

ECE1512 - Homework 1

Yang Wang
Student Number: 1001319227
Due Date: Sep 30, 2015
September 27, 2015

1 PART A

1.1 TECHNICAL DISCUSSION

Figure 1.1 shows a magnetic resonance image(MRI) of an upper thoracic human spine with a fracture dislocation and spinal cord impingement. Because the original image is predominantly dark, an expansion of intensity levels is desirable. This can be accomplished with Log Transformation and Power-Low(Gamma) Transformation.



Figure 1.1: Original Image

1.1.1 LOG TRANSFORMATIONS

The general form of the log transformation is

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

where r is the original pixel value between 0 and 1, s is the pixel value after log transformation and c is a constant. By observing the function we could find that this transformation maps a narrow range of low intensity values in the input into a wider range of output levels. We could use this kind of transformation to expand the values of dark pixels in the given MRI and compressing the higher-level values, which could show the details of the fractured part more obvious.

1.1.2 POWER-LAW(GAMMA) TRANSFORMATIONS

Power-law transformations have the basic form

$$s = cr^\gamma$$

Where where r is the original pixel value between 0 and 1 and s is the pixel value after transformation. c and γ are positive constants. By observing the equation we could find that power-law transformation also map a narrow range of dark input values into a wider range of output values. By analyzing, we could notice that when $\gamma > 1$, the produced image tend to be darker than the original image. However, when $\gamma < 1$, the produced image will show more details on dark area of original image. In the following experiments, we need to find appropriate c and γ to show more details of the given MRI.

1.2 DISCUSSION OF RESULTS

1.2.1 EXPERIMENTS OF LOG TRANSFORMATIONS

The experiments results of log transformation shows below: As shown in Figure 1.2b, the pro-

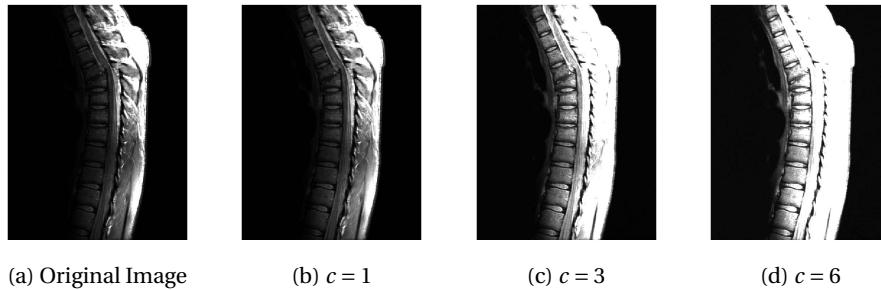


Figure 1.2: Experiments results of log transformation when $c = 1, 3$ and 6

duced image shows more details on the dark area compare with the given MRI. However, the fractured area on the dark side still not clear enough. When we have $c = 6$, we could find even

more details on the dark area as it shows in Figure 1.2d. But we could also notice that those area around the fractured point become totally white, which may result in missing of important information. As shown in Figure 1.2c When $c = 3$, we could have a relative desirable output image since it provides us enough information about the fracture also dark area.

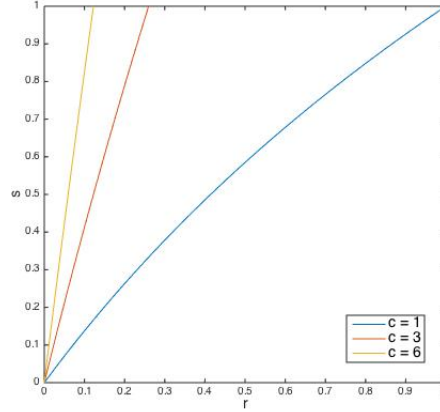


Figure 1.3: The curve of $c \log(r + 1)$ when $c = 1, 3$ and 6

The reason behind it could be found in Figure 1.3. As shown in the figure, we could notice that when $c = 1$ the output pixel value s just grow a little compare with original value r . This is why it shows a little bit more details but the fractured area is still not clear enough. We note that, as c increased to 3, the output pixel grows much higher than the original pixel value. So the produced image shows more details compare with $c = 1$. According to the curve above, s will equal to 1 when $r > 0.26$, which may result in missing some information but still at an acceptable level. However, as c increase from 3 to 6, although we could have more details on the darker area, any input pixel value $r > 0.11$ will map to 1 according to the curve. It means we will miss lots of detail informations on the lighter side of image by selecting $c = 6$. So, when $c = 3$ the output image could have best visual enhancement.

1.2.2 EXPERIMENTS OF POWER-LAW TRANSFORMATIONS

The experiments results of Power-Law(gamma) transformation shows in Figure 1.4.

Since the basic form of Power-law transformations is

$$s = cr^\gamma$$

By selecting c and γ , we can get the optimal parameters with best visual result.

As shown in 1.4b, when $c = 1$ and $\gamma = 1.5$, the produced image is darker than original image and area near fractured point became invisible. With $c = 1$ and γ decreased from 0.8 to 0.4 as shown in Figure 1.4c, 1.4d, 1.4e, more detaild became visible. As shown in Figure 1.4f A

further decrease of γ to 0.3 enhanced a little more detail in background but begin to reduce contrast to the point where the image started to have a slight washed-out appearance. If we decreased c to 0.5 and let $\gamma = 0.4$, the produced image became much darker than original image as shown in 1.4g. Similarly, As shown in Figure 1.4h, if $c = 1.5$ and $\gamma = 0.4$, although the produced image shows more details on the dark side compare with original image, image on the lighter side became totally white and missing some details. By comparing all results, produced image will have best visual result with $c = 1$ and $\gamma = 0.4$.

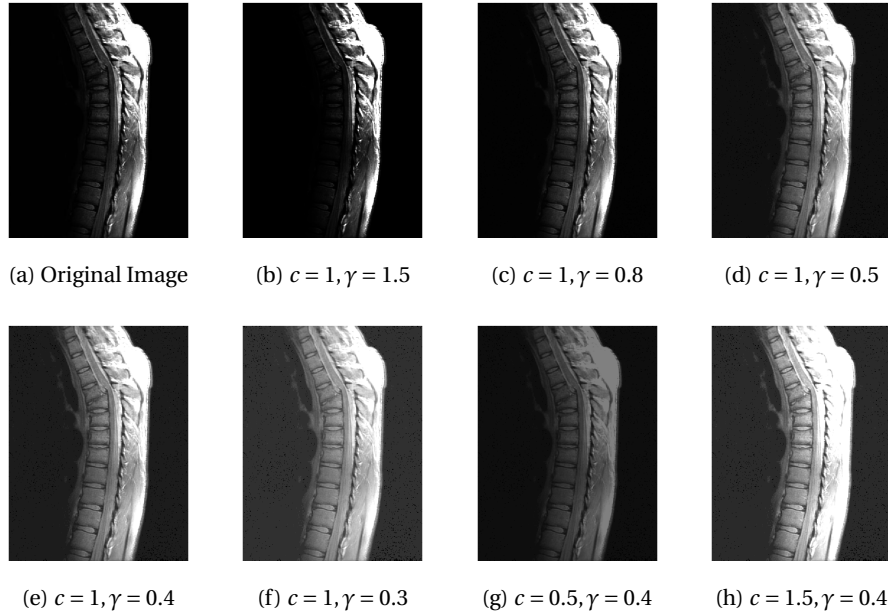


Figure 1.4: Experiments results of Power-Law transformation

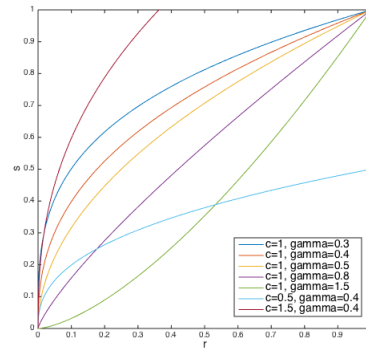


Figure 1.5: The curve of $s = cr^\gamma$

The reason behind it could be found in Figure 1.5. As shown in the figure, we could notice that with $c = 1$ and γ decreased from 0.8 to 0.3, output pixel value s will grow more compare

with the original input pixel value r . When $\gamma = 0.3$, we can note that the output pixel value will become around 0.5 even the input pixel value just around 0.1. This is the reason why it begin to reduce contrast to the point where the image started to have a slight washed-out appearance. When $c = 0.5$, the output value will in a range from 0 to 0.5. So, it makes the produced image darker than original image. Similarly, If we set $c = 1.5$, we could note that when input pixel value $r \geq 0.4$, the output pixel value will be 1. So the lighter side of the produced image became white. So, when $c = 1$ and $\gamma = 0.4$ the output image could have best visual enhancement.

2 PART B

2.1 TECHNICAL DISCUSSION

2.1.1 HISTOGRAM PROCESSING

The histogram of a digital image with intensity levels in the range $[0, L - 1]$ is a discrete function $h(r_k) = n_k$, where r_k is the k th intensity value and n_k is the number of pixels in the image with intensity r_k . It is common practice to normalize a histogram by dividing each of its components by the total number of pixels in the image, denoted by the product MN , where, as usual, M and N are the row and column dimensions of the image. Thus, a normalized histogram is given by $p(r_k) = n_k/MN$, for $k = 0, 1, 2, \dots, L - 1$. Loosely speaking, $p(r_k)$ is an estimate of the probability of occurrence of intensity level r_k in an image. The sum of all components of a normalized histogram is equal to 1.

2.1.2 HISTOGRAM EQUALIZATION

As we know, if we view input intensities r and output intensity s as random variables and their histograms as probability density functions(pdf) $p_r(r)$ and $p_s(s)$.

if $p_r(r)$ and $T(r)$ are known and $T(r)$ is continuous and differentiable, then,

$$p_s(s) = p_r(r) \frac{1}{\left| \frac{ds}{dr} \right|} = p_r(r) \left| \frac{dr}{ds} \right|$$

Cumulative distribution function of a random Variable:

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw$$

To find $p_s(s)$ we have to compute

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L - 1) \frac{d}{dr} \int_0^r p_r(w) dw = (L - 1) p_r(r)$$

Substituting this result:

$$\frac{ds}{dr} = (L - 1) p_r(r)$$

to

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|$$

yields

$$p_s(s) = p_r(r) \left| \frac{1}{(L-1)p_r(r)} = \frac{1}{L-1} \right|, 0 \leq s \leq L-1$$

The formula for histogram equalisation in the discrete case is given

$$s_k = T(r_k) = (L-1) \sum_{j=0}^k p_r(r_j) = \frac{L-1}{MN} \sum_{j=0}^k n_j$$

2.2 DISCUSSION OF RESULTS

2.2.1 COMPUTING THE HISTOGRAM OF AN IMAGE

The original image is shown in Figure 2.1 To get the histogram of given image, we need to



Figure 2.1: Original Image

iterate all pixel values of the image and get the frequency of each intensity. Figure 2.2 shows the output histogram of Figure 2.1 after normalisation. We can note that most pixel values are on the dark side.

2.2.2 IMPLEMENTATION OF HISTOGRAM EQUALIZATION

To perform histogram equalization on given image, we could use the formula for histogram equalization in the discrete case

$$s_k = T(r_k) = \frac{L-1}{MN} \sum_{j=0}^k n_j$$

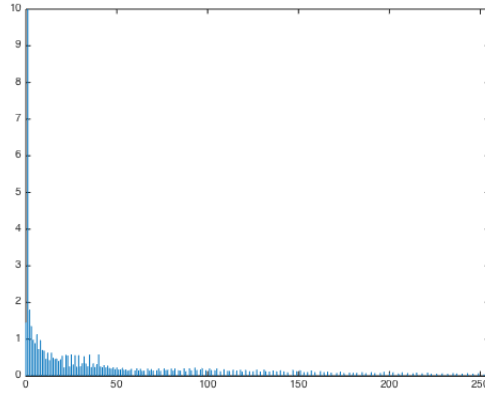


Figure 2.2: Histogram of Figure 2.1

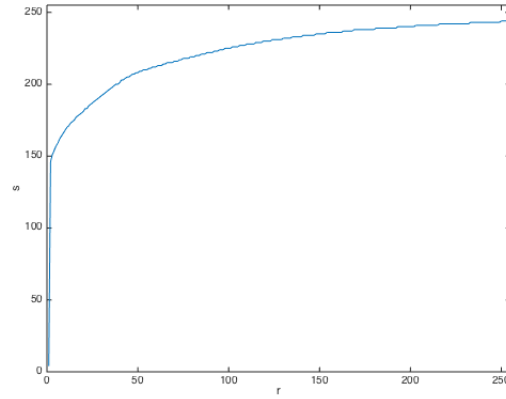
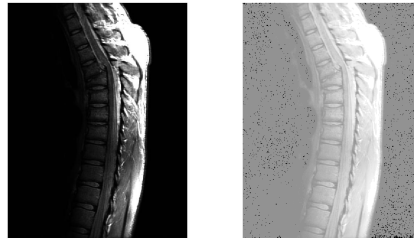


Figure 2.3: Transformation function

Figure 2.3 shows the transformation function. Figure 2.4 shows the comparison between the original image and the image after histogram equalization. We can notice that the produced image is much lighter than original image and we could get more details on the darker side of the image. However, the background became too light after histogram equalization, which reduce the contrast of image. Figure 2.5 shows the histogram of the produced image.

The reason of the background became too light can be described as follows. As we can see from the histogram of original image in Figure 2.2, most of the pixel values in orinal image are concentrated neat 0. According to the formula for histogram equalization in the discrete case $s_k = T(r_k) = \frac{L-1}{MN} \sum_{j=0}^k n_j$, if there are too many pixel values nears 0, then the output value s will grow a lot for those input pixels with value near 0. As a result, the produced image have a

washed-out appearance.



(a) Original Image (b) Image after histogram equalization

Figure 2.4: Experiments results of histogram equalization

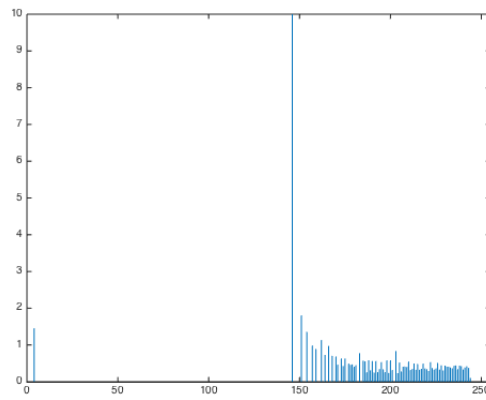


Figure 2.5: Histogram of produced image

3 APPENDIX

3.1 PROGRAM OF LOG TRANSFORMATION

```
%*****  
%Student Name: Yang Wang  
%Student #: 1001319227  
%Title: Log Transformation  
%*****  
clear;  
clc;  
cla;  
c = 3;  
G = imread('0308.tif');  
[rows, cols] = size(G);  
for i = 1 : rows  
    for j = 1 : cols  
        G(i, j) = c * log2(double(G(i, j))/255.0 + 1) * 255.0;  
    end  
end  
figure();  
imshow(G);
```

3.2 POWER-LAW TRANSFORMATION

```
%*****  
%Student Name: Yang Wang  
%Student #: 1001319227  
%Title: Log Transformation  
%*****  
clear;  
clc;  
cla;  
c = 1;  
gamma = 0.4;  
G = imread('0308.tif');  
[rows, cols] = size(G);  
for i = 1 : rows  
    for j = 1 : cols  
        G(i, j) = c * 255 * (double(G(i, j))/255)^gamma;  
    end  
end  
figure(2);  
imshow(G);
```

3.3 PROGRAM OF COMPUTING HISTOGRAM

```
%*****
%Student Name: Yang Wang
%Student #: 1001319227
%Title: Computing Histogram of image
%*****

clear;
clc;
cla;
G = imread('0308.tif');
H = zeros(1, 256);
[rows, cols] = size(G);
for i = 1 : rows
    for j = 1 : cols
        H(1, G(i, j) + 1) = H(1, G(i, j) + 1) + 1;
    end
end
%show histogram of image after normalization
x = 0 : 255;
figure(1)
H = H * 100 / (rows * cols);
stem = stem(x, H);
set(stem, 'Marker', 'none');
axis([0 255 0 10]);
```

3.4 PROGRAM OF HISTOGRAM EQUALIZATION

```
%*****
%Student Name: Yang Wang
%Student #: 1001319227
%Title: Histogram Equalization
%*****

clear;
clc;
cla;
G = imread('0308.tif');
H = zeros(1, 256);
[rows, cols] = size(G);
for i = 1 : rows
    for j = 1 : cols
        H(1, G(i, j) + 1) = H(1, G(i, j) + 1) + 1;
```

```

    end
end
nums = rows * cols - 1;
collects = zeros(1, 256);
for i = 1 : 256
    if(i == 1)
        collects(i) = (double(H(i)) / nums) * 255;
    else
        collects(i) = (double(H(i)) / nums) * 255 + collects(i - 1);
    end
end
k = uint8(collects);
for i = 1 : rows
    for j = 1 : cols
        G(i, j) = k(1, G(i, j) + 1);
    end
end
%show image after HE
figure(1);
imshow(G);
x = 1 : 256;

%transformation function
figure(2);
s2 = plot(x, k);
xlabel('r');
ylabel('s');
axis([0 255 0 255]);

%histogram after HE
H = zeros(1, 256);
[rows, cols] = size(G);
for i = 1 : rows
    for j = 1 : cols
        H(1, G(i, j) + 1) = H(1, G(i, j) + 1) + 1;
    end
end
x = 0 : 255;

figure(3)
H = H * 100 / (rows * cols);
stem = stem(x, H);
set(stem, 'Marker', 'none');
axis([0 255 0 10]);

```