Image Processing Lab 3

許木羽 / 111000177

1. Project 04-01

From the general formula for 2D DFT as:

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y) * e^{-2\pi i(\frac{ux}{N}+\frac{vy}{M})}$$

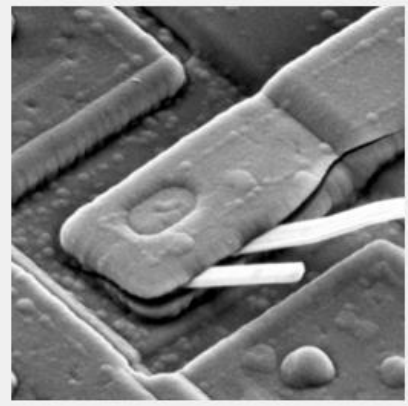
We can easily make this using for loops:

```
for u = 0:N-1
    for v = 0:M-1
        msum = 0;
        for x = 0:N-1
            for y = 0:M-1
                msum = msum + input(x+1, y+1) * exp(-2 * pi * 1i * (x*u/N + y*v/M));
            end
        end
        output(u+1, v+1) = msum;
    end
end
```

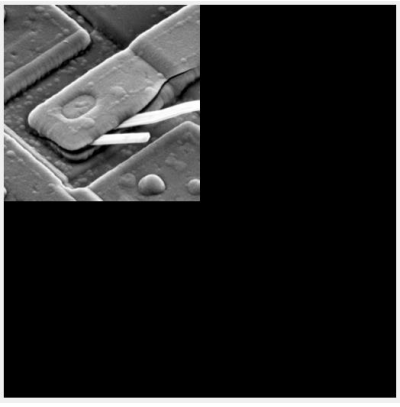
However, it takes a long time, so we need to improvise.

First, we can calculate the exponential separately, as $e^{-2\pi i(\frac{ux}{N})}$ will be stored in 2D matrix U * X called power1, the same thing as $e^{-2\pi i(\frac{vy}{M})}$ as 2D as V * Y matrix, called power2, power(v, y) indicates exponent function power of (u, x) or (v, y). We will operate power1 * input to apply DFT for the horizontal direction, and the result is multiplied by power2 to apply DFT for vertical direction.

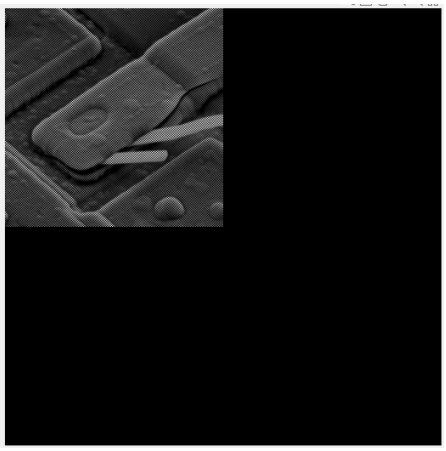
a) Original Image

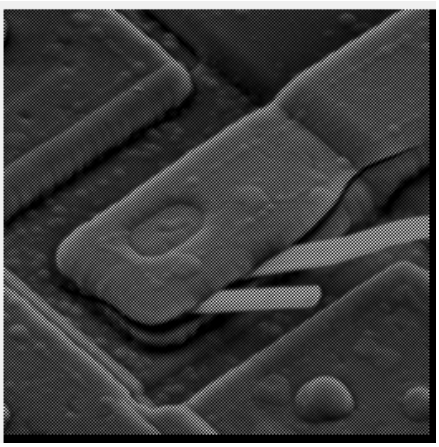


b) Shrink to ¼

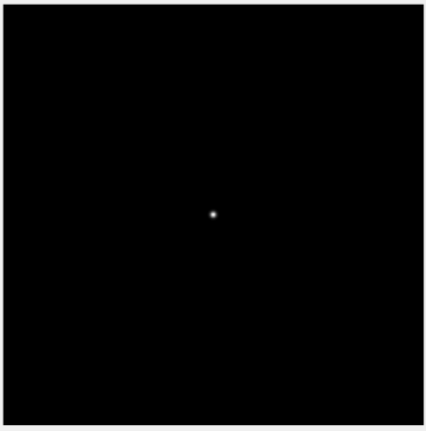


c) Multiply by noise F(x, y) = f(x, y) * (-1) ^ (x + y) => “Right picture is zoomed”

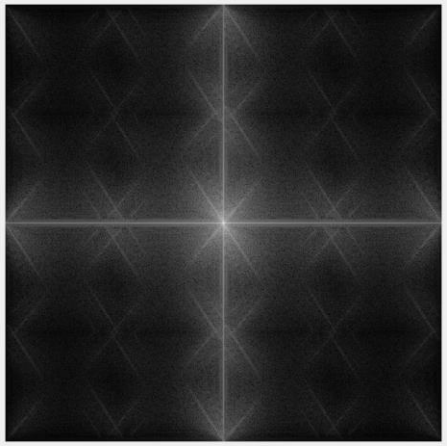




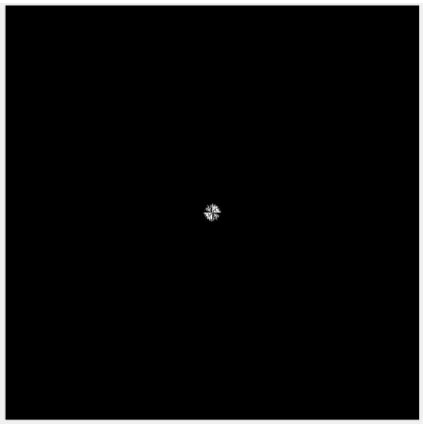
d) Spectrum



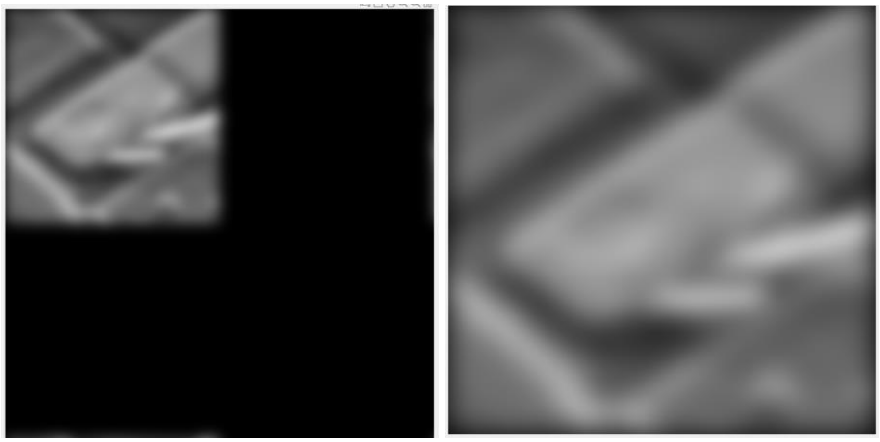
e) Gaussian Low Pass Filter Spectrum



f) Spectrum product with GLPF

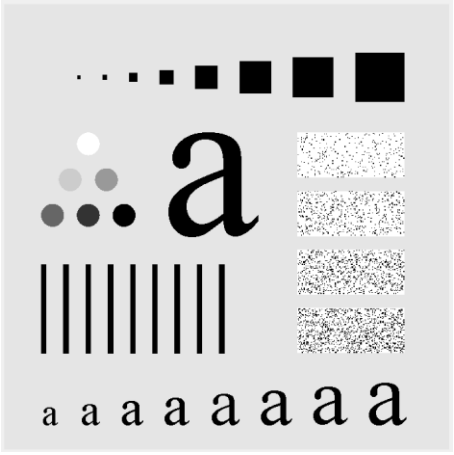


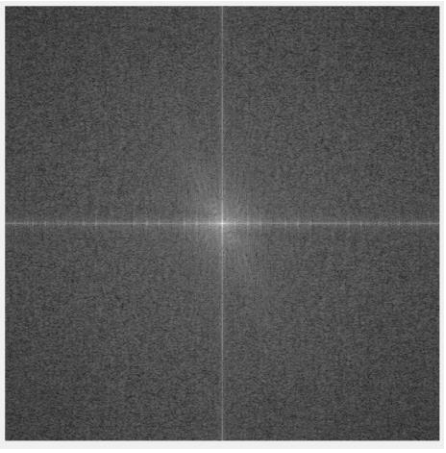
g) Image filtered by IFFT and using noise F(x, y) = f(x, y) * (-1) ^ (x + y)



Compared without original padding (3/4 black color), we will get image below. Notice the border doesn't have any shadow.

2. Project 04-02





For the mean, I compute the original image and got 0.8130, while the center of the spectrum, with radius of around 30 pixels is 0.82, and the spectrum average is 0.82 also.

```
M1 =

  single

    0.8130

M2 =

  single

    0.8130
```

3. Project 04-03

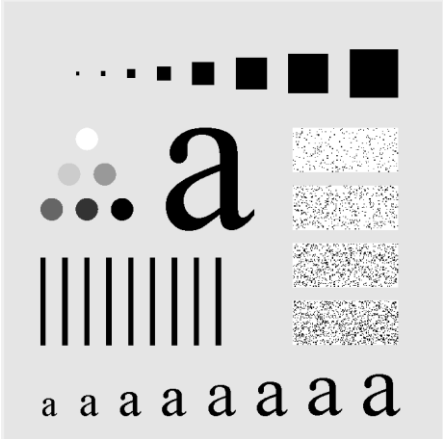
I implemented as the formula is:

$$F(x,y) = e^{-\frac{D(x,y)}{2* {D_0}^2}}$$

$$D(x,y) = (x - c_x)^2 + \left(y - c_y\right)^2$$

as c_x indicates center x and c_y indicates center y

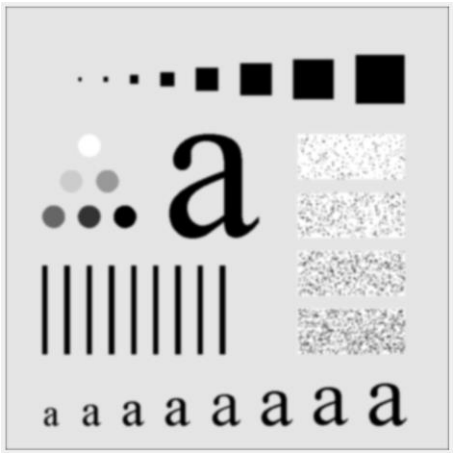
Below is implementation of original image, to D0 = 10, 25, 75, 125, 200, and 400. (Left to right, up to down)



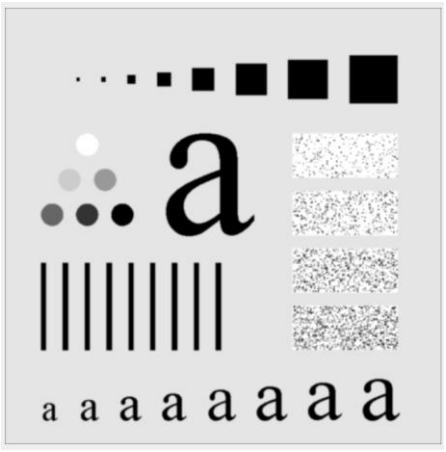
Original Image

D0 = 25

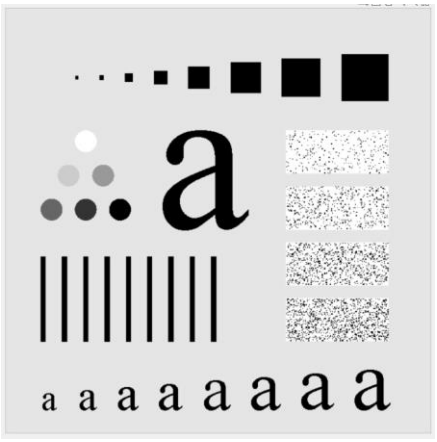
D0 = 75



D0 = 125



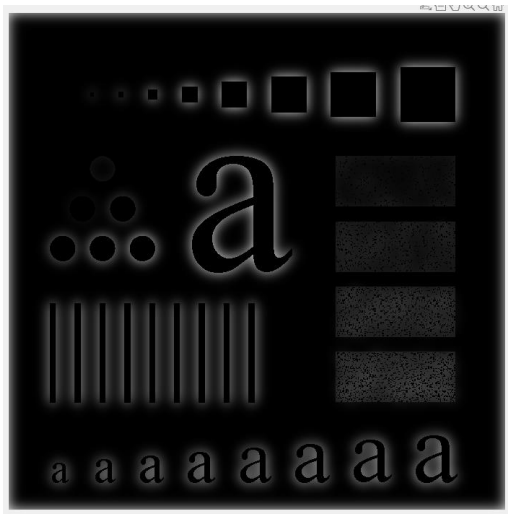
D0 = 200

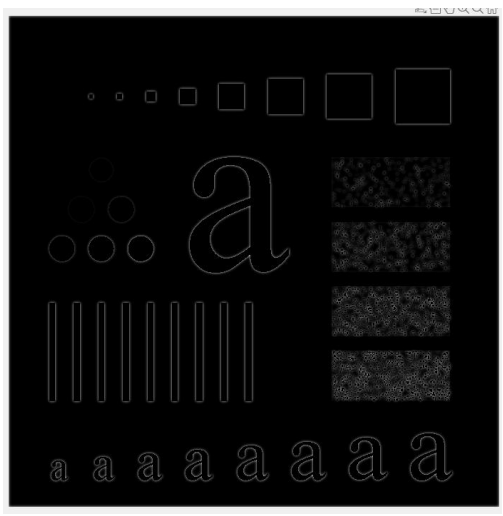


D0 = 400

4. **Project 04-04**

   I implemented the same way as Gaussian Low Pass Filtering, but instead the output need an extra step, as $F(x,y) = 1 - f(x,y)$ as $f(x,y)$ is GLPF.





D0 = 25 and 120