Problem 3.4 Report

I. Programing

i. Environment

Programing language: C++ with OpenCV 2.4.13.

Execution environment: Ubuntu 16.04

Compile: `cmake && make`

ii. Usage: `$ ./main <image_path>`

This program performs filtering with Gaussian highpass (lowpass) filter in frequency domain, and outputs the spectrum of filtered image, the filter, and the output image. Options or parameters are asked to input during execution.

opt={0, 1} to choose which filter to use

D0=constant of Gaussian filter

iii. Implementation

In this program, the following functions are implemented:

```
Mat doDFT(Mat img);
Mat doIDFT(Mat imgComplex);
Mat makeSpecImg(Mat imgComplex);
void shift(Mat &specImg);

void createGHPF(Mat &mask, float d0);
void createGLPF(Mat &mask, float d0);
```

Mat doDFT(Mat img), Mat doIDFT(Mat imgComplex): performs discrete Fourier transform and inverse transform, respectively.

void shift(Mat &specImg): shift the spectrum to the center.

Mat makeSpecImg(Mat imgComplex): create spectrum image.

void createGHPF(Mat &mask, float d0), void createGLPF(Mat &mask, float d0): create GHPF or GLPF masks.

And the following functions are called:

copyMakeBorder(), getOptimalDFTSize(): expand border to the form of 2^k with zeros to make DFT faster.

dft(): performs discrete Fourier transform.

idft(): performs inverse discrete Fourier transform.

magnitude(): compute the spectrum magnitude.

normalize(): normalize the spectrum image to the range of visible image (and then scale to the discrete range of 0 to 255)

mulSpectrums(): multiply image with masks in frequency domain.

II. Results and Discussion

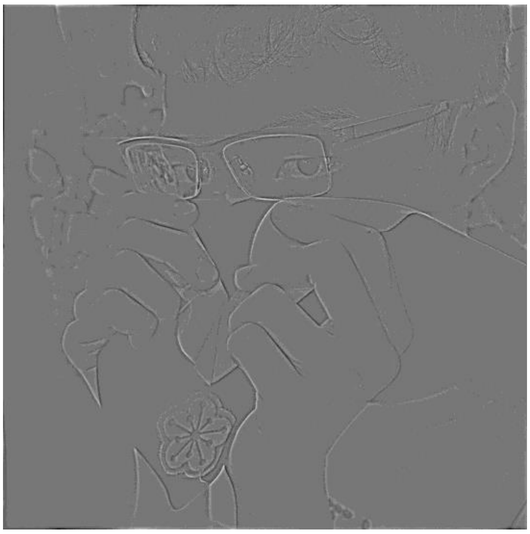
1. Gaussian Highpass Filter

$H_{\text{hp}}(u, v) = 1 - H_{\text{lp}}(u, v)$, where $H_{lp}(u, v) = e^{-D^2(u,v)/2\sigma^2}$.

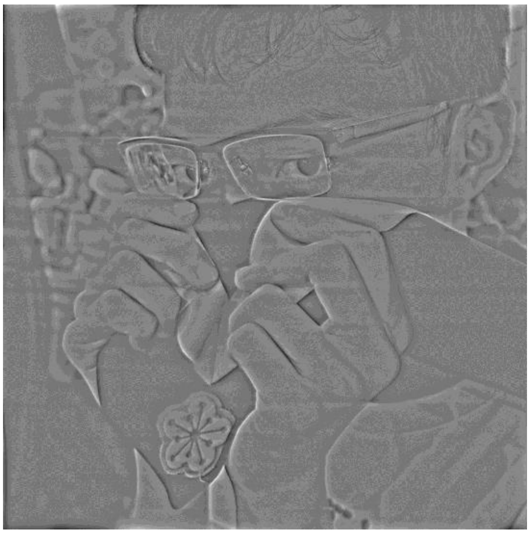
With different D0 in H(u, v), we have following results:

D0=10

D0=50



D0=20

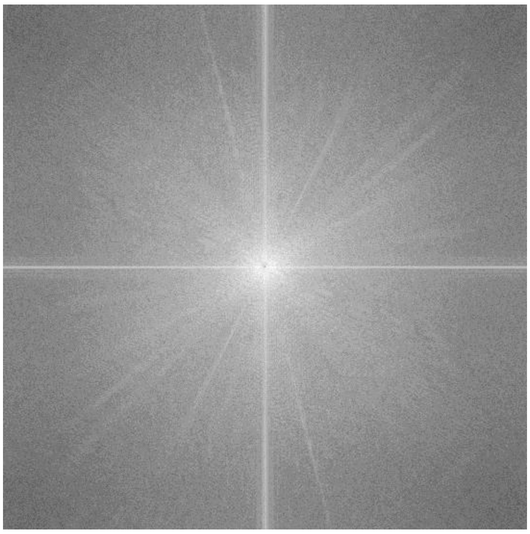


D0=100

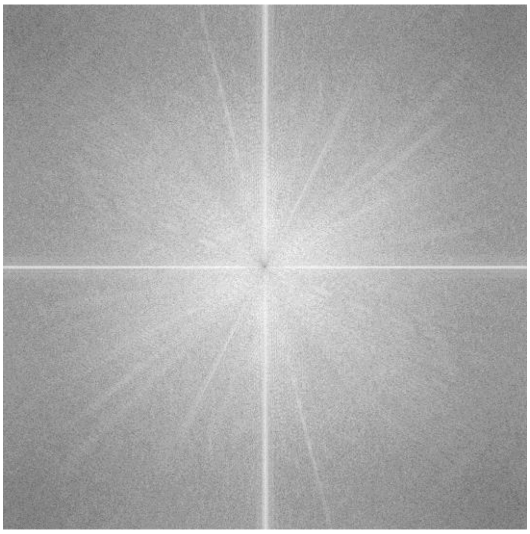
We may observe that the more D0 grows, the sharper remaining edges are. The output images can be further used in image sharpening.

,whose spectra are:

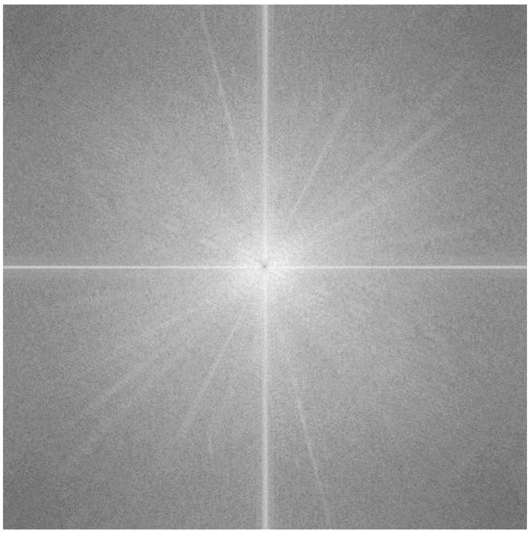
D0=10



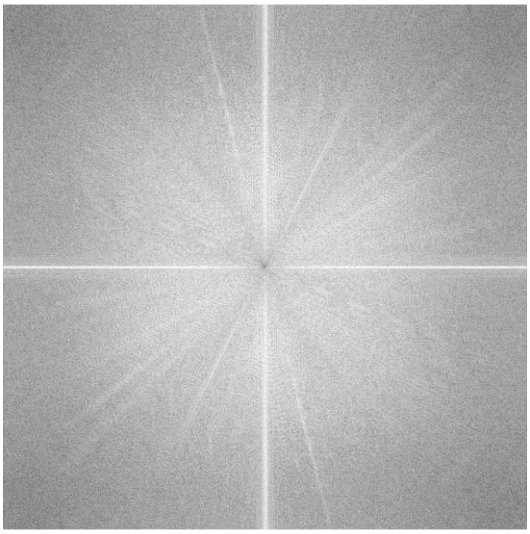
D0=50



D0=20



D0=100



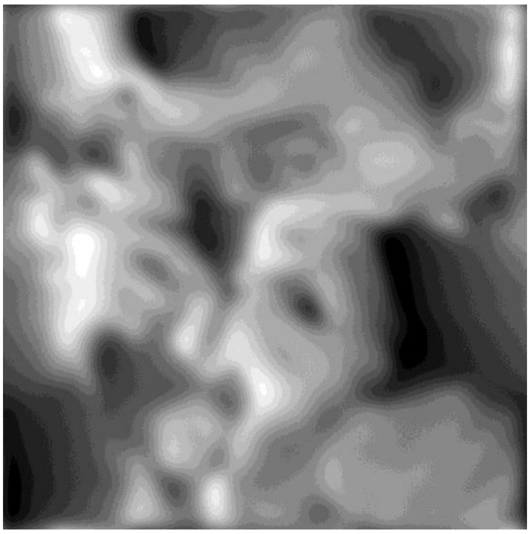
Since the spectra are normalized to the highest magnitude, we may observe the contrast between the low frequency part and the high one. The higher the contrast is, the stronger the magnitude of the low frequency part is. That is, with D0 higher, we have filtered out more of the low frequency part.

2. Gaussian Lowpass Filter

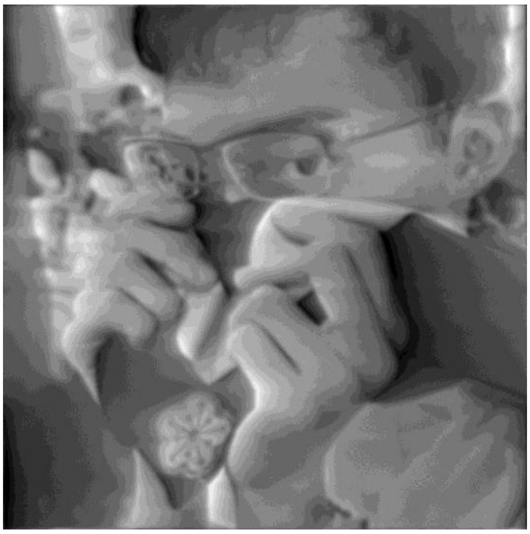
$$H(u, v) = e^{-D^2(u, v)/2\sigma^2}.$$

With different D0 in H(u, v), we have following results:

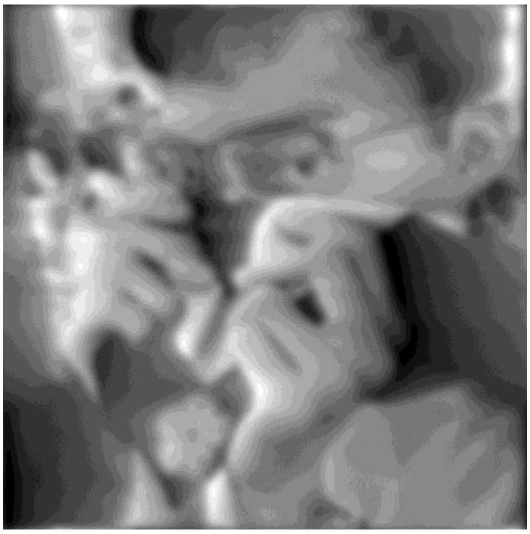
D0=10



D0=50



D0=20

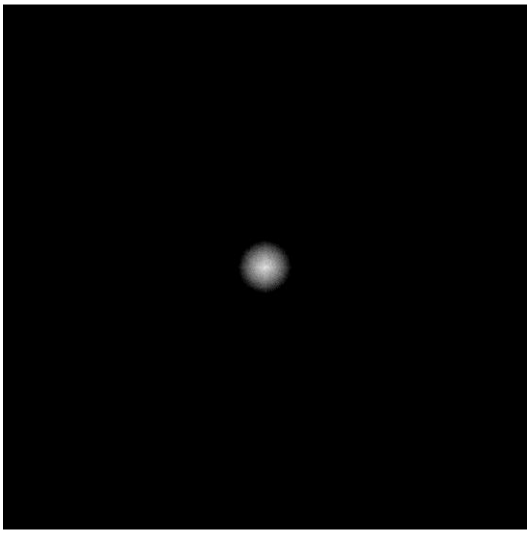


D0=100

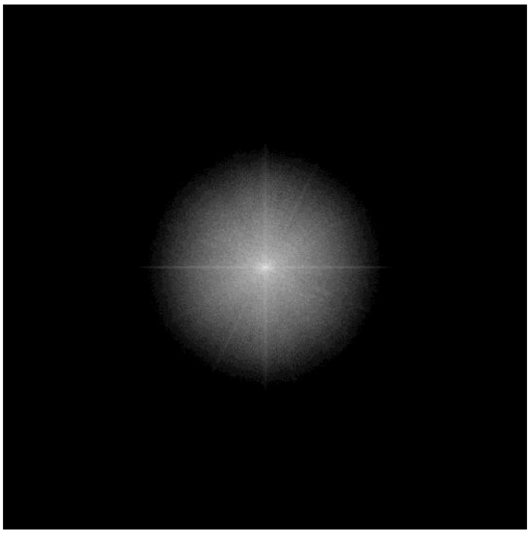
We may observe that the less D0 is, the output image becomes smoother. Since the edges can be regarded as high frequency responses in the image, we can blur the image by filtering out the high frequency parts.

,whose spectra are:

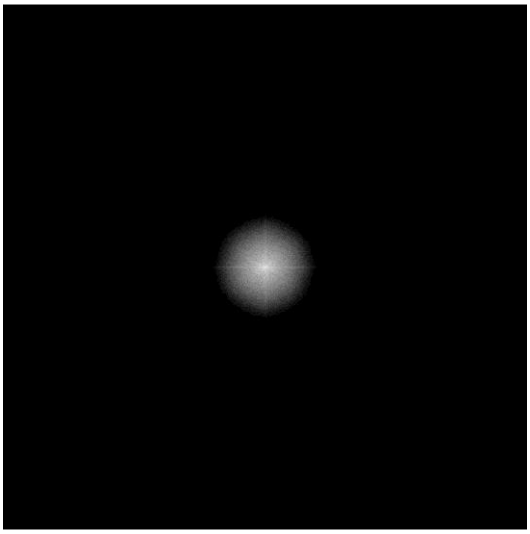
D0=10



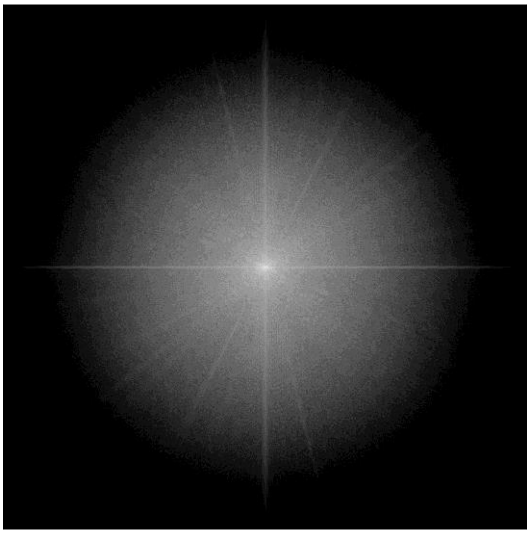
D0=50



D0=20



D0=100



The high frequency parts are filtered out in a gradual fashion.

Reference:
http://www.mamicode.com/info-detail-589826.html
http://vgg.fiit.stuba.sk/2012-05/frequency-domain-filtration-source/