Intro. to Image Processing HW2

110550093 蔡師睿

Method

Histogram Equalization

First, I generate the PDF of Q1.jpg by counting the values of all pixels. After getting PDF, we obtain the CDF, and then multiply all its values by 255 and round off to the nearest integer. Finally, map the pixel values of the input image to new values using the CDF.

Histogram Specification

First, compute the PDF of Q1.jpg and Q2.jpg by counting the values of all pixels, and we can obtain their CDF. Multiply CDF by 255 and round off to the nearest integer. Then, for each pixel values in Q1, map it to the CDF of Q1 and use the value in CDF of Q1 to find the minimal index of closest value in CDF of Q2. Each index we find is the final value of each new imagae pixels.

Gaussian Filter

To begin with, generate the 5x5 gaussian kernel with $K = 1, \sigma = 25$. We get :

$$\begin{bmatrix} 0.03987213 & 0.03996794 & 0.03999993 & 0.03996794 & 0.03987213 \\ 0.03996794 & 0.04006398 & 0.04009604 & 0.04006398 & 0.03996794 \\ 0.03999993 & 0.04009604 & 0.04012813 & 0.04009604 & 0.03999993 \\ 0.03996794 & 0.04006398 & 0.04009604 & 0.04006398 & 0.03996794 \\ 0.03987213 & 0.03996794 & 0.03999993 & 0.03996794 & 0.03987213 \end{bmatrix}$$

Second, convolve the kernel with the Q3.jpg. However, after convolving, the size of image will shrink. To avoid this, we pad 0 outside the image to make sure that the size remains unchanged after convolving. Then we can do convolution and round off all the values to the nearest integer and the result is what we expect.

Result

Result of Histogram Equalization

Result of Histogram Specification

Result of Gaussian Fiilter

Feedback

In the beginning, I discovered that my program runs so slowly. After I optimized my convolution, my gaussian filter runs faster and I'd like to point out that numpy is really powerful !