Introduction to Image Processing – Lab03 Report

107062230 張力仁

Proj04-01: Two-Dimensional Fast Fourier Transform

(1) Implementation

myDFT2:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} = \sum_{x=0}^{M-1}\Big(\sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(\frac{vy}{N})}\Big)\, e^{-j2\pi(\frac{ux}{M})}$$

By viewing 2D DFT as two separable 1D DFTs, we can convert it into two matrix multiplications. We can first apply 1D DFT to each column and then to each row, which can be expressed as $DFT_{col} * X * DFT_{row}^T$, where

$$DFT_{col} = exp(-i * 2 * \pi * [0{:}M-1]^T * [0{:}M-1]\,/\,M) \text{ and}$$
$$DFT_{row} = exp(-i * 2 * \pi * [0{:}N-1]^T * [0{:}N-1]\,/\,N).$$

myIDFT2:

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

$$= \frac{1}{M}\sum_{u=0}^{M-1}\Big(\frac{1}{N}\sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(\frac{vy}{N})}\Big)\, e^{j2\pi(\frac{ux}{M})}$$

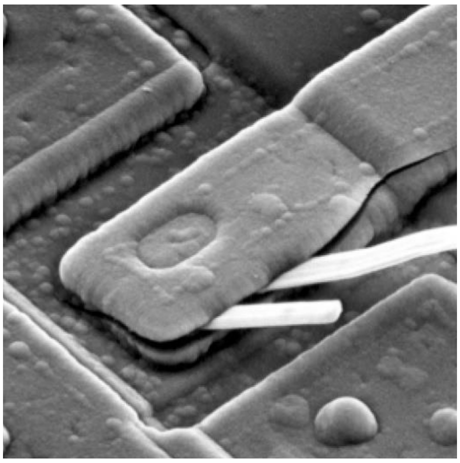
Similarly, we can write 2D IDFT as matrix multiplications of $(IDFT_{col} * X * IDFT_{row}^T)\,/\,(MN)$, where

$$IDFT_{col} = exp(i * 2 * \pi * [0{:}M-1]^T * [0{:}M-1]\,/\,M) \text{ and}$$
$$IDFT_{row} = exp(i * 2 * \pi * [0{:}N-1]^T * [0{:}N-1]\,/\,N).$$

(2) Results

Fig 4.35 (a)

Fig 4.35 (b)



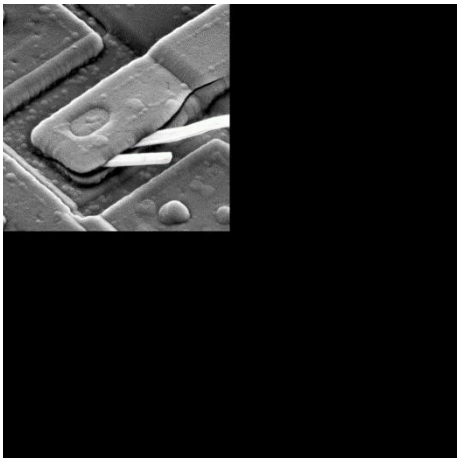


Fig 4.35 (c)

Fig 4.35 (d)

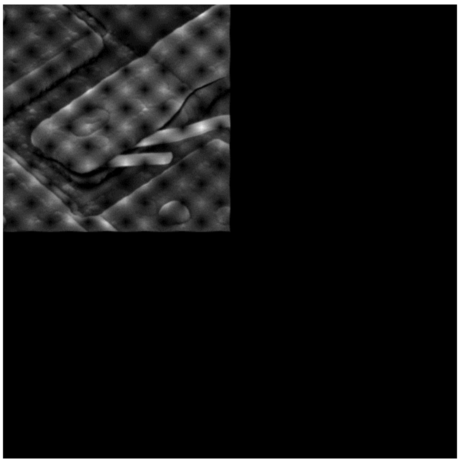


Fig 4.35 (e)

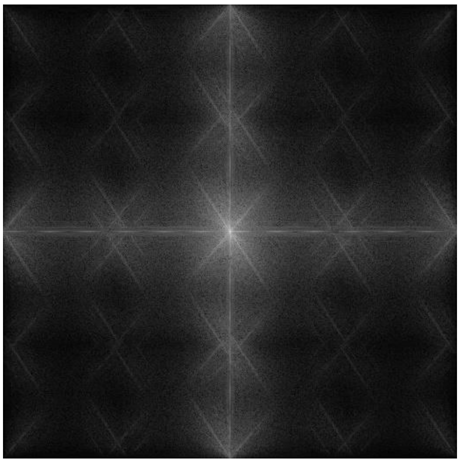


Fig 4.35 (f)

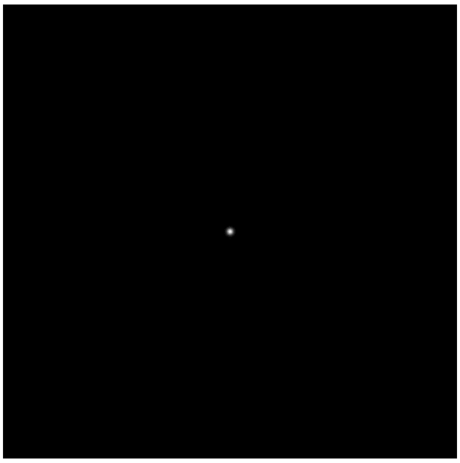


Fig 4.35 (g)

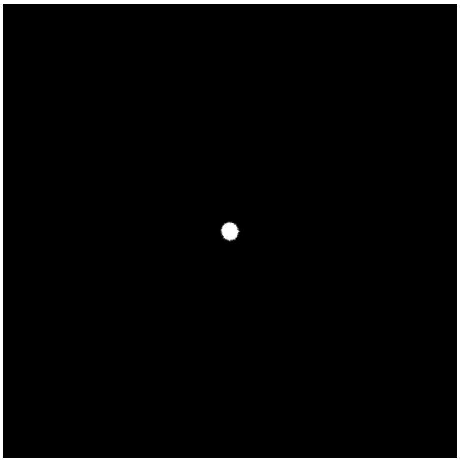


Fig 4.35 (h)

(3) Method to Accelerate DFT

Refactoring 2D DFT into matrix multiplications can make calculation of DFT significantly faster than merely using nested for-loops in MATLAB.

(4) Padding vs No Padding before Converting into the Frequency Domain

Without padding, filtering in frequency domain will be equivalent to circular convolution in spatial domain. With padding, filtering in frequency domain will become linear convolution in spatial domain.

With Padding

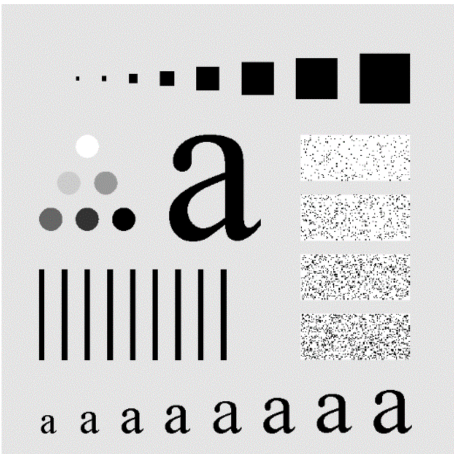
Without Padding

The main difference of visual effect is the black borders.

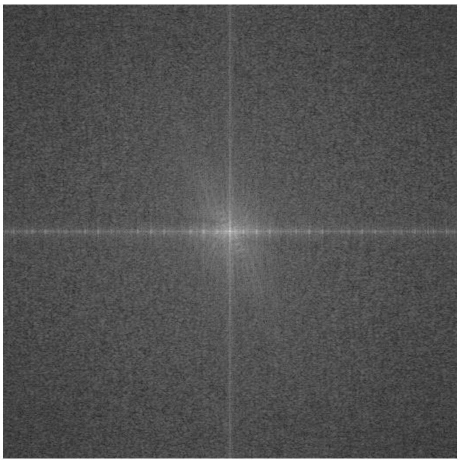
Proj04-02: Fourier Spectrum and Average Value

(1) Results

Fig 4.41 (a)



Fourier Spectrum



Note that the Fourier spectrum is displayed with log transformation and scaling to $[0, 255]$.

(2) Calculate Mean from Image vs from Center of Fourier Spectrum

```
>> main
Mean from image: 0.813469
Mean from DC: 0.812999
12.860555
fx >>
```

The difference between mean values computed by these two methods is quite small.

Proj04-03: Lowpass Filtering

(1) Implementation of Gaussian Lowpass Filter

for i = 1 to M

  for j = 1 to N

    $u = i - 1;\ v = j - 1;$

    $D = \left(\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right)^{\frac{1}{2}};$

    $output(i,\ j) = exp(-D^2\ /\ (2\ *\ D_0^2));$

  end

end

(2) Results

Fig 4.44 (a)

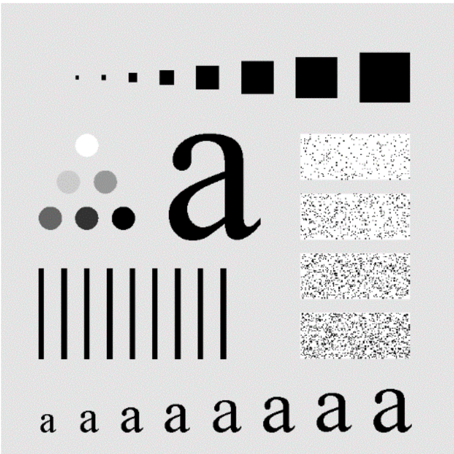


Fig 4.44 (b)

Fig 4.44 (c)

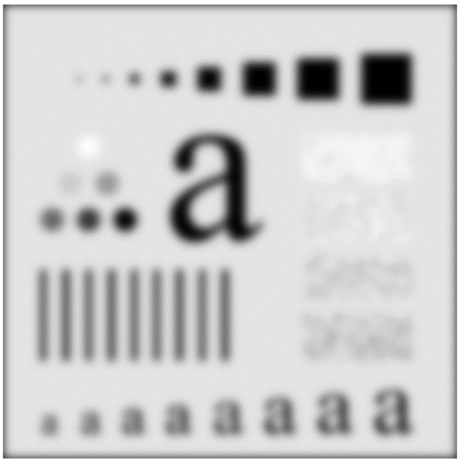
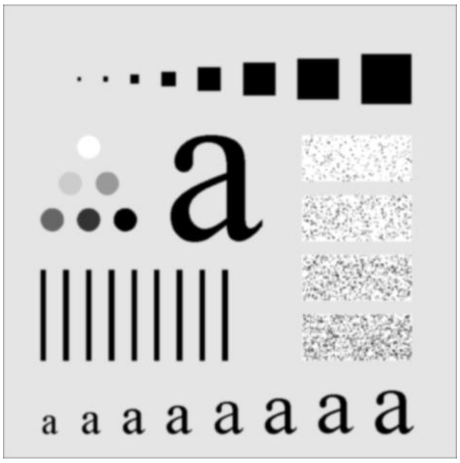
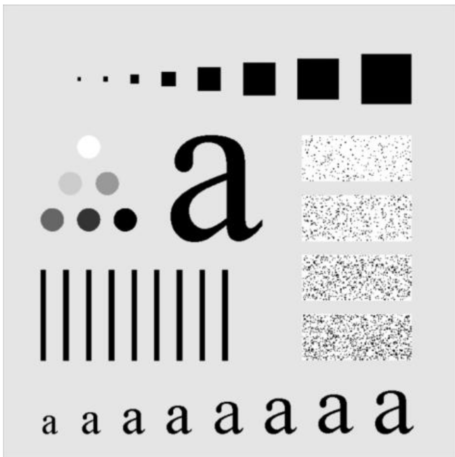


Fig 4.44 (d)

Fig 4.44 (e)

Fig 4.44 (f)





The black borders presented in the images are caused by the characteristic of zero padding.

Proj04-04: Highpass Filtering

(1) Implementation of Gaussian Highpass Filter

for i = 1 to M

for j = 1 to N

$u = i - 1;\ v = j - 1;$

$D = \left(\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right)^{\frac{1}{2}};$

$output(i,\ j) = 1 - exp(-D^2\ /\ (2\ *\ D_0^2));$

end

end

(2) Results

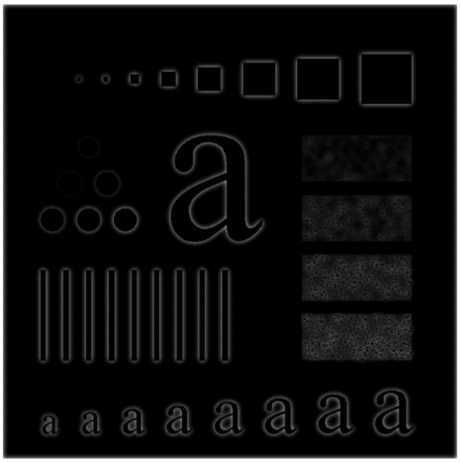


Fig 4.53 (b)

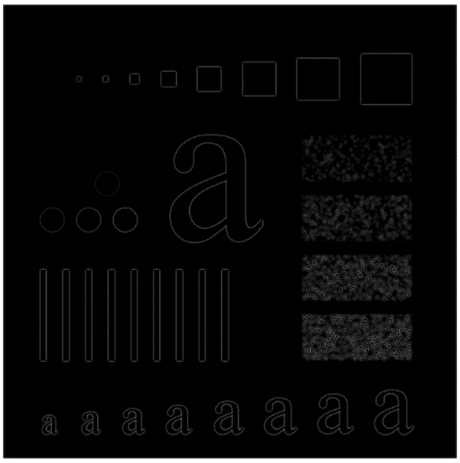


Fig 4.53 (e)