School of Data and Computer Science, Sun Yat-sen University

PROJECT 03-02
I - Histogram Equalization
and
II - Gamma Transformation

17341137 Zhenpeng Song

September 8, 2019

Abstract

Implementation of two experiments: I - Histogram Equalization to enhance images, and II - Gamma Transformation to adjust images.

1 Technical discussion

- I - Histogram Equalization

  First of all, I define some variables as:

  - k: the number of types of pixels of different value.
  - n_k, n: the number of pixels of different value.
  - r_k: the value of the pixel k.
  - $p_r(r_k)$: the existing probability of pixel k in all types of pixels.
  - $P(r_k)$: the cumulative distribution function.
  - $s_k = T(r_k)$: the transformation function.

  Histogram Equalization is conducted with the following steps:

  1. Probability: $p_r(r_k) = \dfrac{n_k}{n}$
  2. Cumulative distribution function: $P(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \dfrac{n_j}{n}$
  3. The transformation function: $s_k = T(r_k) = (L-1) \times \sum_{j=0}^{k} \dfrac{n_j}{n}$
  4. Take $\lceil s_k \rceil since pixel value is always an integer.$

- II - Gamma Transformation

  Gamma Transformation is used to improve image, in detail, to darken or lighten the image with different γ:

  1. $s = Cr^{\gamma}$
  2. In this term, I choose an image which is too dark, and try a series of γ to improve it.

2 Discussion

- I - Histogram Equalization

  - I did Histogram Equalization(noted as HE) with built-in function hsteq(), and got a satisfying result which is shown in Figure 1.
  - Then, I did HE with my own codes enclosed below, however, the result didn't satisfy me.
  - Analyzing the transforming function, I realize how it works and why my result took on a worse look.
  - After all, histogram equalization will definitely enhance the image.

- II - Gamma Transformation

  - Gamma Transformation(noted as GT) affects every pixel of the image, as a kind of intensity transformation, GT has its own properties.
  - For dark image, GT with $\gamma < 1$ will effectively lighten it, and with $\gamma > 1$ we can darken those light images.

3 Results

- I - Histogram Equalization

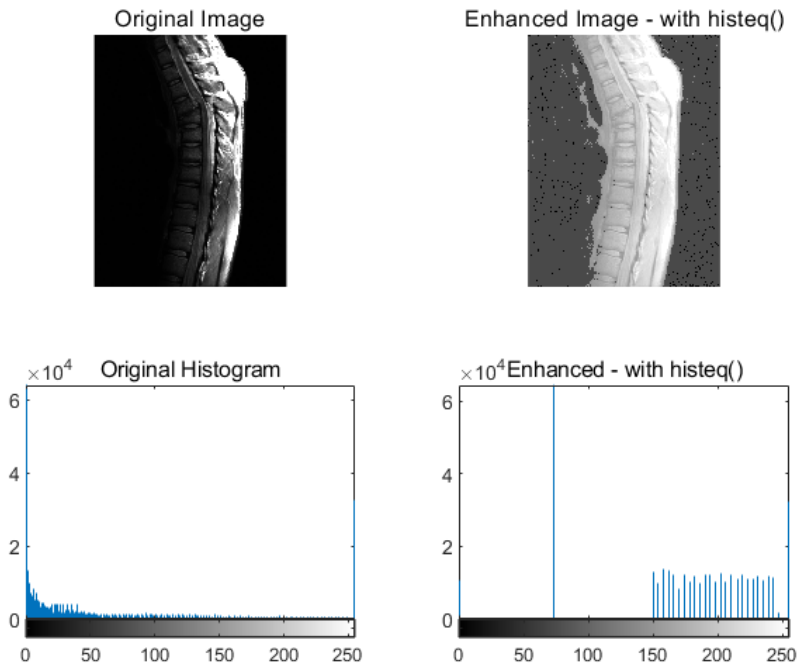


Figure 1: result with built-in func. histeq()

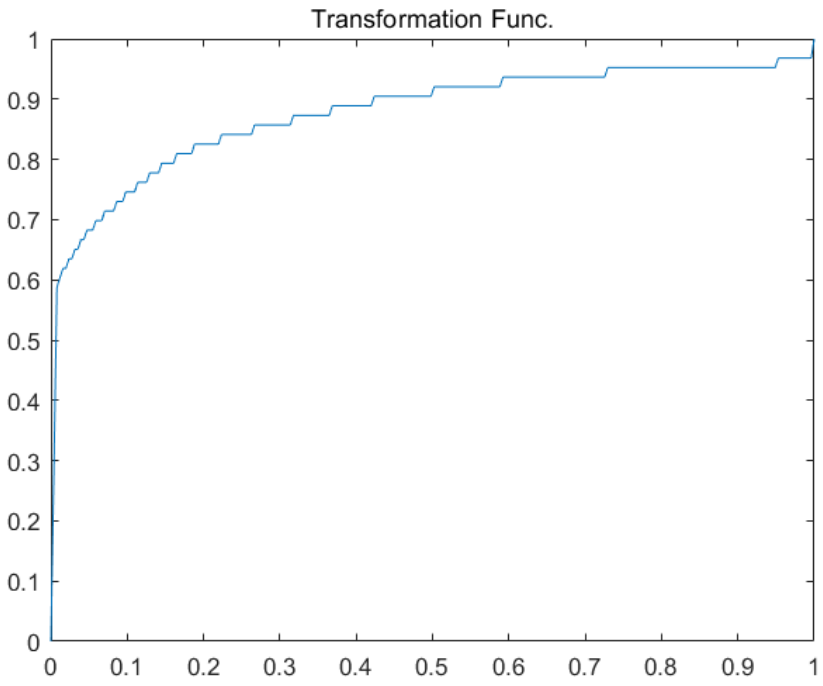


Figure 2: transformation func. With built-in func. histeq()

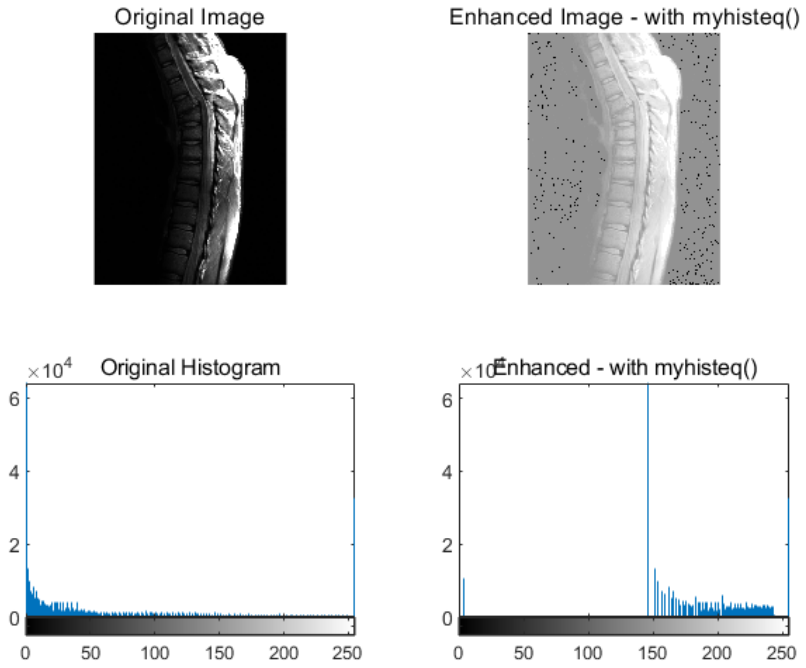


Figure 3: result with raw codes.

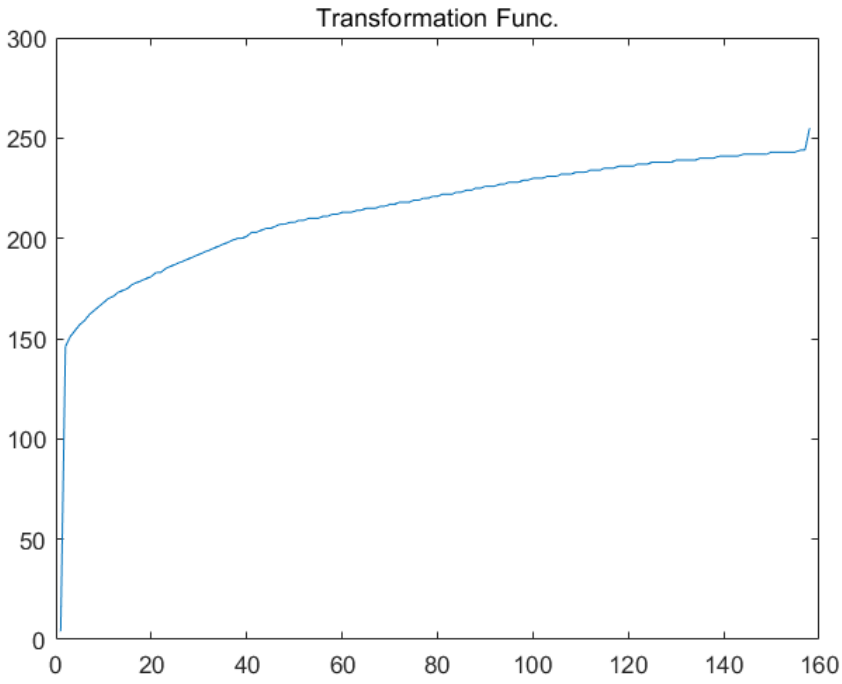
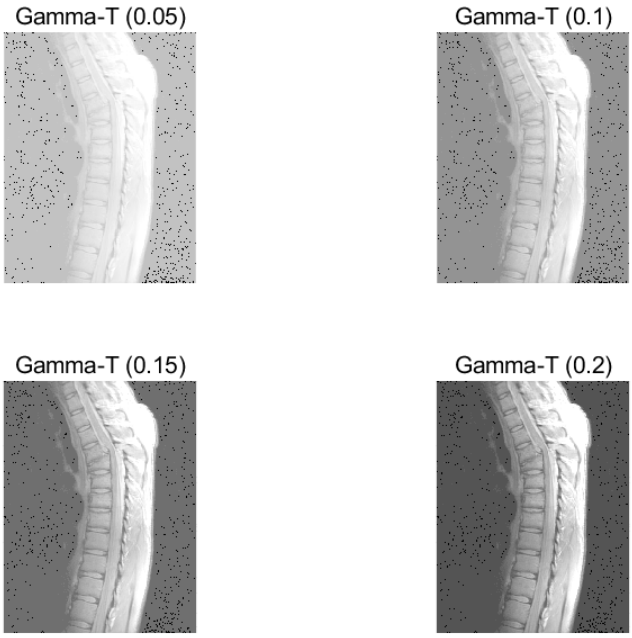


Figure 4: transformation func. with raw codes.

- II - Gamma Transformation

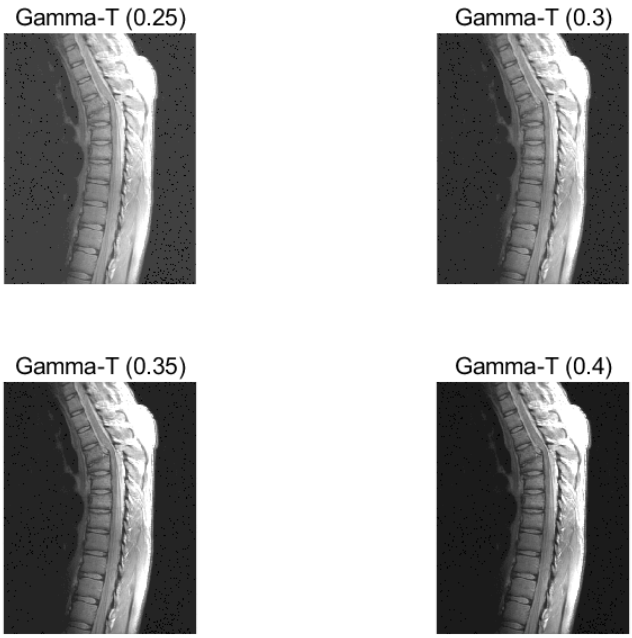
1.png

Figure 5: Original Image - Grey-scale Image.



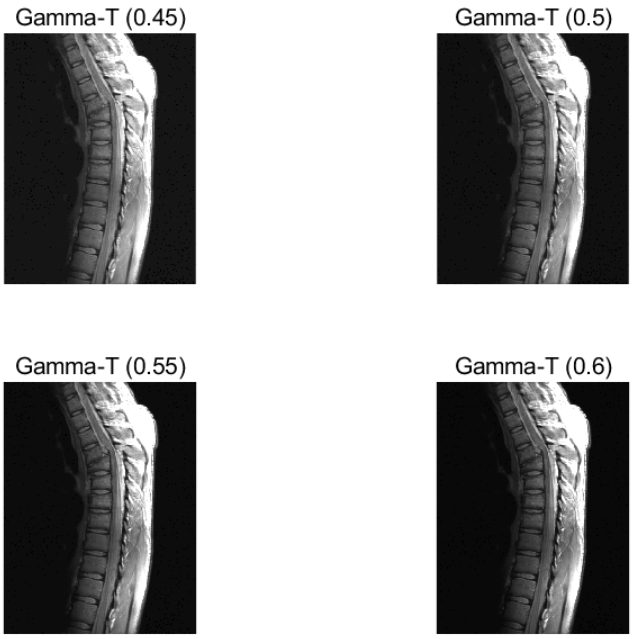
2.png

Figure 6: transformation func. with raw codes.



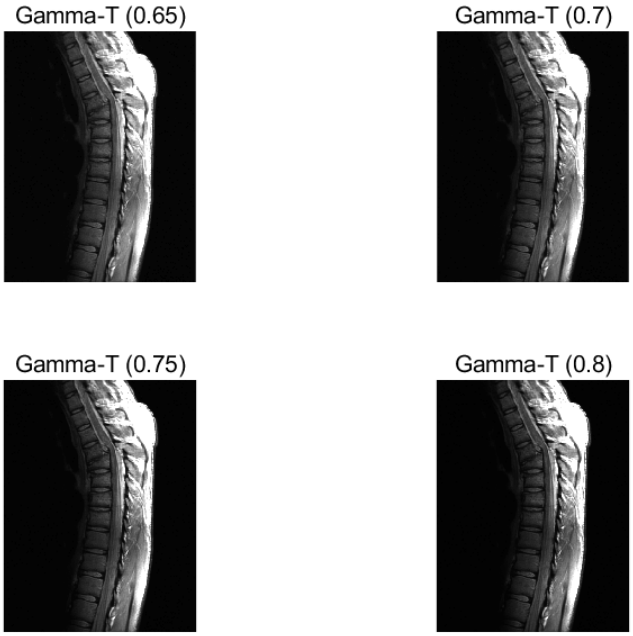
3.png

Figure 7: transformation func. with raw codes.



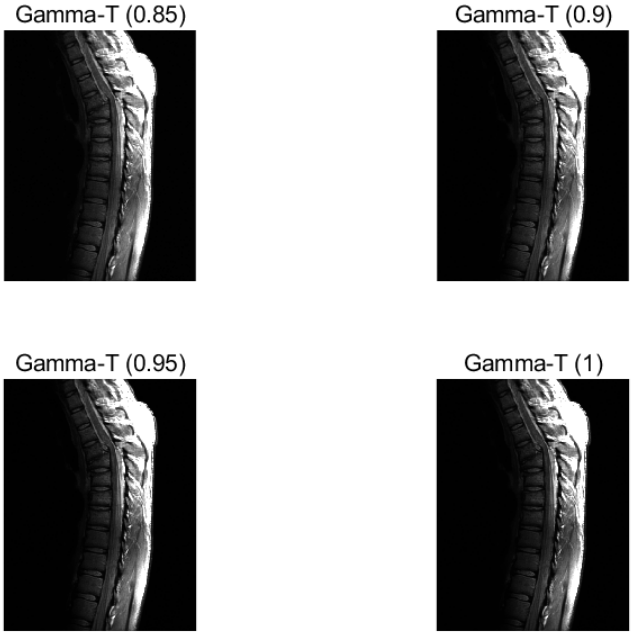
4.png

Figure 8: transformation func. with raw codes.



5.png

Figure 9: transformation func. with raw codes.



6.png

Figure 10: transformation func. with raw codes.

4 Appendix

- I - Histogram Equalization

```
% Function my_histeq()
function [en_img, s] = my_histeq(I)
    [m, n] = size(I);
    f_img = reshape(I, m * n, 1); % f for flattened
    sf_img = sortrows(f_img);
    [A, ib, ic] = unique(sf_img);

    % calculate p
    for i = 1:size(A)-1
        pxnum(i) = ib(i+1) - ib(i);
        p(i) = pxnum(i)/(m*n);
    end
    pxnum(i+1) = m*n - ib(i+1) + 1;
    p(i+1) = pxnum(i+1)/(m*n);

    % calculate P & s
    sum = 0;
    for i = 1:size(A)
        sum = sum + p(i);
        P(i) = sum;
        s(i) = floor(double(max(sf_img) - min(sf_img)) * P(i) + 0.5);
        % s(i) = floor(double(256 .* P(i) + 0.5));
    end

    % transformation
    for i = 1:m*n
        enf_img(i) = s(A==f_img(i));
    end
    en_img = uint8(reshape(enf_img, m, n));

    % Solution
    clear, clc;

    I = imread('./Fig0308(a)(fractured_spine).tif');
    [J, T] = histeq(I);
    figure(1);
    subplot(221); imshow(I), title('Original Image');
    subplot(223); imhist(I), title('Original Histogram');
    subplot(222); imshow(J), title('Enhanced Image - with histeq()');
    subplot(224); imhist(J), title('Enhanced - with histeq()');
    figure(2); plot([0:255]/255, T), title('Transformation Func.');


    [J, T] = my_histeq(I);
    figure(3);
```

```
subplot(221); imshow(I), title("Original Image");
subplot(223); imhist(I), title("Original Histogram");
subplot(222); imshow(J), title("Enhanced Image - with myhisteq()");
subplot(224); imhist(J), title("Enhanced - with myhisteq()");
figure(4); plot(T), title("Transformation Func.");
```

- II - Gamma Transformation

```
% Solution
clear, clc;

I = imread("./Fig0308(a)(fractured_spine).tif");
g_img= mat2gray(I, [0 255]);

% Gamma Transformation: s = C*r^Gamma
C = 1;
Gamma_lighten = [0.05:0.05:1];
Gamma_darken = [1:20];

figure(1);
subplot(121); imshow(I), title("Original Image");
subplot(122); imshow(g_img), title("Gray-scale Image");
saveas(gcf,"Gamma-T 1.png");

for f = 2:6
    figure(f);
    for i = 1:4
        subplot(2, 2, i);
        imshow(C*g_img.^Gamma_lighten(i+(f-2)*4), [0 1]);
        title("Gamma-T ("+Gamma_lighten(i+(f-2)*4)+")");
    end
    path_name = "Gamma-T "+ f +".png";
    saveas(gcf,path_name);
end
```