms of the original image and the power-law images, before and after they are histogram equalized.

Figure 9a is the image histogram before equalization and Figure 9b is the image histogram after equalization. Before the image is clustered to lower intensities and after equalization it is more spread out.

Figure 9c is the power law transformed(gamma=0.3) image histogram before equalization and Figure 9d is the power law transformed(gamma=0.3) image histogram after equalization. Before equalization the image has the intensities clustered in the middle and after it is spread out just like 9b.

Figure 9e is the power law transformed(gamma=3) image histogram before equalization and Figure 9f is the power law transformed(gamma=3) image histogram after equalization. Before equalization the image is very dark hence the it has very low intensity. After equalization, the image has higher intensity and is more spread out.

PART 1 - PROBLEM 4

Histogram matching is a transformation of an image to make its histogram match another histogram.

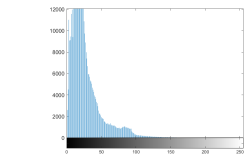
The procedure for histogram matching is as follows:

- First we get the probability distribution of the input data $p_r(r_k)$

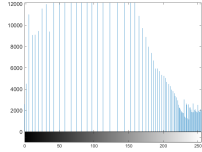
PART 1 - PROBLEM 5

The histogram before equalization is shown in Figure 10, while the histogram after equalization is shown in Figure 11. The calculation of histogram equalization is as follows:

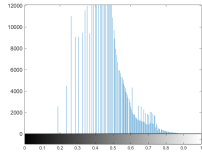
r_k	n_k	$p_r(r_k) = n_k/MN$	s_k	Rounded s_k
$r_0 = 0$	0	0	$s_0 = 0$	$s_0 = 0$
$r_1 = 1$	7	0.35	$s_1 = 2.45$	$s_1 = 2$
$r_2 = 2$	3	0.15	$s_2 = 3.50$	$s_2 = 4$
$r_3 = 3$	2	0.10	$s_3 = 4.20$	$s_3 = 4$
$r_4 = 4$	3	0.15	$s_4 = 5.25$	$s_4 = 5$
$r_5 = 5$	1	0.05	$s_5 = 5.60$	$s_5 = 6$
$r_6 = 6$	1	0.05	$s_6 = 5.95$	$s_6 = 6$
$r_7 = 7$	3	0.15	$s_7 = 7.00$	$s_7 = 7$



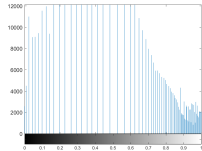
(a) Before-Original Image



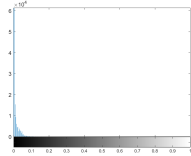
(b) After-Original Image



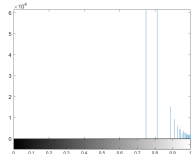
(c) Before-Power Law (gamma=0.3)



(d) After-Power Law (gamma=0.3)



(e) Before-Power Law (gamma=3)



(f) After-Power Law (gamma=3)

Fig. 9. Before and After Histogram Equalization

- Second, do Histogram Equalization on input $s = T(r)$, where T is the histogram equalization.
- Thirdly, we compute $s = G(z)$.
- After, we get the inverse mapping $z = G^{-1}(s)$.
- The output image with values z is the histogram matched image.

You will not obtain the same histogram as the desired one after histogram matching because of rounding to discrete values that happens during histogram equalization.

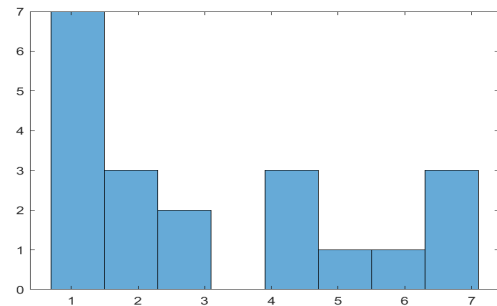


Fig. 10. Before Histogram Equalization

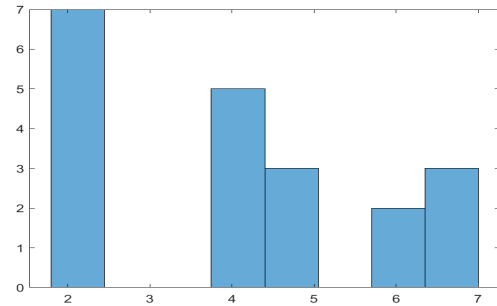


Fig. 11. After Histogram Equalization

Before Histogram Equalization:

1	2	4	7	3
2	4	7	3	1
5	6	2	1	1
4	7	1	1	1

After Histogram Equalization:

2	4	5	7	4
4	5	7	4	2
6	6	4	2	2
5	7	2	2	2

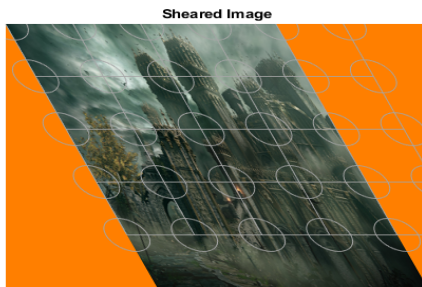
PART 2

Explanation of each step for padding and shearing an image:

- **Step 1:** We use simple shear to transform the image. Simple shear is an affine transformation, we make an affine with maketform with the simple shear matrix values. What shear does is it moves a point further away from an axis, the more further away it is from another axis, which creates a bent rotated like shape.



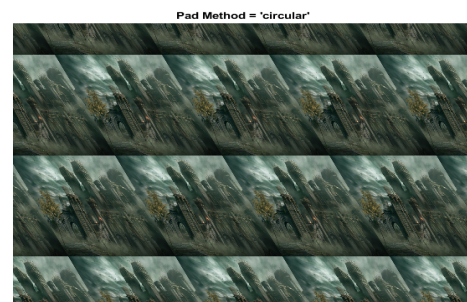
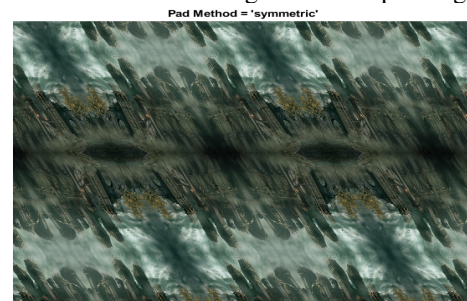
- **Step 2:** Here we draw lines and circles to examine how the simple shear transformation affected these shapes. We can see the circles and lines have been bent according to the shear.



- **Step 3:** We compare different padding options, for "fill" we simply fill with a orange static color. Replicate padding makes a copy of the input, reverses it and applies that as the padding. Bound padding applies a strict distinction between the padding and the input. Bound has clear boundaries compared to fill method.



- **Step 4:** Here we try two padding methods, circular and symmetric. The circular method takes copies of the image and replicates it. The symmetric method takes the mirror of the image and fills padding with it.



REFERENCES

- [1] Code for this assignment at ./a1.m
- [2] Assignment source code: <https://github.com/UdbhavPrasad072300/CPS843-Assignments>