Spatial Image Enhancement

Ming-Huang Hsieh, 408410071

November 6, 2021

HW1 of Image Processing at National Chung Cheng University

1 Introduction

This is an report of HW1 of the Image Processing course.

2 Technical description

In this section, I will show how pixels are processed to enhance the image.

Tools and Script The tool I used to process the images is MATLAB by The MathWorks, Inc. You can use MATLAB to run the main.m script file to see how the processes work and get the result images.

main.m In the main script, it will read the image and convert to a uint8 matrix which represents gray levels of each pixel. Then pass the image matrix to my transform functions (described below) to get new images.

2.1 Power-Law (Gamma) Transformation

The power-law transformation equation is as shown below.

$$s = c(r + \epsilon)^{\gamma} \tag{1}$$

In my gammaTrans.m function, c is fixed to 1 and ϵ is fixed to 0, so the final transformation equation I use is

$$s = r^{\gamma} \tag{2}$$

where r is the input gray level [0..1], γ is a number greater than zero and s is the output gray level [0..1].

The function will take the 8-bits unsigned integer matrix as the source image, mapping the gray level of each pixel from [0..255] to [0..1], then calculate the power of the value according to γ and then mapping back to [0..255] gray level as output result.

2.2 Histogram Equalization

The function name is histoTrans.m. First, the function will find the percentage of each gray level (i.e. PDF), then accumulate values of PDF to construct CDF, finally each gray level times its corresponding CDF value to get the output gray level. Below is concrete process:

$$T(r_k) = \sum_{i=0}^{k} \frac{n_i}{n}, \quad r_k = \text{gray level}, n_i = \text{number of pixels whose gray level is } i \tag{3}$$

New image: Mapping r_k to $r_k \times T(r_k)$ for each pixel, then round off to get rid off floating point.

2.3 Laplacian Operator

The function name is laplacianTrans.m. For each new pixel, it is calculated from original gray level of itself and up/down/left/right pixels as equation below.

$$p_{x,y}^{new} = 5 \times p_{x,y}^{old} - p_{x-1,y}^{old} - p_{x+1,y}^{old} - p_{x,y-1}^{old} - p_{x,y+1}^{old} \tag{4}$$

3 Experimental results

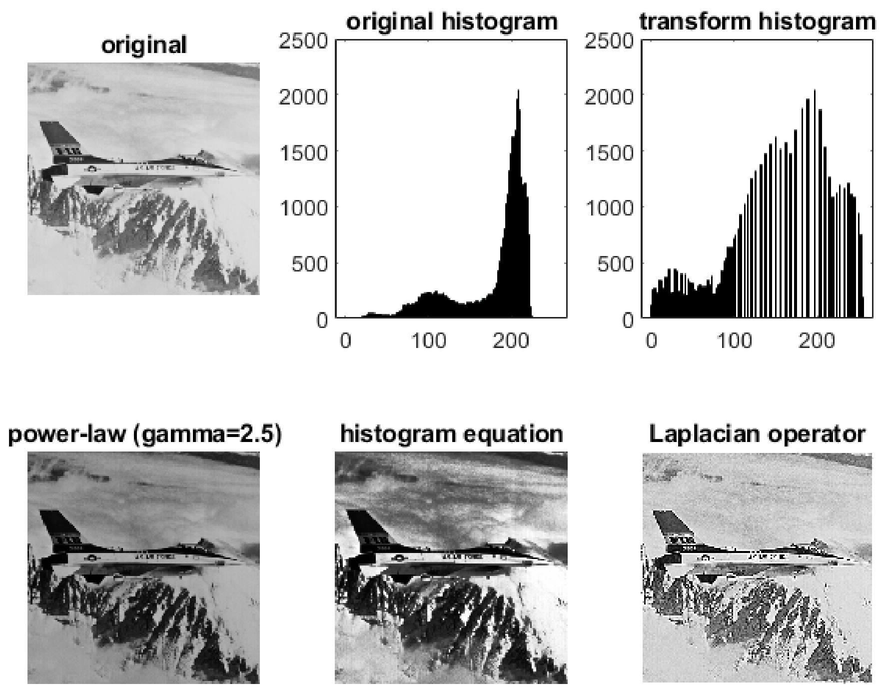

Figure 1: Jetplane.bmp result

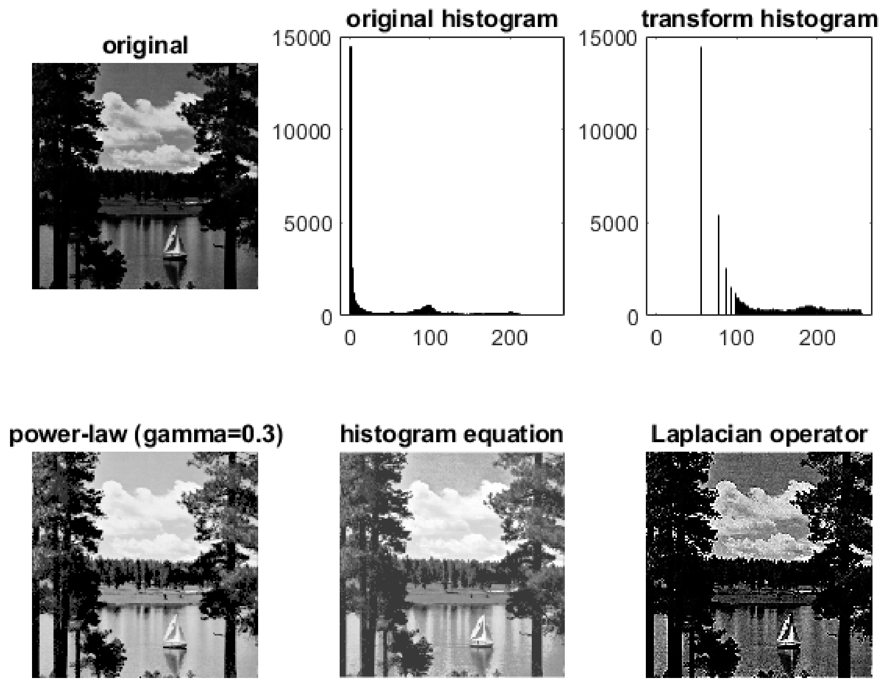

Figure 2: Lake.bmp result

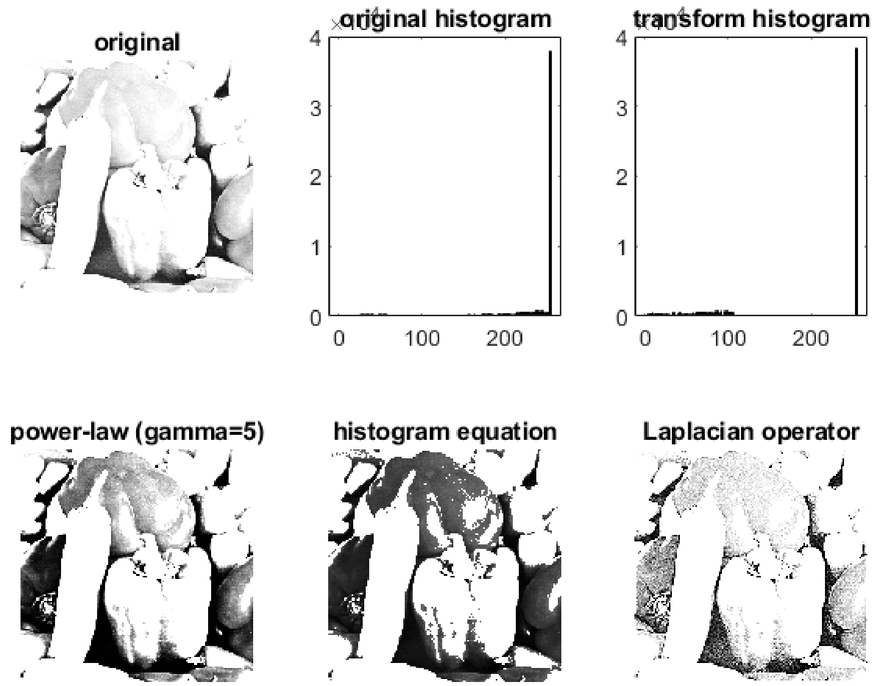

Figure 3: Peppers.bmp result

4 Discussions

There are something interesting and discovery when I was doing this homework.

1. If γ greater than 1, the output image will be darker, it's useful when a image is over-exposure; On the other hand, if γ between 0 and 1, the output image will be brighter.

2. When we apply histogram equation to an image, the bright and dark part will be more "balance", but if the image is too bright or too dark, this approach will have just a little bit effect. We can confirm that by comparing original image's histogram and output image's histogram.

5 References and Appendix

The MATLAB scripts and functions are included in the archive file.