Image Processing Mini Project

0123456 林允柔 資工系四年級

Programing language: C++

OpenCV version (if needed): 3.2

Develop environment: Visual Studio 2013

A. Program flow

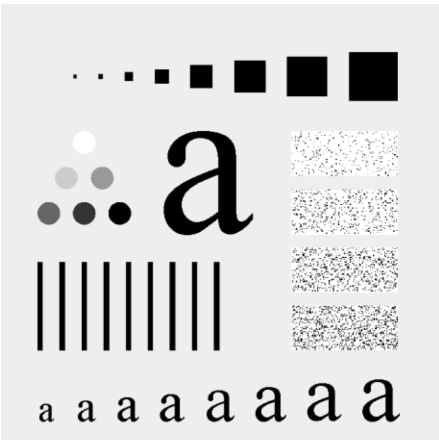
I. Method 1

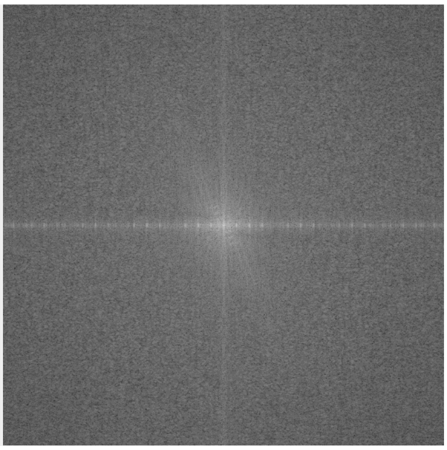
1. Construct h(x, y)

   ($x_0(t)=y_0(t)=t/10$, T=1)

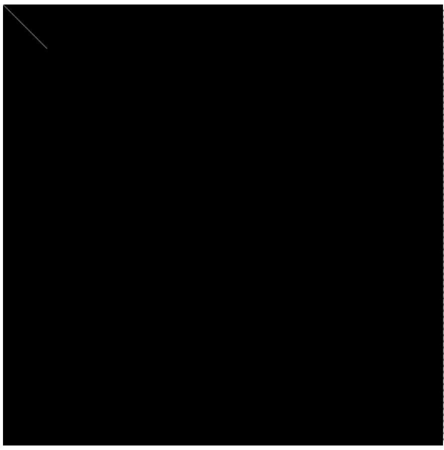
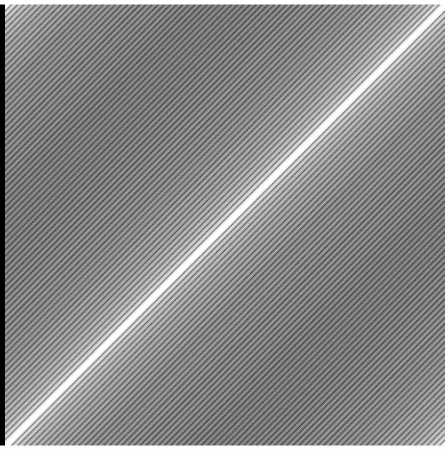
2. Obtain F(u, v) and H(u, v) from f(x, y) and h(x, y)

   Use openCV function **`getOptimalDFTSize()`** to enlarge the image for optimal dft, and also pad the original image by **`copyMakeBorder()`** for borderType of `BORDER_REFLECT_101`.Then use openCV function **`dft()`** to implement Fourier Transform.



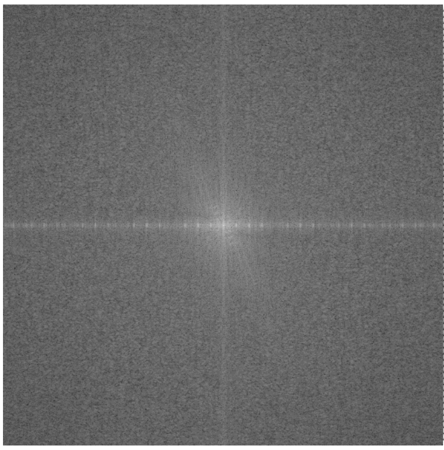
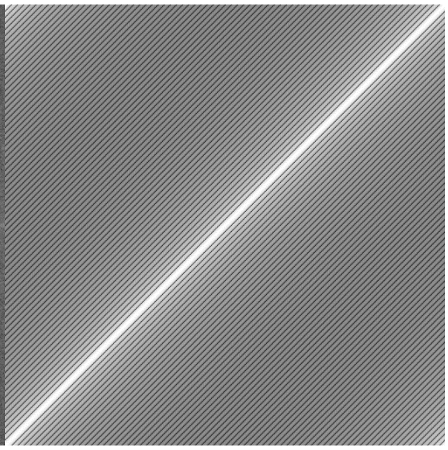


f(x, y) → F(u, v)

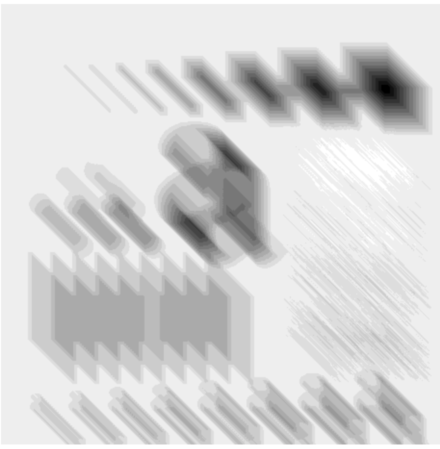
h(x, y) → H(u, v)

3. Filtering in frequency domain by elementwise multiplication of the 2 spectrums F(u, v) and H(u, v).

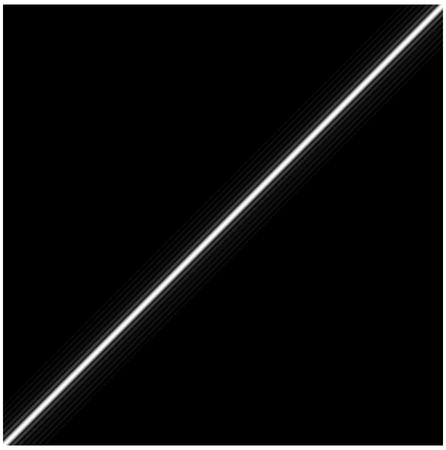
F(u, v) • H(u, v)

4. Then do inverse DFT to obtain the uniform motion blur image, and also crop the image from 720x720 to 688x688.

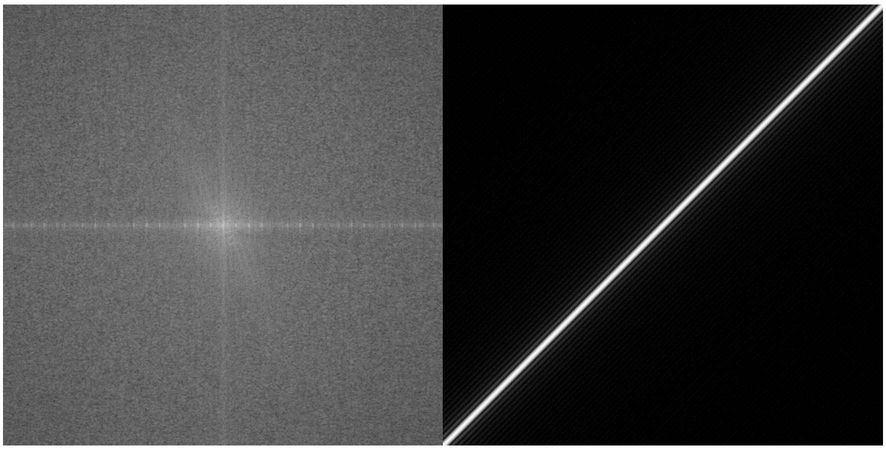


II. Method 2

1. Construct H(u, v) in frequency domain by the function in slide32.

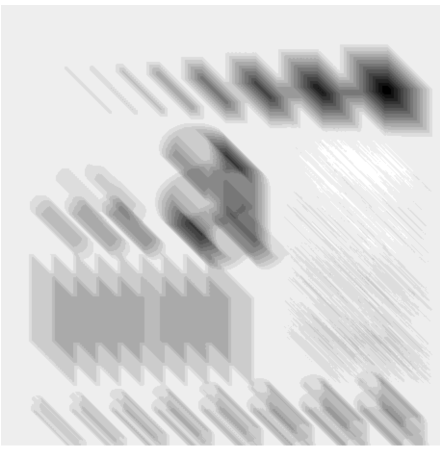


2. Filtering in frequency domain by elementwise multiplication of the 2 spectrums F(u, v) and H(u, v).



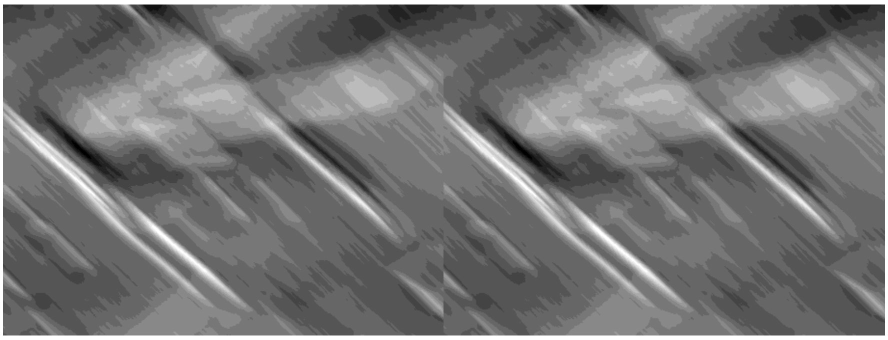
F(u, v) • H(u, v)

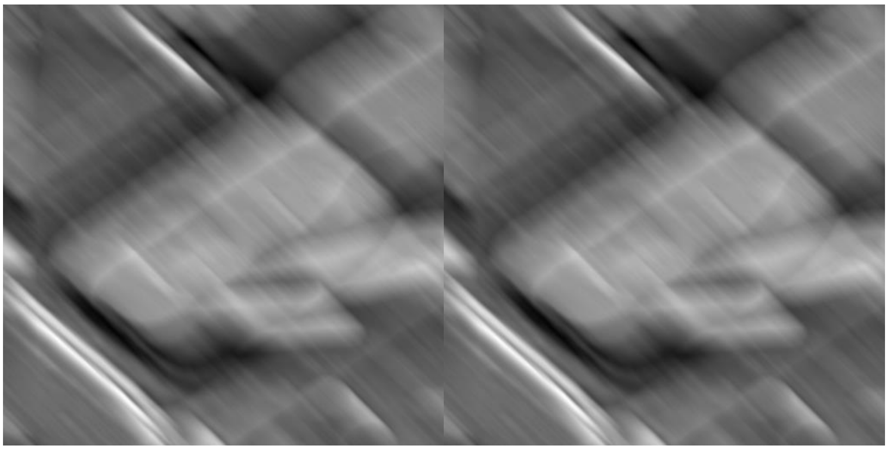
3. Then do inverse DFT to obtain the uniform motion blur image, and also crop the image from 720x720 to 688x688.

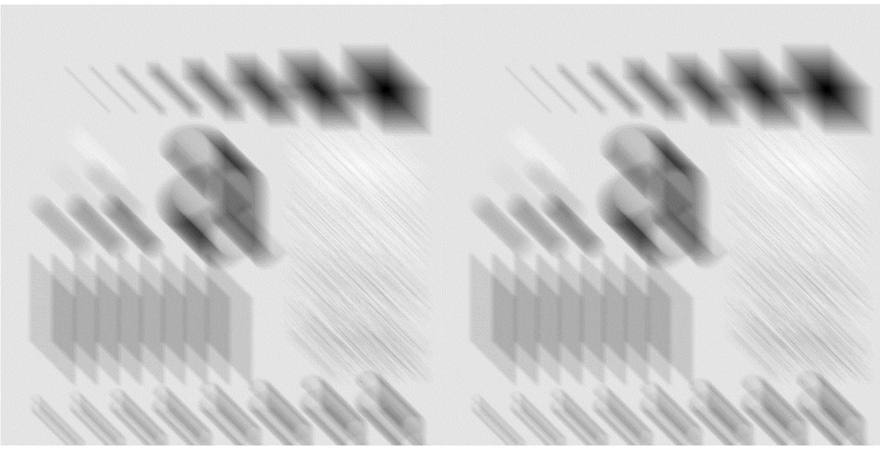


B. Result and Analysis

a. Results for method 1 and 2.







b. Analysis

It seems to be little different between results for method 1 and 2. The obvious differences we can see are the magnitude image of H(u, v) and h(x, y). For method 2, H(u, v) looks darker than that for method 2. Also, there is a light cross on the right-lower white pixel in h(x, y), and the h(0, 0) is not 255 (white), since the average intensity of H(u, v) is not 1.