Lecture 4
Filtering in the Frequency Domain

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2014

Outline

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction

Background

- Fourier analysis (Fourier series and Fourier transforms) is quite useful in many engineering fields
- Linear image filtering can be performed in the frequency domain
- A working knowledge of the Fourier analysis can help us have a thorough understanding of the image filtering

Background

- Jean Baptiste Joseph Fourier was born in 1768, in France
  - Most famous for his work “La Théorie Analitique de la Chaleur” published in 1822
  - Translated into English in 1878: “The Analytic Theory of Heat”

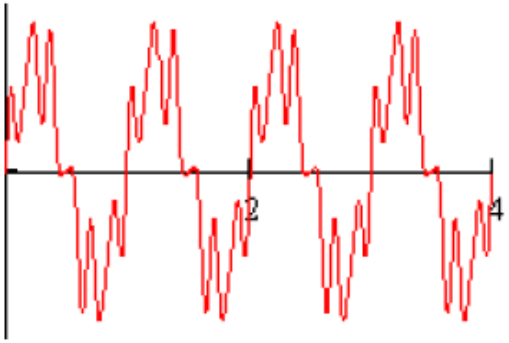
21 March 1768 – 16 May 1830

Outline

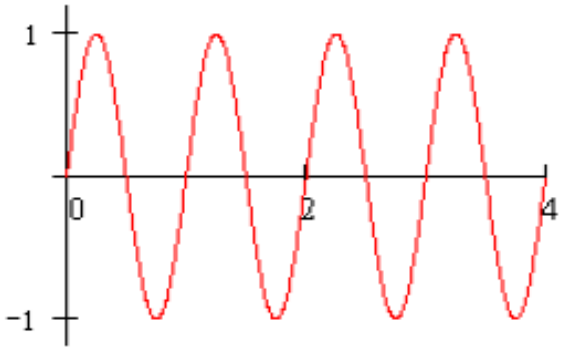
- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters

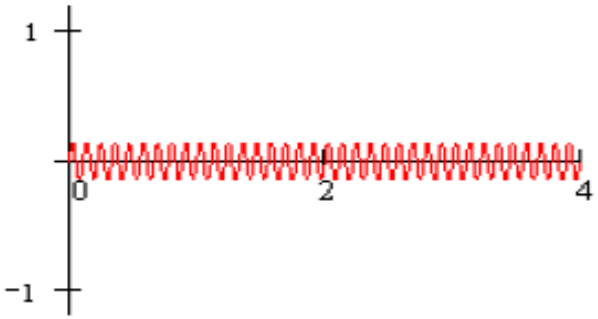
Fourier Series

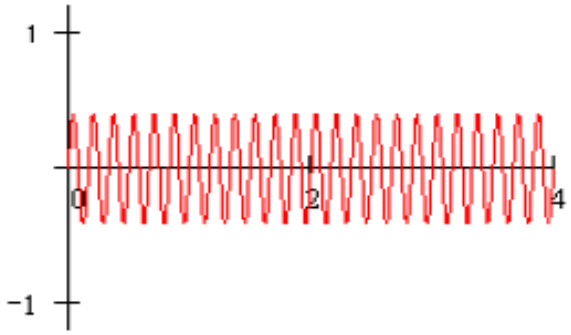
- For any periodic function $f(t)$, how to extract the component of f at a specific frequency?



is composed of the following components

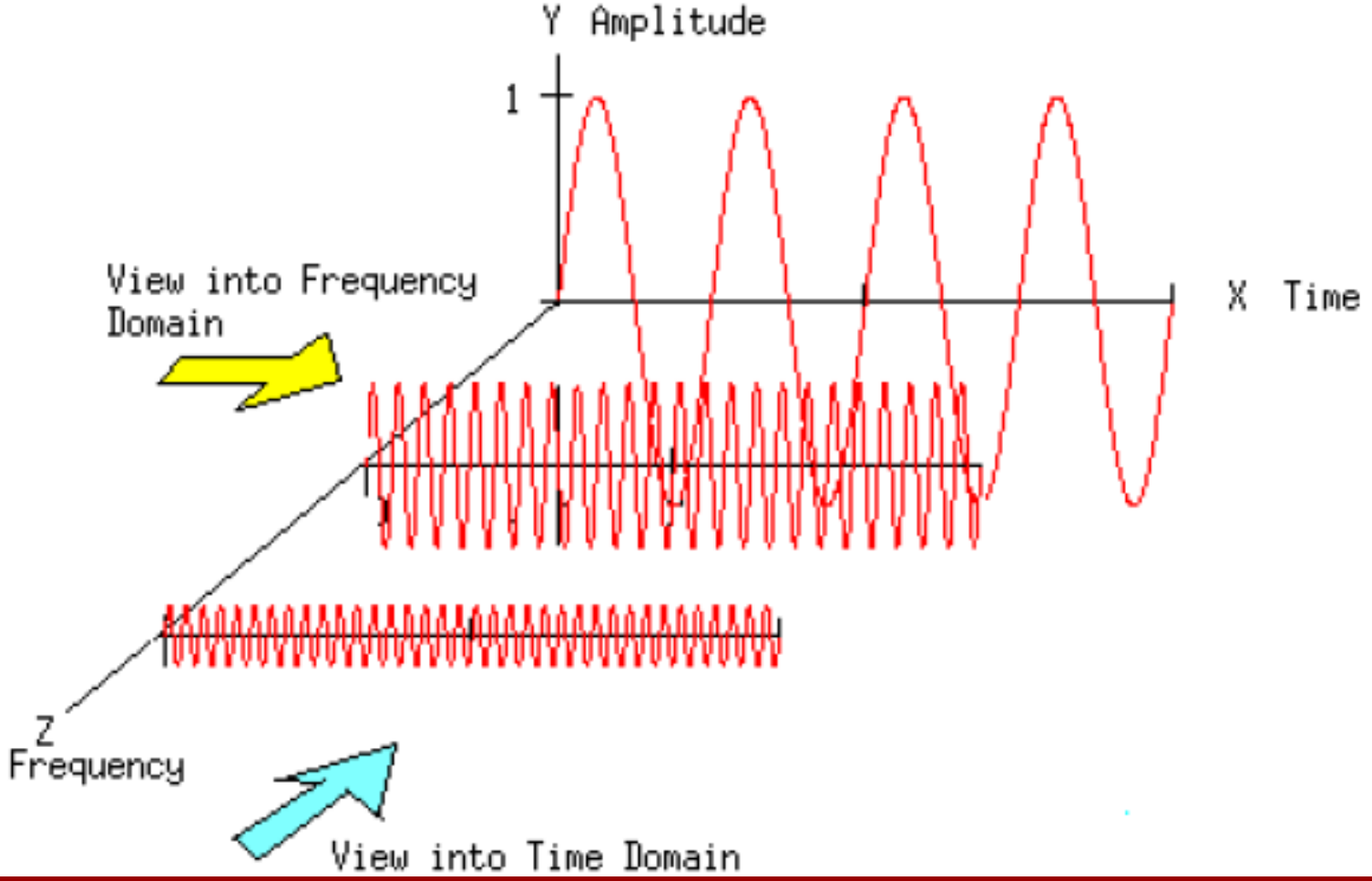






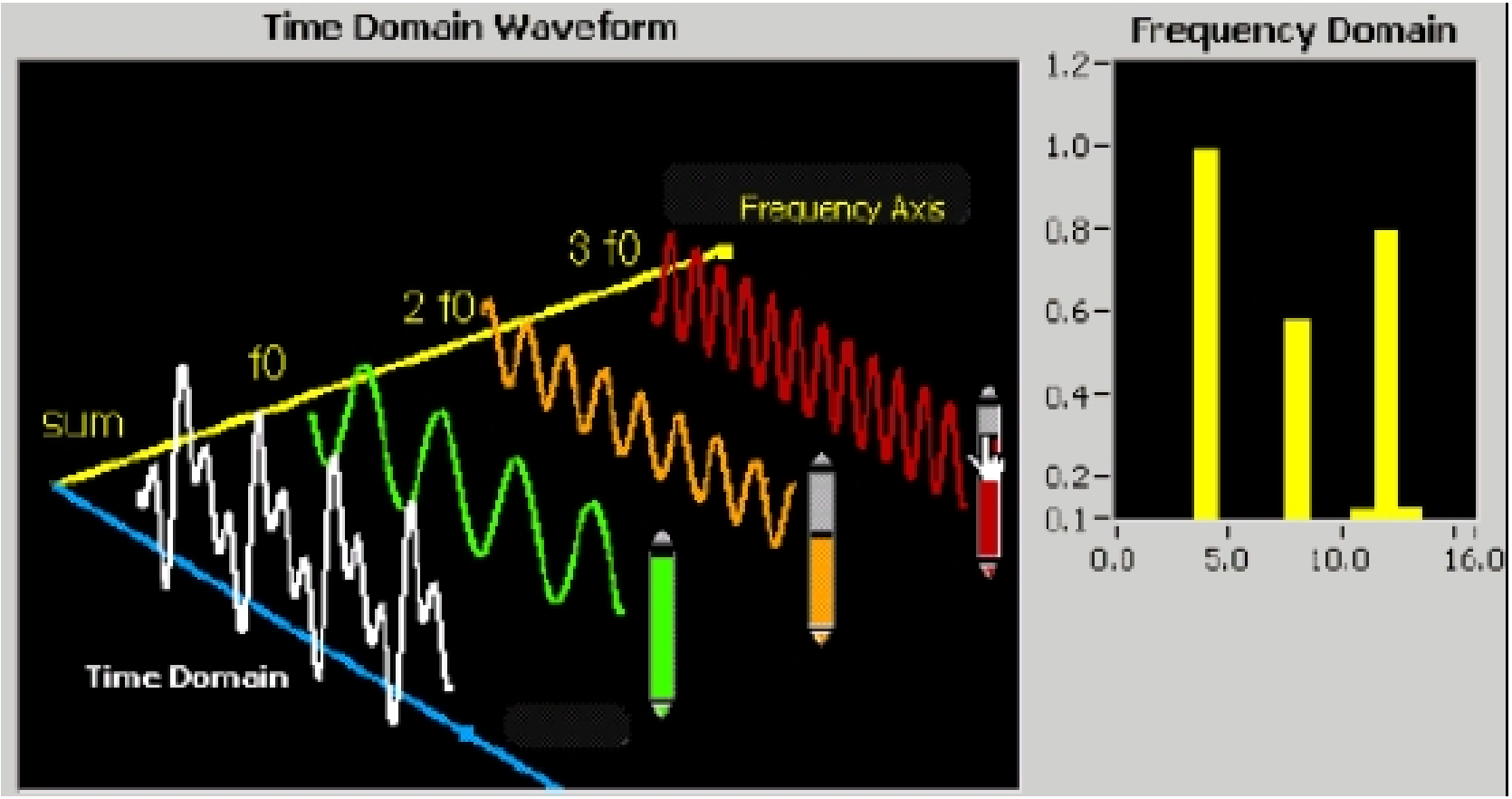
Fourier Series

- For any periodic function $f(t)$, how to extract the component of f at a specific frequency?



Fourier Series

- For any periodic function $f(t)$, how to extract the component of f at a specific frequency?



Fourier Series

- For any periodic function $f(t)$, how to extract the component of f at a specific frequency?

Fourier Series

Any periodic function can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos n\omega t + b_n \sin n\omega t \right)$$

more details

Fourier Series

For a periodic function $f(t)$, with period T

Fourier Series

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos n\omega t + b_n \sin n\omega t \right)$$

where

$$\omega = \frac{2\pi}{T}$$

Redundant!

$$a_0 = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)dt \,,$$

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cos n\omega t dt$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \sin n\omega t dt$$

Fourier Transforms

Fourier transform of $f(t)$ (maybe is not periodic) is defined as

$$F(\mu) = \int_{-\infty}^{+\infty} f(t) e^{-j2\pi\mu t} dt$$

Inverse Fourier transform $f(t) = \int_{-\infty}^{+\infty} F(\mu) e^{j2\pi\mu t} d\mu$

How to get these formulas?

Let's start the story from Fourier series to Fourier transform…

From Fourier Series to Fourier Transforms

According to Euler formula $e^{j\theta} = \cos\theta + j\sin\theta$

Easy to have

$$\cos n\omega t = \frac{e^{jn\omega t} + e^{-jn\omega t}}{2}, \sin n\omega t = -j\frac{e^{jn\omega t} - e^{-jn\omega t}}{2}$$

Then, Fourier series become

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos n\omega t + b_n \sin n\omega t\right)$$

From Fourier Series to Fourier Transforms

According to Euler formula $e^{j\theta} = \cos\theta + j\sin\theta$

Easy to have

$$\cos n\omega t = \frac{e^{jn\omega t} + e^{-jn\omega t}}{2}, \sin n\omega t = -j\frac{e^{jn\omega t} - e^{-jn\omega t}}{2}$$

Then, Fourier series become

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n\frac{e^{jn\omega t} + e^{-jn\omega t}}{2} - jb_n\frac{e^{jn\omega t} - e^{-jn\omega t}}{2}\right)$$

$$= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(\frac{a_n - jb_n}{2}e^{jn\omega t} + \frac{a_n + jb_n}{2}e^{-jn\omega t}\right)$$

Then, let

$$c_0 = \frac{a_0}{2}, \frac{a_n - jb_n}{2} = c_n, \frac{a_n + jb_n}{2} = d_n$$

Then,

$$f(t) = c_0 + \sum_{n=1}^{\infty}\left(c_n e^{jn\omega t} + d_n e^{-jn\omega t}\right) (1)$$

From Fourier Series to Fourier Transforms

$$c_0 = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)dt,$$

$$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\left(\cos n\omega t - j\sin n\omega t\right)dt = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{-jn\omega t}dt \quad (2)$$

$$d_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\left(\cos n\omega t + j\sin n\omega t\right)dt = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{jn\omega t}dt$$

We can see that

$$d_n = c_{-n}$$

Thus,

$$\sum_{n=1}^{\infty} d_n e^{-jn\omega t} = \sum_{n=1}^{\infty} c_{-n} e^{-jn\omega t} = \sum_{n=-\infty}^{-1} c_n e^{jn\omega t} \quad (3)$$

From Fourier Series to Fourier Transforms

$$f(t) = c_0 + \sum_{n=1}^{\infty}\left(c_n e^{jn\omega t} + d_n e^{-jn\omega t}\right) \text{ (according to (1))}$$

$$= c_0 e^{j0\omega t} + \sum_{n=1}^{\infty} c_n e^{jn\omega t} + \sum_{n=1}^{\infty} d_n e^{-jn\omega t}$$

$$= c_0 e^{j0\omega t} + \sum_{n=1}^{\infty} c_n e^{jn\omega t} + \sum_{n=-\infty}^{-1} c_n e^{jn\omega t} \text{ (according to (3))}$$

$$= \sum_{n=-\infty}^{+\infty} c_n e^{jn\omega t} \text{ ,where } c_n \text{ is defined by (2)}$$

This is the Fourier series in complex form

How about a non-periodic function?

From Fourier Series to Fourier Transforms

$f(t)$ is a non-periodic function

We make a new function $f_T(t)$ which is periodic and the period is T

$$f_T(t) = f(t), if\ t \in [-T/2, T/2]$$

If $T \to +\infty$, $f_T(t)$ becomes $f(t)$

According to Fourier series

$$f_T(t) = \sum_{n=-\infty}^{+\infty} c_n e^{jn\omega t}, c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t) e^{-jn\omega t} dt$$

Let $s_n = n\omega$

$$f_T(t) = \sum_{n=-\infty}^{+\infty} \left(\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t) e^{-js_n t} dt \right) e^{js_n t} = \frac{1}{T}\sum_{n=-\infty}^{+\infty} \left(\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t) e^{-js_n t} dt \right) e^{js_n t}$$

From Fourier Series to Fourier Transforms

$$f_T(t) = \frac{1}{T}\sum_{n=-\infty}^{+\infty}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t)e^{-js_nt}dt\right)e^{js_nt}$$

when $T \to +\infty$

$$f(t) = \lim_{T\to+\infty} f_T(t) = \lim_{T\to+\infty}\frac{1}{T}\sum_{n=-\infty}^{+\infty}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t)e^{-js_nt}dt\right)e^{js_nt}$$

$$\Delta s = s_n - s_{n-1} = \omega = \frac{2\pi}{T} \quad \Rightarrow \quad T = \frac{2\pi}{\Delta s}$$

$$f(t) = \lim_{\Delta s\to 0}\frac{\Delta s}{2\pi}\sum_{n=-\infty}^{+\infty}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t)e^{-js_nt}dt\right)e^{js_nt}$$

$$= \frac{1}{2\pi}\lim_{\Delta s\to 0}\sum_{n=-\infty}^{+\infty}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t)e^{-js_nt}dt\right)e^{js_nt}\Delta s$$

From Fourier Series to Fourier Transforms

$$f(t)=\frac{1}{2\pi}\lim_{\Delta s\to 0}\sum_{n=-\infty}^{+\infty}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t)e^{-js_nt}dt\right)e^{js_nt}\Delta s$$

when $T\to+\infty(\Delta s\to 0)$

$s_n \Rightarrow s$, $\Delta s \Rightarrow ds$, $\sum \Rightarrow \int$

$$f(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty} f(t)e^{-jst}dt\right)e^{jst}ds$$

Denote by $F(s)$

$$\begin{cases} F(s)=\int_{-\infty}^{+\infty} f(t)e^{-jst}dt & \text{Fourier transform} \\ f(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(s)e^{jst}ds & \text{Inverse Fourier transform} \end{cases}$$

From Fourier Series to Fourier Transforms

$$\begin{cases} F(s) = \int_{-\infty}^{+\infty} f(t) e^{-jst} dt \\ f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(s) e^{jst} ds \end{cases}$$

s here actually is the angular frequency

In the signal processing domain, we usually use another form by substituting s by $s = 2\pi\mu$, where μ is the frequency (measured by Herz)

$$\begin{cases} F(\mu) = \int_{-\infty}^{+\infty} f(t) e^{-j2\pi\mu t} dt \\ f(t) = \int_{-\infty}^{+\infty} F(\mu) e^{j2\pi\mu t} d\mu \end{cases}$$

Related Concepts to Fourier Transform

- Fourier transform $F(\mu)$ is complex in general

$$F(\mu) = \int_{-\infty}^{+\infty} f(t)\cos(2\pi\mu t)dt - j\int_{-\infty}^{+\infty} f(t)\sin(2\pi\mu t)dt$$

$$\equiv R(\mu) + jI(\mu)$$

In polar form, it can be expressed as

$$F(\mu) = |F(\mu)|e^{j\phi(\mu)}$$

where

$$|F(\mu)| = (R^2(\mu) + I^2(\mu))^{1/2}$$ ***Fourier Spectrum***

$$\phi(u) = \text{atan}\,2\frac{I(\mu)}{R(\mu)}$$ ***Phase Angle***

$P(\mu) = |F(\mu)|^2 = R^2(\mu) + I^2(\mu)$ is called the ***power spectrum***

Related Concepts to Fourier Transform

Implementation Tips

1) For computing the image's Fourier transform, you can use fft2()
2) ifft2() can compute the inverse Fourier transform
3) abs() can compute the Fourier spectrum
4) angle() can compute the phase angle

Outline

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction

Symmetry Properties of the Fourier Transform

Real $f(t)$	Fourier transform $F(\mu)$	Symmetry of $F(\mu)$
general	complex	conjugate symmetric $\left(F(-\mu)=F^*(\mu)\right)$
even $\left(f(t)=f(-t)\right)$	only real	even $\left(F(\mu)=F(-\mu)\right)$
odd $\left(-f(t)=f(-t)\right)$	only imaginary	odd $\left(-F(\mu)=F(-\mu)\right)$

Proof?

Why Study Fourier Transform?

- Observe the image in the frequency domain; has some related applications, e.g., de-noising and phase-based image matching; directly manipulating the image in the frequency domain
- We can make use of Fourier transform to compute the convolution efficiently; thanks to FFT

The underlying theory is the convolution theorem!

Convolution Theorem

- Still remember the convolution? (Lecture 3)

  For 1D continuous case, it is defined as

$$f(t) * h(t) = \int_{-\infty}^{+\infty} f(\tau) h(t-\tau) d\tau$$

- Convolution theorem
  - The Fourier transform of a convolution is the point-wise product of Fourier transforms

  if $\mathcal{F}\left(f(t)\right) = F(\mu), \mathcal{F}(h(t)) = H(\mu)$

  then $\mathcal{F}\left(f(t) * h(t)\right) = F(\mu) \cdot H(\mu)$

  Proof:

Convolution Theorem

$$\mathcal{F}\left(f(t)*h(t)\right)=\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty}f(\tau)h(t-\tau)d\tau\right)e^{-j2\pi\mu t}dt$$

$$=\int_{-\infty}^{+\infty}f(\tau)\left(\int_{-\infty}^{+\infty}h(t-\tau)e^{-j2\pi\mu t}dt\right)d\tau \quad \text{(Let } x=t-\tau\text{)}$$

$$=\int_{-\infty}^{+\infty}f(\tau)\left(\int_{-\infty}^{+\infty}h(x)e^{-j2\pi\mu(x+\tau)}dx\right)d\tau$$

$$=\int_{-\infty}^{+\infty}f(\tau)\left(\int_{-\infty}^{+\infty}h(x)e^{-j2\pi\mu x}dx\right)e^{-j2\pi\mu\tau}d\tau$$

$$=\int_{-\infty}^{+\infty}f(\tau)H(\mu)e^{-j2\pi\mu\tau}d\tau$$

$$=H(\mu)\int_{-\infty}^{+\infty}f(\tau)e^{-j2\pi\mu\tau}d\tau=H(\mu)F(\mu)$$

Revisit Gaussian Filter

In spatial domain

$$G(x,y)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

In frequency domain

$$\mathcal{G}(u,v)=\exp\left(-\frac{(u^2+v^2)\sigma^2}{2}\right)=\exp\left(-\frac{(u^2+v^2)}{2\left(\frac{1}{\sigma}\right)^2}\right)$$

The Fourier transform of a Gaussian function is also of a Gaussian shape in the frequency domain

Revisit Gaussian Filter

Gaussian filter is a low-pass filter

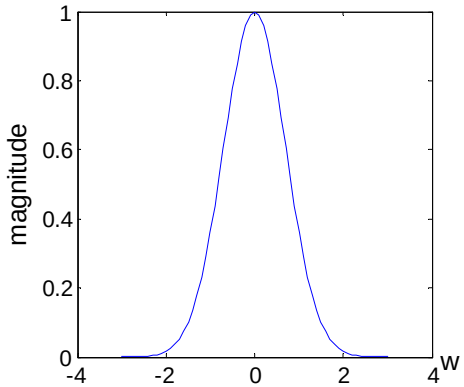
Consider the 1D case

$f(x)$ if filtered by $g(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$

What will happen to the frequency components of $f(x)$?

$$FT\left(f(x) * g(x)\right) = F(\omega)\cdot G(\omega) = F(\omega)\cdot \exp\left(-\frac{\omega^2}{2\left(\frac{1}{\sigma}\right)^2}\right)$$

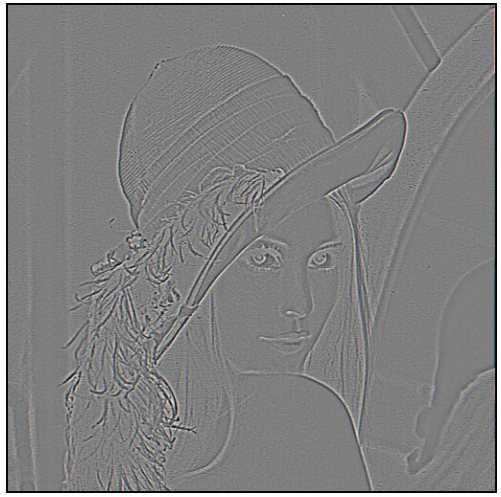
$$= F(\omega)\cdot$$



Revisit Gaussian Filter

original

smoothed (5x5 Gaussian)



smoothed – original

Why does this work?

Outline

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction

Discrete Fourier Transform (DFT) in 1D Case

Given a discrete sequence with M points

$$f = [f_0 \;\; f_1, \ldots., \; f_{M-1}]$$

Its DFT also has M points

$$F = [F_0 \;\; F_1, \ldots., \; F_{M-1}]$$

and

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}, u = 0,1,2,\ldots,M-1$$

$$\text{IDFT} \;\; f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M}, x = 0,1,2,\ldots,M-1$$

For DFT, there is a fast algorithm for computation, FFT (Fast Fourier Transform)

Discrete Fourier Transform (DFT) in 2D Case

In continuous case

$$F(u,v)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy$$

$$f(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{j2\pi(ux+vy)}dudv$$

In discrete case

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

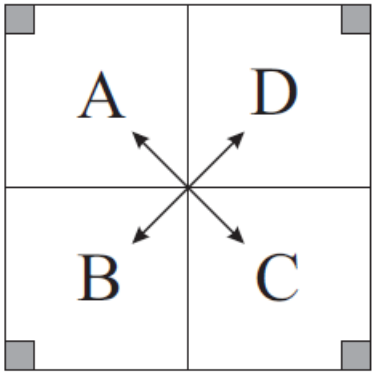
$$where\ u=0,1,\ldots,M-1; v=0,1,\ldots,N-1$$

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

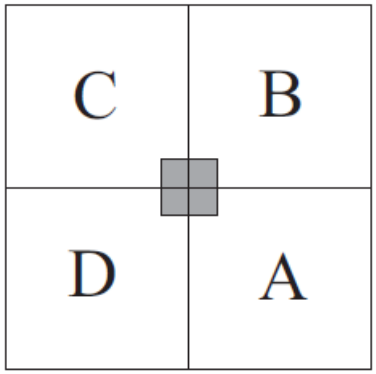
$$where\ x=0,1,\ldots,M-1; y=0,1,\ldots,N-1$$

Some Notes on DFT Visualization

- For DFT, the origin is not at the center of the matrix
  - Assume the original spectrum is divided into four quadrants; the small gray-filled squares in the corners represent positions of low frequencies
  - Due to the symmetries of the spectrum the quadrant positions can be swapped diagonally and the low frequencies locations appear in the middle of the image



original spectrum
low frequencies in corners



shifted spectrum
with the origin at ($M/2$, $N/2$)

Some Notes on DFT Visualization

- **For visualization**, we usually rearrange the DFT matrix to make its low frequencies at the center of the rectangle; it equals to $f(x,y)(-1)^{x+y}$

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$$

Implementation Tips

In Matlab, it can simply implemented by using fftshift

DFT Visualization Samples

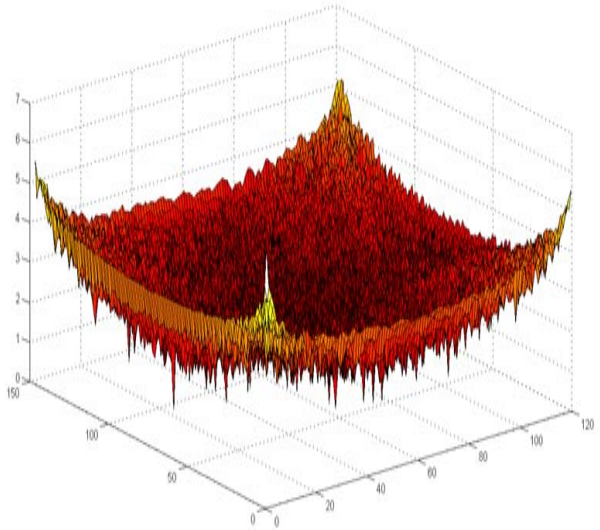
- Since each field of the Fourier transform is a complex number, we cannot show Fourier map in a single figure; instead, magnitude and phase maps are shown separately

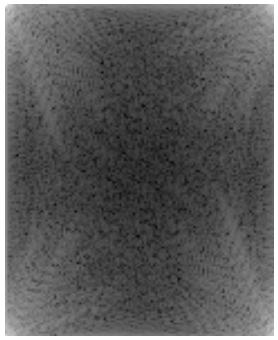
```
im = imread('im.bmp');
figure;
imshow(im,[]);

imfft = abs(fft2(im));
imfftlog = log10(1+imfft);
figure;
imshow(imfftlog,[]);

imfftshifted = fftshift(imfftlog);
figure;
imshow(imfftshifted,[]);
```

DFT Visualization Samples

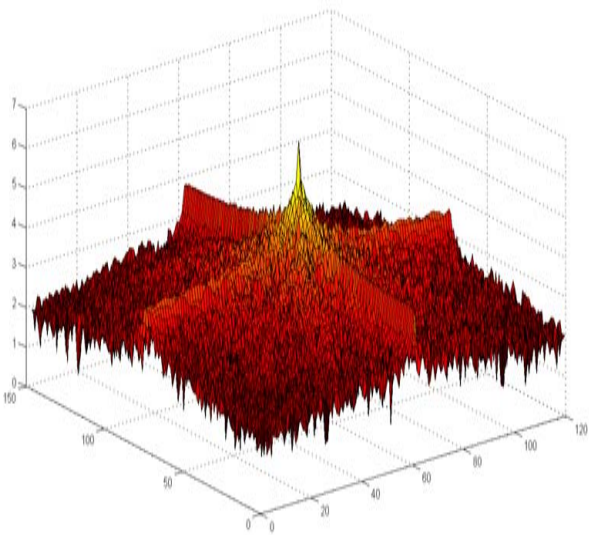
- An example

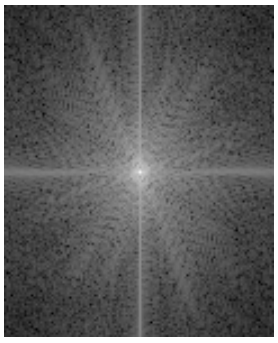
Original image





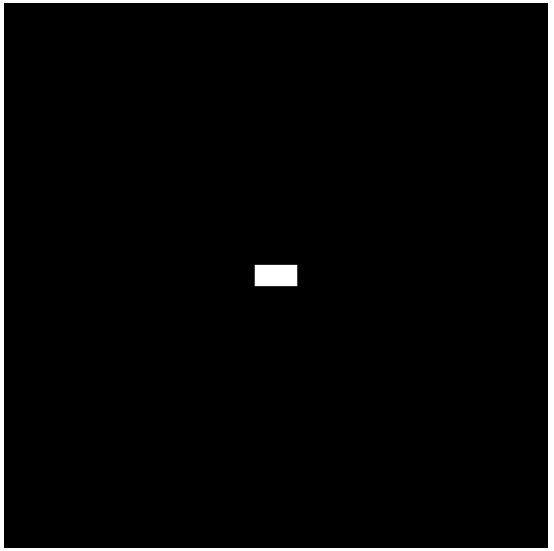
Spectrum without using fftshift



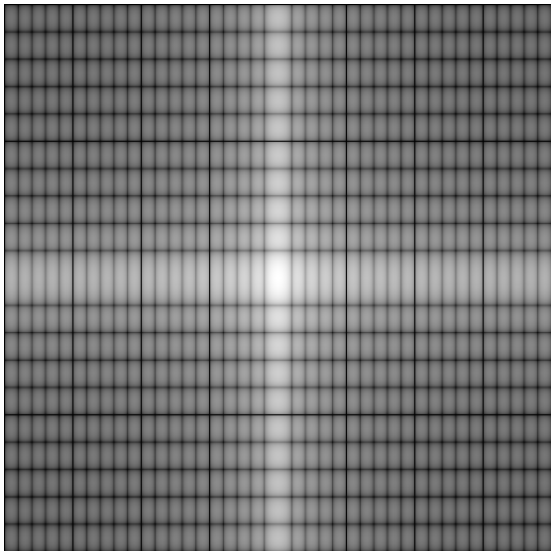


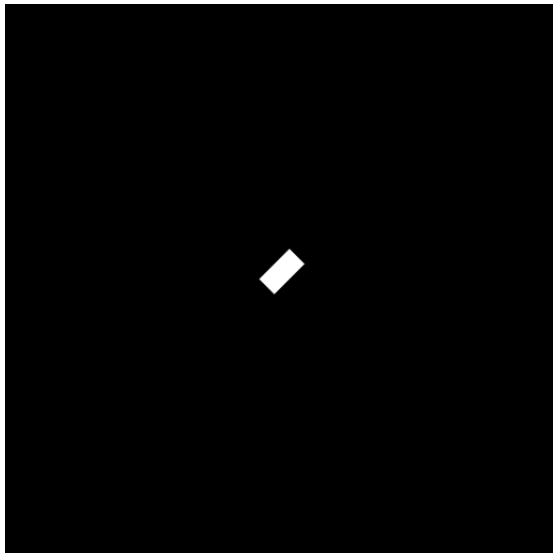
Spectrum using fftshift

DFT Visualization Samples

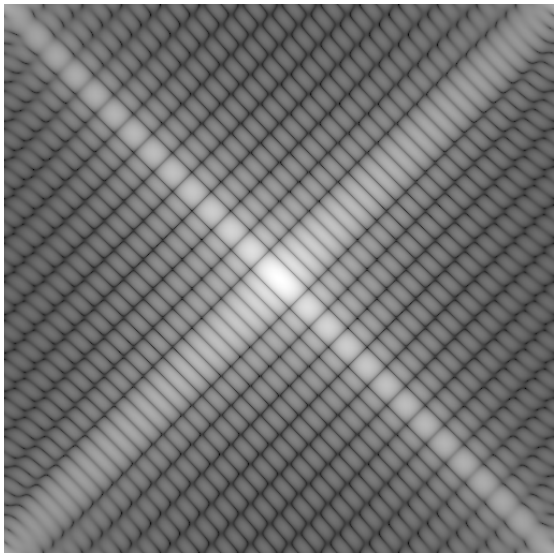


DFT





DFT



Any relationship?

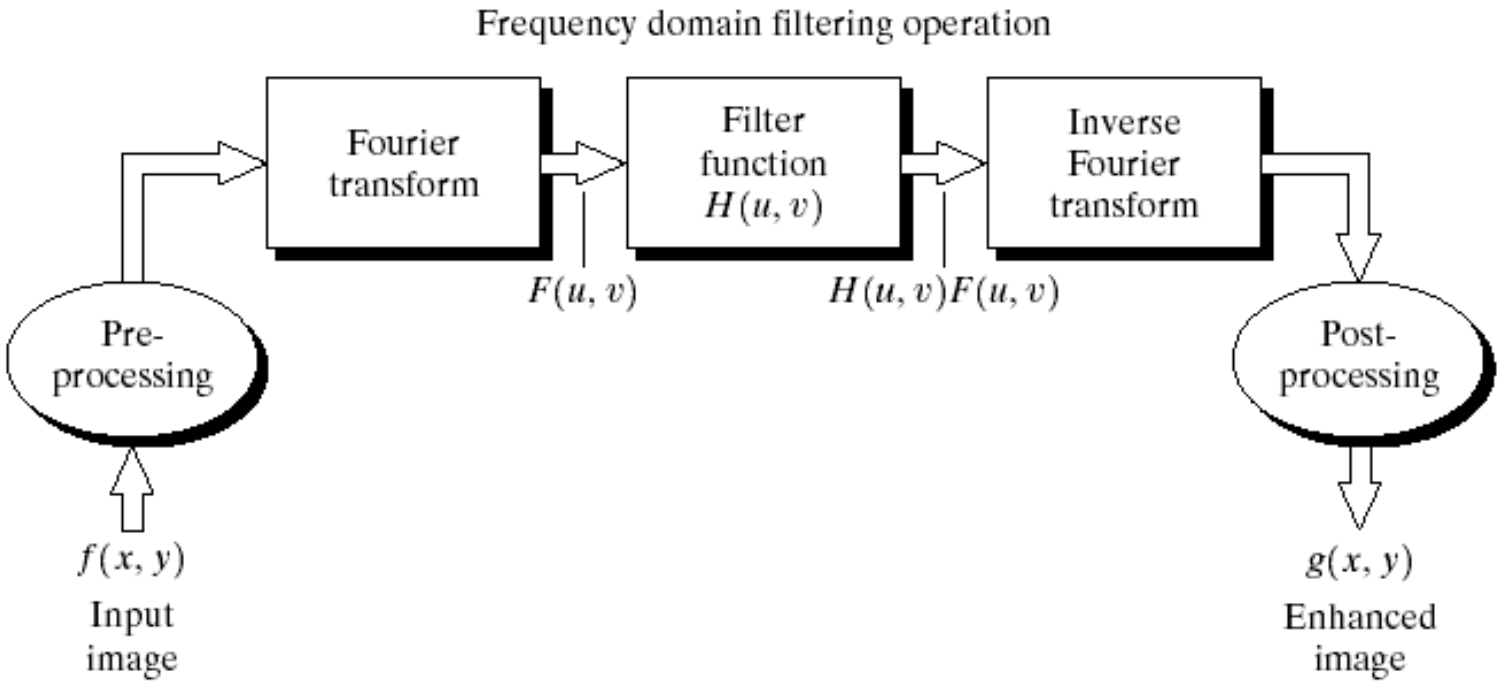
Outline

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction

The Basics of Filtering in the Frequency Domain

To filter an image in the frequency domain:

1. Compute $F(u,v)$, the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result



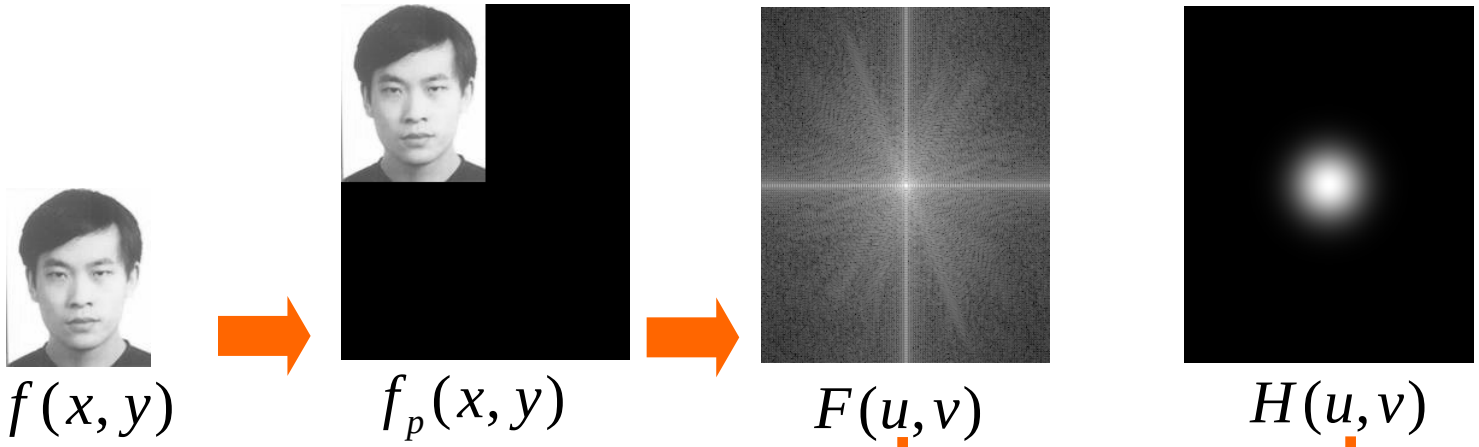
Directly Filtering in the Frequency Domain

1. Given an image $f(x, y)$ of size $M \times N$, set $P = 2M$ and $Q = 2N$
2. Form a padded image, $f_p(x, y)$ of size $P \times Q$ by appending the necessary number of zeros to $f(x, y)$
3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center its transform
4. Compute $F(u, v)$ of $f_p(x, y)$
5. Generate a filter function $H(u, v)$ of the size $P \times Q$
6. Get the modified Fourier transform $G(u,v) = F(u,v)H(u,v)$
7. Obtain the processed image

$$g_p(x, y) = \mathcal{F}^{-1}(G(u,v))(-1)^{x+y}$$

8. Obtain the final result $g(x, y)$ by extracting the $M \times N$ region from the top, left corner of $g_p(x, y)$

Directly Filtering in the Frequency Domain—Example



$f(x,y)$ $f_p(x,y)$ $F(u,v)$ $H(u,v)$

$G(u,v)=F(u,v)H(u,v)$ $g_p(x,y)$ $g(x,y)$

Source codes are available on our course website

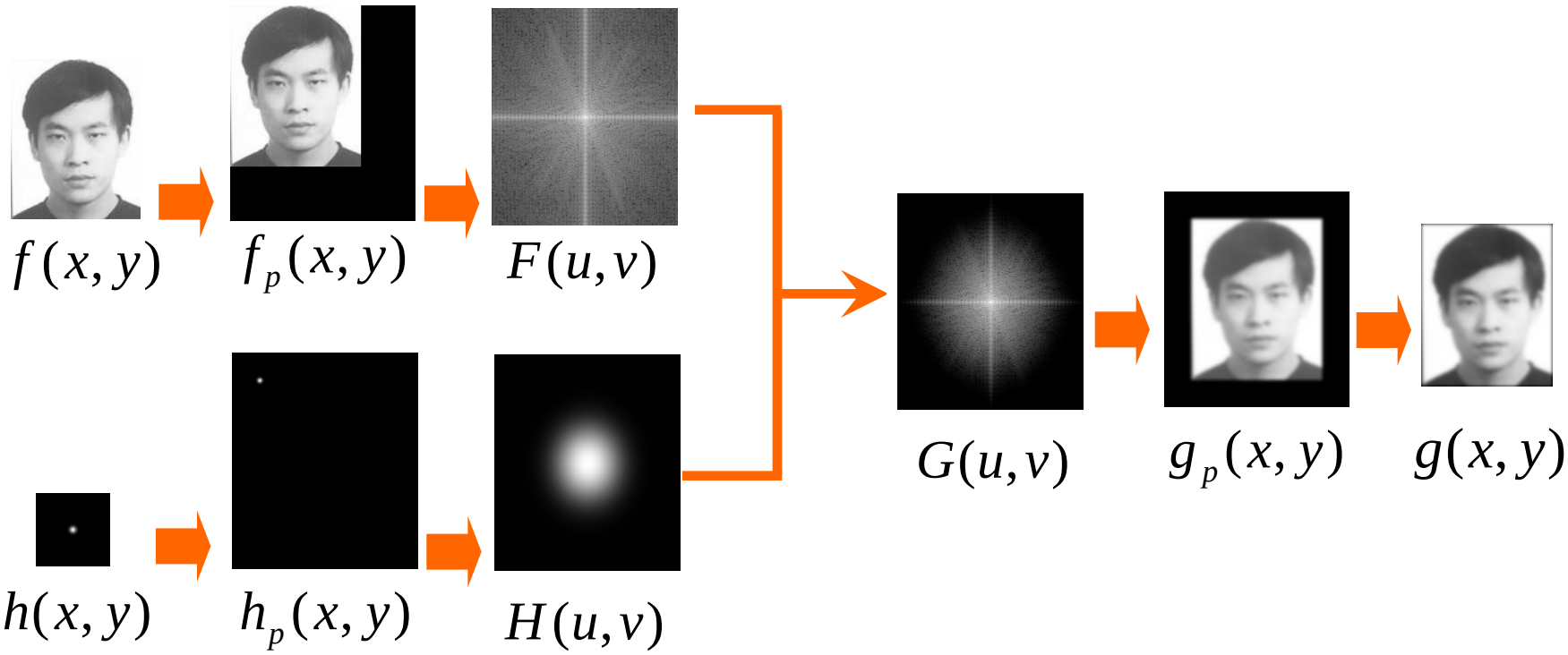
Convolution via Fourier Transform

1. Given an image $f(x, y)$ of size $A \times B$, and a spatial filter $h(x, y)$ of size $C \times D$; set $P >= A+C-1$ and $Q >= B+D-1$
2. Form a padded image f_p of size $P \times Q$ by appending the necessary number of zeros to $f(x, y)$; form a padded filter h_p of size $P \times Q$ in a similar way
3. Compute the DFT $F(u, v)$ of the image, and $H(u, v)$ of the filter
4. Get the modified Fourier transform $G(u,v) = F(u,v)H(u,v)$
5. Obtain the processed image

$$g_p(x, y) = \mathcal{F}^{-1}(G(u,v))$$

6. Obtain the final result $g(x, y)$ by extracting the central $A \times B$ region from $g_p(x, y)$

Convolution via Fourier Transform—Example



Source codes are available on our course website

Some Tips on Filtering via Fourier Transform

- When the filter kernel is small, it's better to implement the filtering in the spatial domain; otherwise, you can realize the filtering via the Fourier transform
- In practice, when padding the images or filters, it's better to make it has a size which is the power of 2; this criterion is based on the computer architecture

Outline

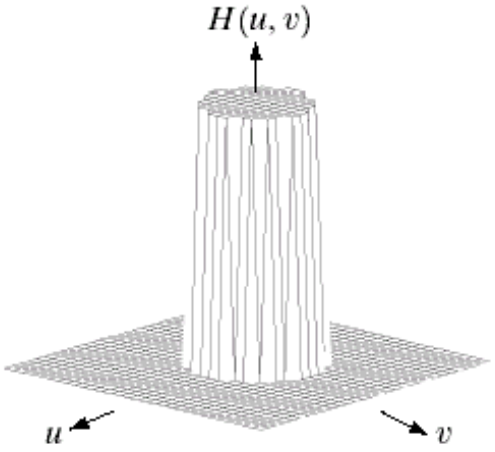
- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction

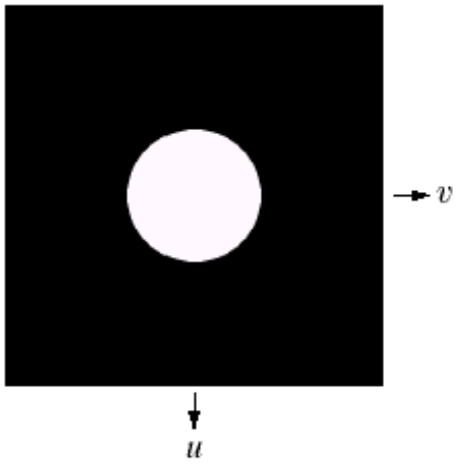
Smoothing is Low-Pass Filtering

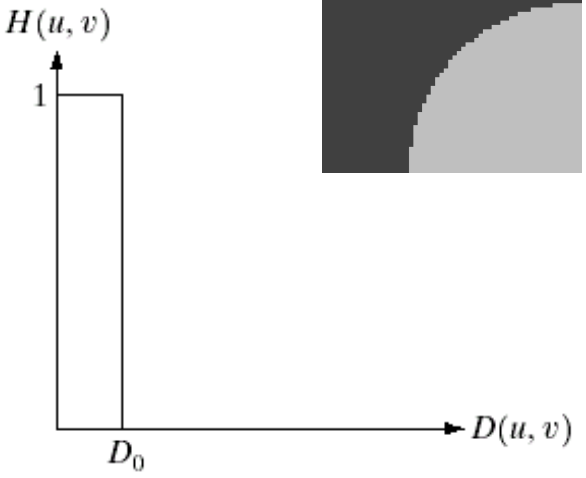
- Image smoothing actually is performing a low-pass filtering to the image
- Edges and other sharp intensity transitions, such as noise, in an image contribute significantly to the high frequency content of its Fourier transform
- Three commonly used low-pass filtering techniques
  - Ideal low-pass filters
  - Butterworth low-pass filters
  - Gaussian low-pass filters

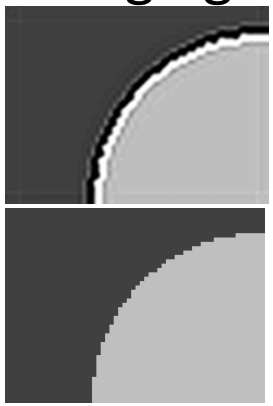
Ideal Low-Pass Filter

- Simply cut off all high frequency components that are within a specified distance D_0 from the origin of the transform
  - Its drawback is that the filtering result has obvious ringing artifacts
  - Ideal low-pass filter is rarely used in practice









Ideal Low-Pass Filter

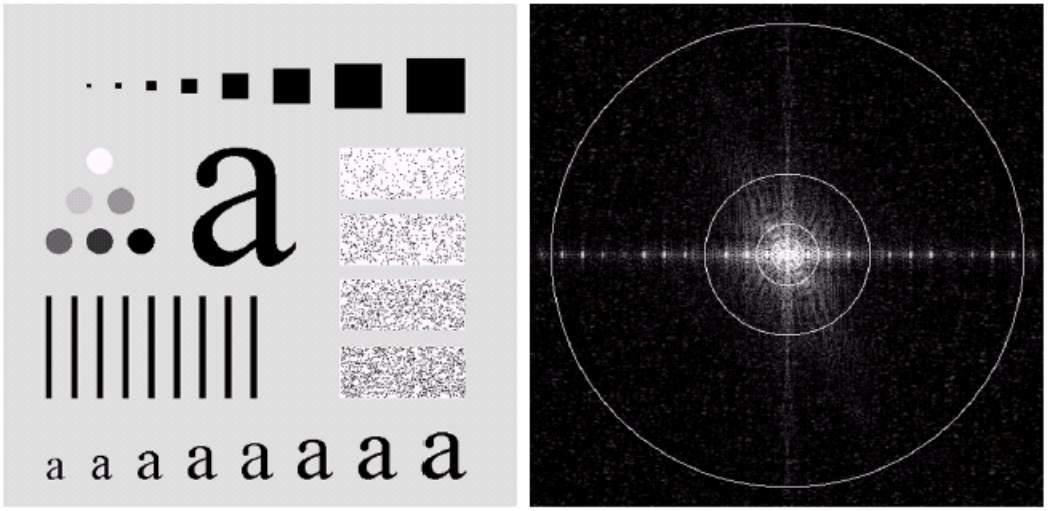
The transfer function for the ideal low pass filter can be given as:

$$H(u,v) = \begin{cases} 1 \text{ if } D(u,v) \le D_0 \\ 0 \text{ if } D(u,v) > D_0 \end{cases}$$

where $D(u,v)$ is the distance of (u, v) to the frequency centre $(0, 0)$ and it is given as:

$$D(u,v) = [u^2 + v^2]^{1/2}$$

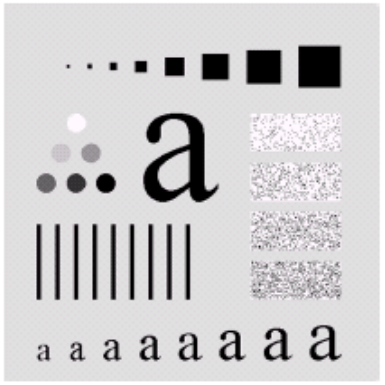
Ideal Low-Pass Filter

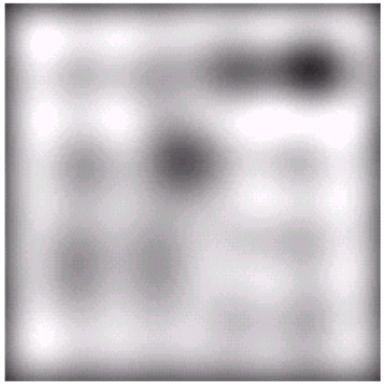


Above we show an image, it's Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it

Ideal Low-Pass Filter

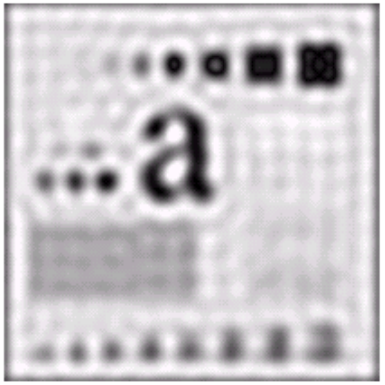
Original image

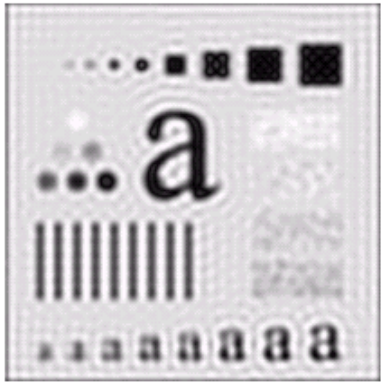




Result of filtering with ideal low pass filter of radius 5

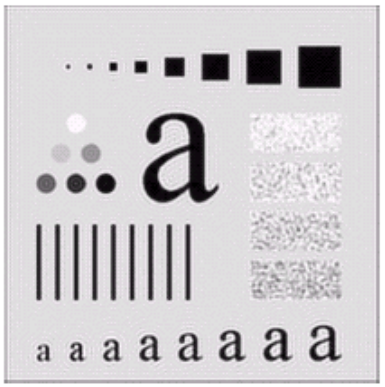
Result of filtering with ideal low pass filter of radius 15

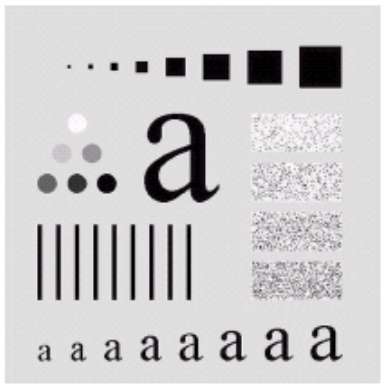




Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80





Result of filtering with ideal low pass filter of radius 230

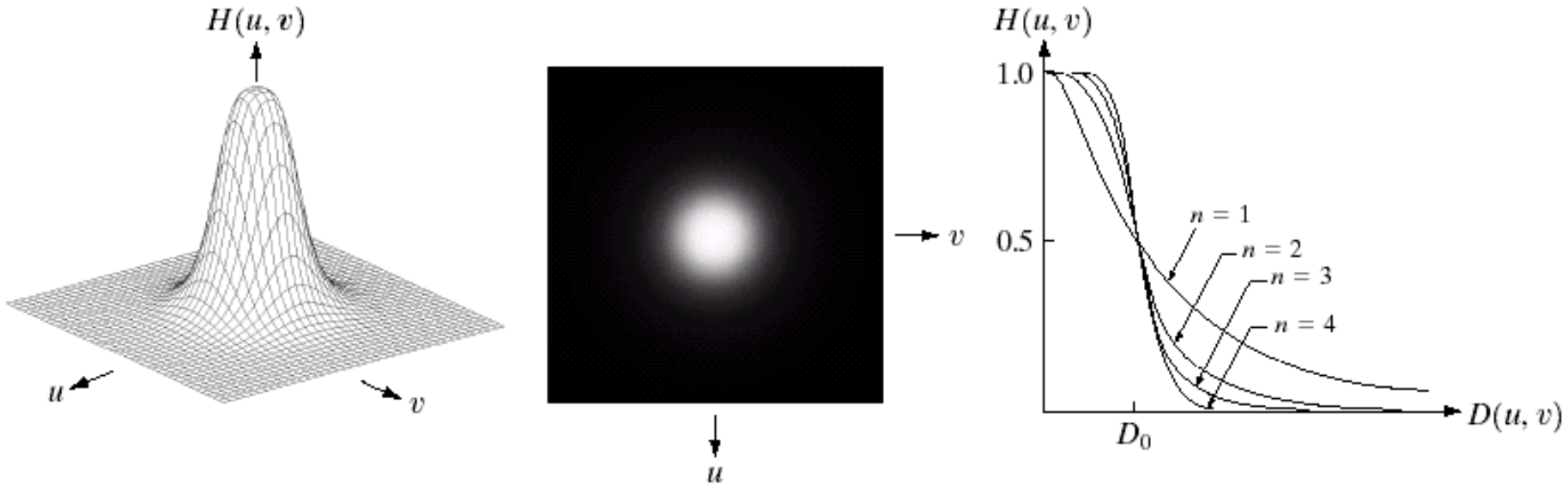
Butterworth Low-pass Filters

- It was proposed by the British engineer Stephen Butterworth
- Filter order can change the shape of the Butterworth filter; for high order values, the Butterworth filter approaches the ideal filter; for low order values, it approaches the Gaussian filter

Butterworth Low-pass Filters

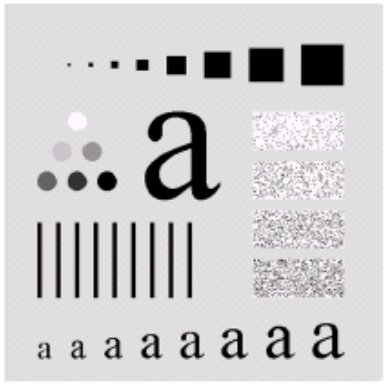
- The transfer function of a Butterworth low-pass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

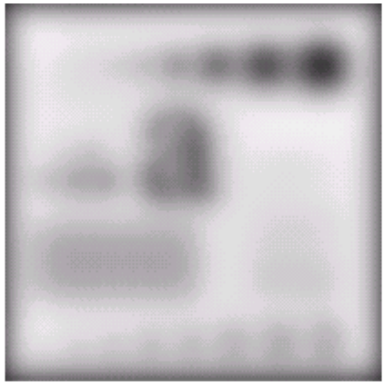
$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$



Butterworth Low-pass Filters

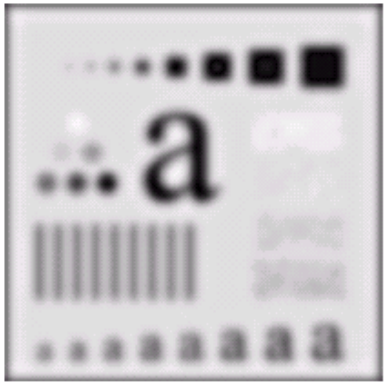
Original image

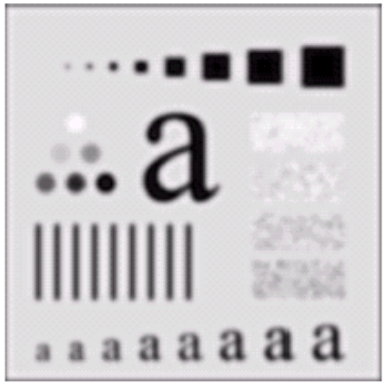




Result of filtering with Butterworth filter of order 2 and cutoff radius 5

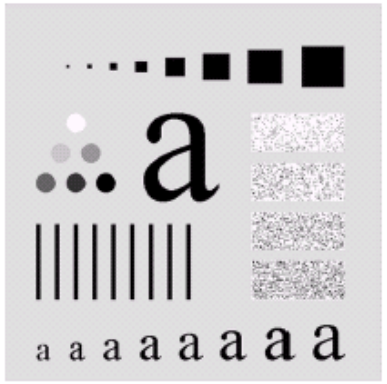
Result of filtering with Butterworth filter of order 2 and cutoff radius 15





Result of filtering with Butterworth filter of order 2 and cutoff radius 30

Result of filtering with Butterworth filter of order 2 and cutoff radius 80



Result of filtering with Butterworth filter of order 2 and cutoff radius 230

Butterworth Low-pass Filters

Original image

Filtering result of Butterworth

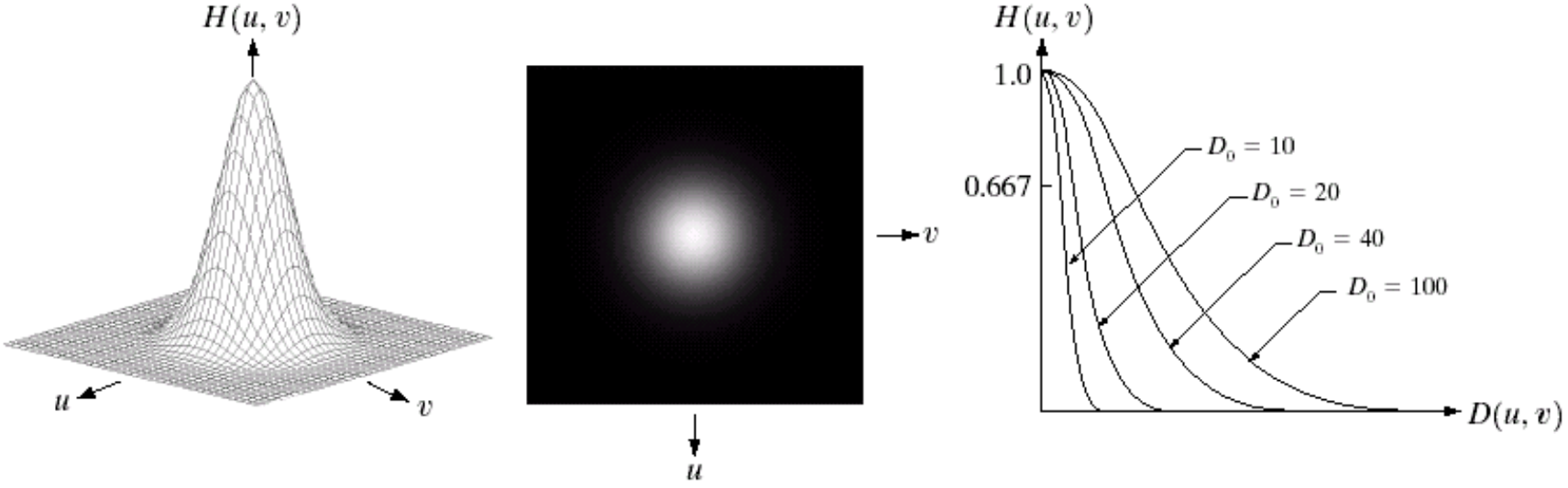
Source codes are available on course website

Gaussian Low-pass Filters

The transfer function of a Gaussian lowpass filter is defined as:

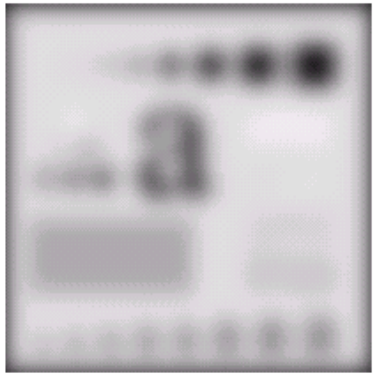
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

where $D(u,v)$ is the distance from the center of the frequency rectangle



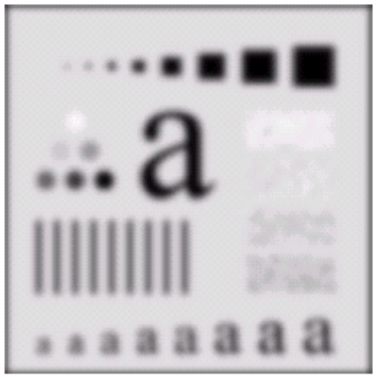
Gaussian Lowpass Filters

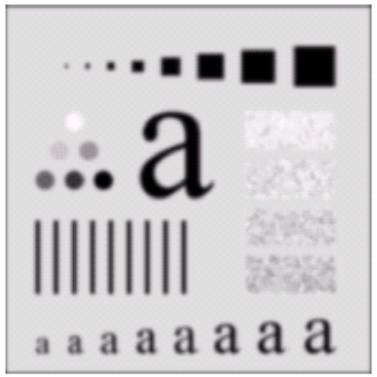
Original image



Result of filtering with Gaussian filter with cutoff radius 5

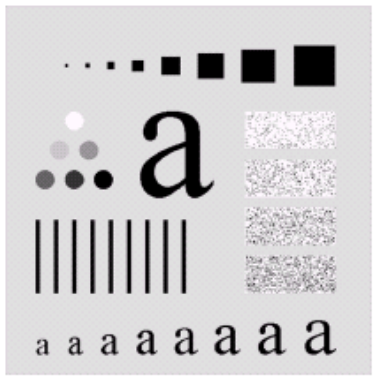
Result of filtering with Gaussian filter with cutoff radius 15





Result of filtering with Gaussian filter with cutoff radius 30

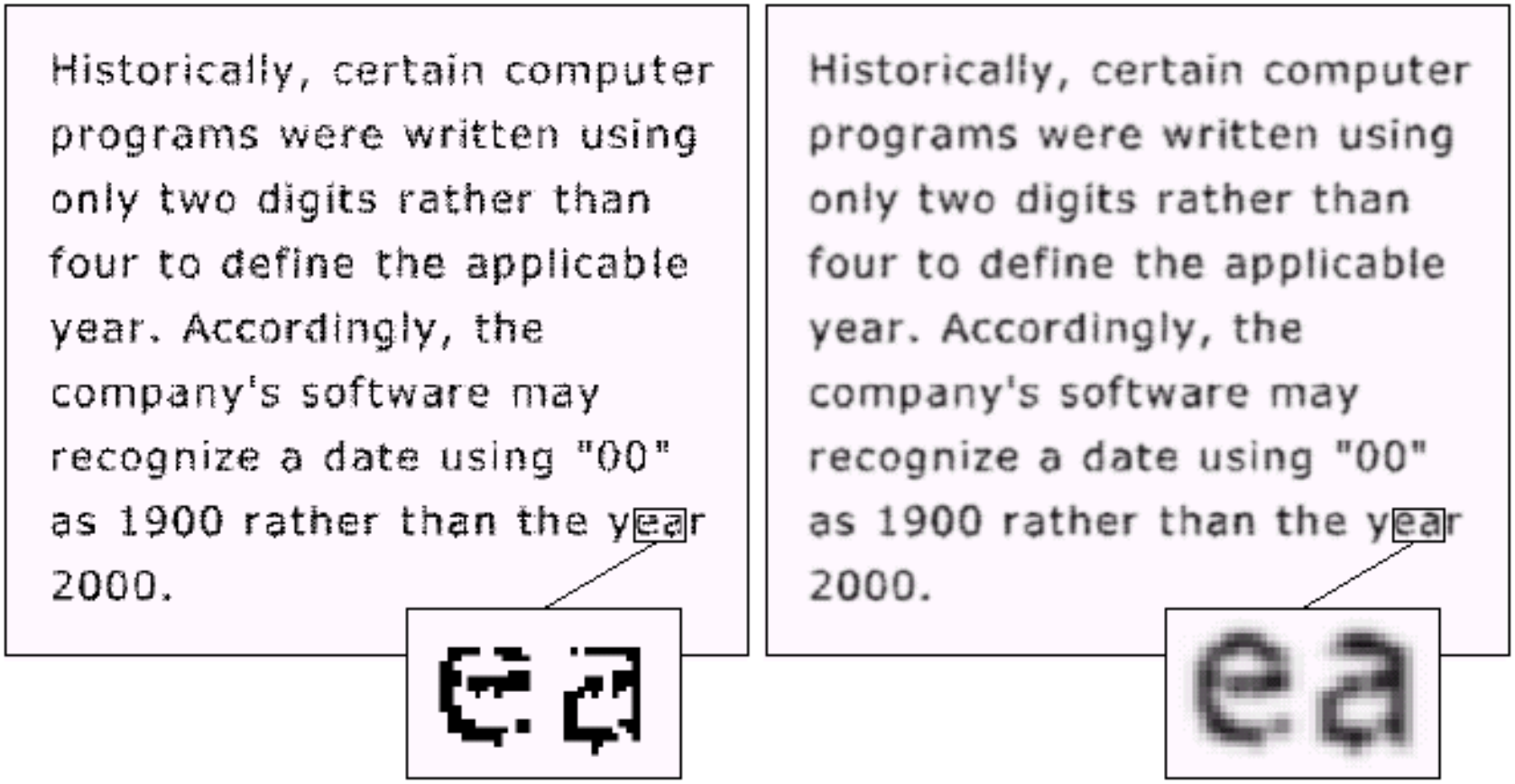
Result of filtering with Gaussian filter with cutoff radius 85



Result of filtering with Gaussian filter with cutoff radius 230

Low-pass Filtering Examples

A low pass Gaussian filter is used to connect broken text



Low-pass Filtering Examples

Gaussian filters used to remove blemishes in a photograph for publishing

Outline

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction

Sharpening in the Frequency Domain

- Edges and fine detail in images are associated with high frequency components
- High pass filters – only pass the high frequencies, drop the low ones
- High pass filters are precisely the reverse of low pass filters, so,

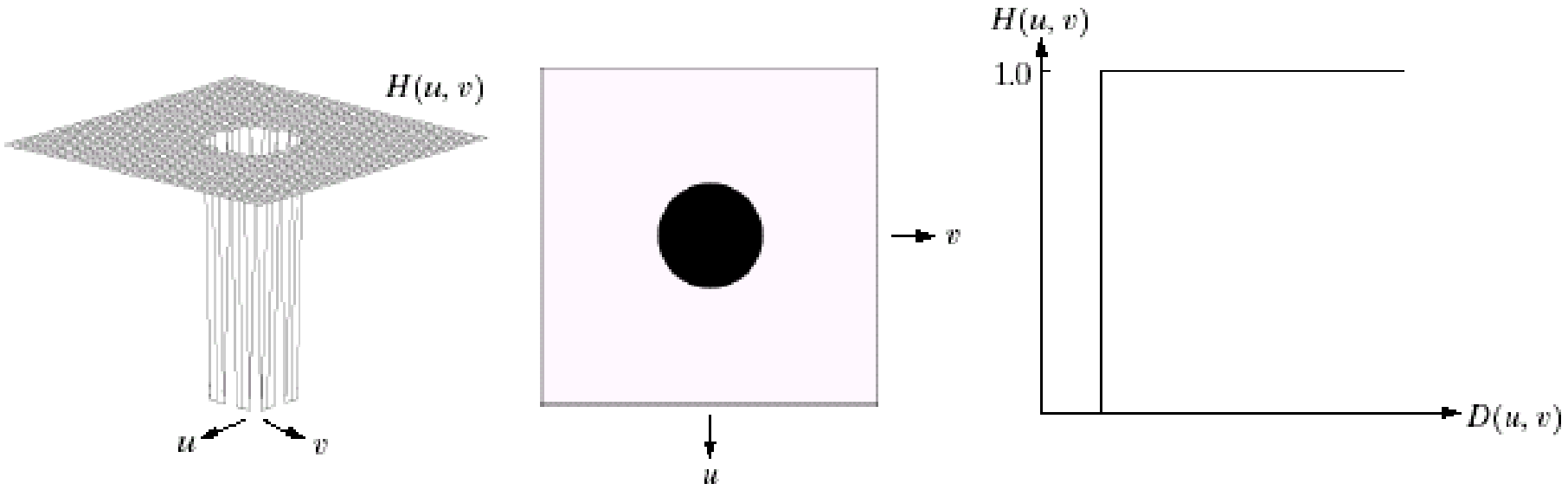
$$H_{HP} = 1 - H_{LP}(u,v)$$

Ideal High-Pass Filters

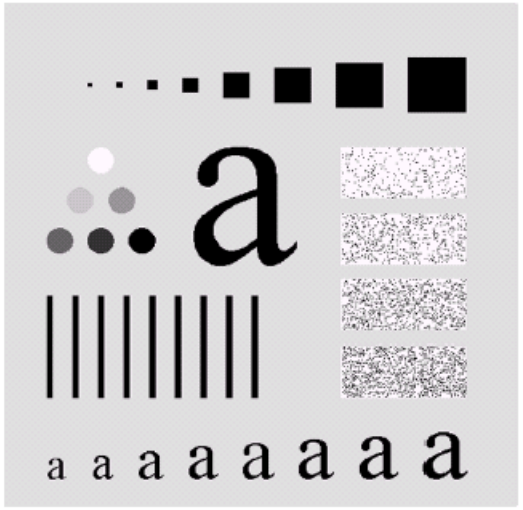
The ideal high pass filter is given as:

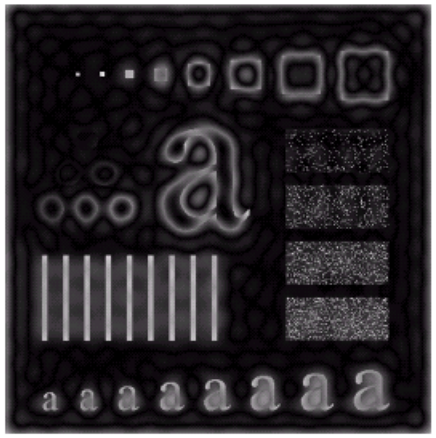
$$H(u,v) = \begin{cases} 0 \text{ if } D(u,v) \le D_0 \\ 1 \text{ if } D(u,v) > D_0 \end{cases}$$

where D_0 is the cut off distance as before

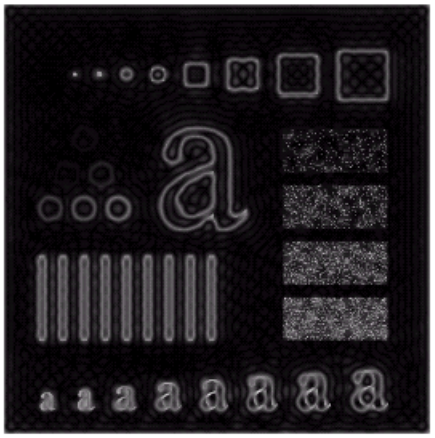


Ideal High-Pass Filters

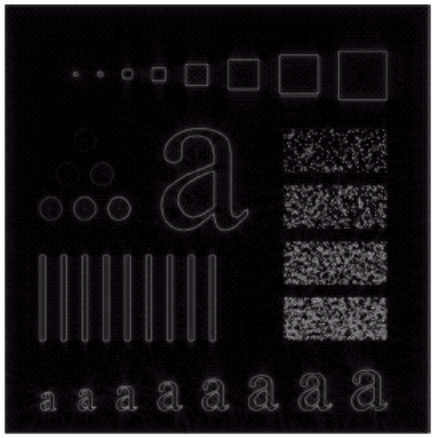




Results of ideal high pass filtering with D_0 = 15



Results of ideal high pass filtering with D_0 = 30



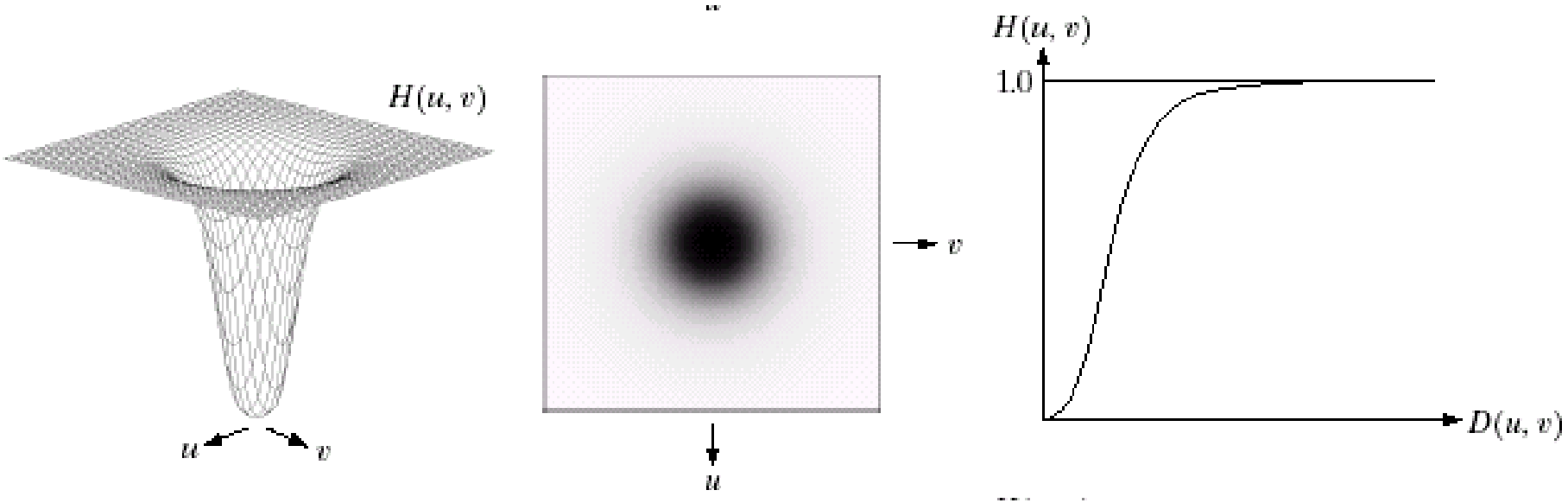
Results of ideal high pass filtering with D_0 = 80

Butterworth High Pass Filters

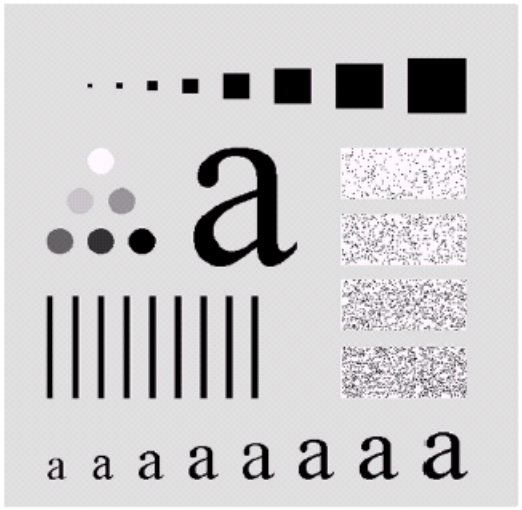
The Butterworth high pass filter is given as:

$$H(u,v) = \frac{1}{1+[D_0 / D(u,v)]^{2n}}$$

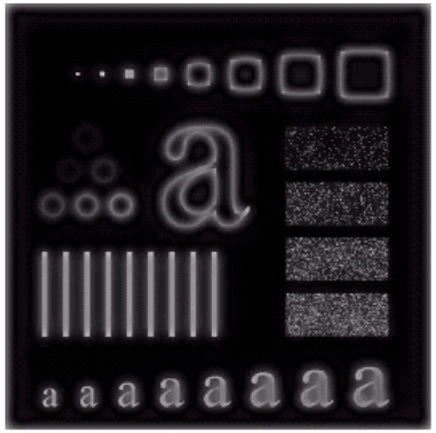
where n is the order and D_0 is the cut off distance as before

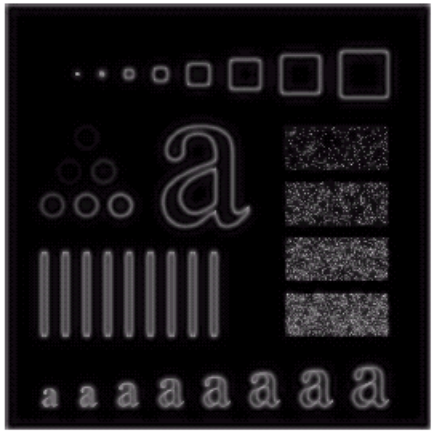


Butterworth High Pass Filters



Results of Butterworth high pass filtering of order 2 with D_0 = 15



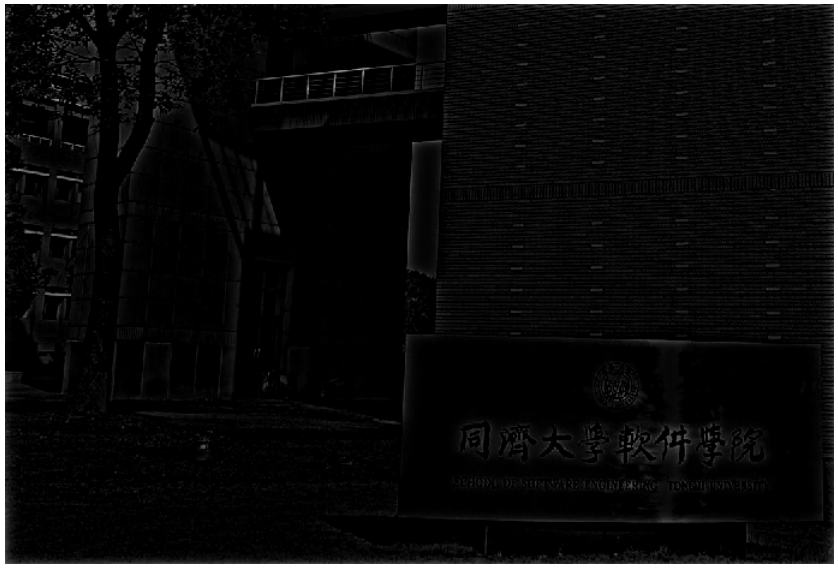


Results of Butterworth high pass filtering of order 2 with D_0 = 80

Results of Butterworth high pass filtering of order 2 with D_0 = 30

Butterworth High Pass Filters

Original image



Filtering result of Butterworth high-pass filtering

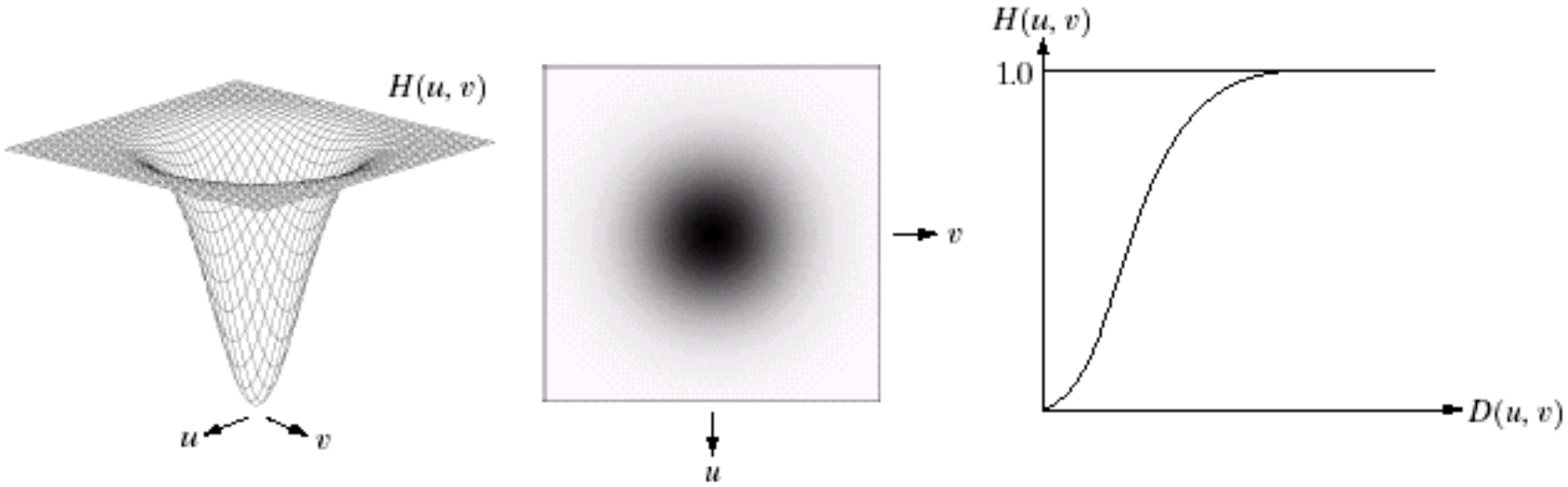
Source codes are available on course website

Gaussian High Pass Filters

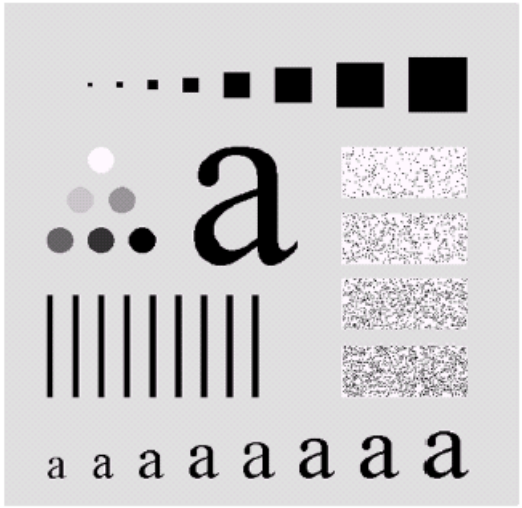
The Gaussian high pass filter is given as:

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

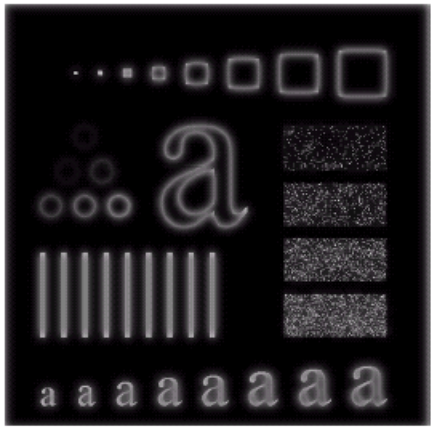
where D_0 is the cut off distance as before



Gaussian High Pass Filters



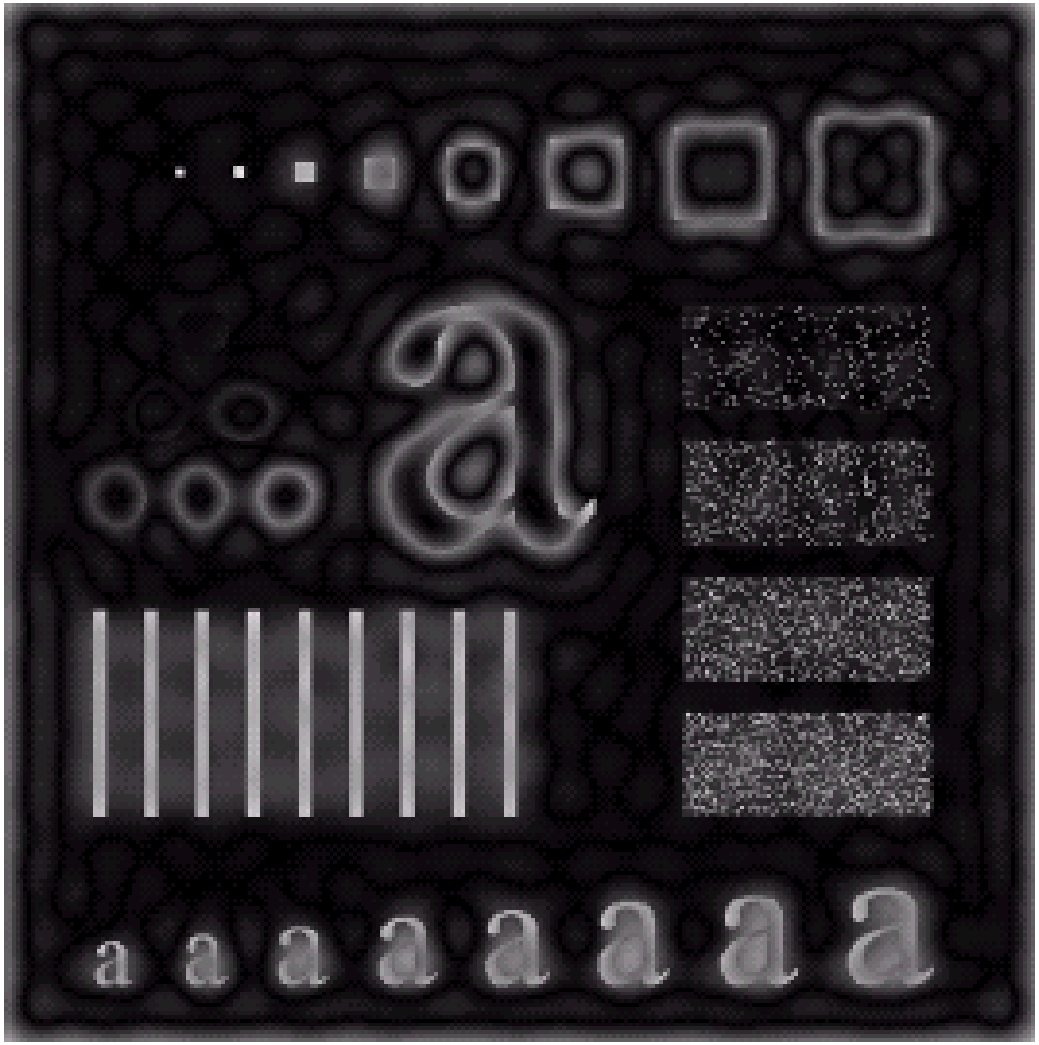
Results of Gaussian high pass filtering with D_0 = 15



Results of Gaussian high pass filtering with D_0 = 80

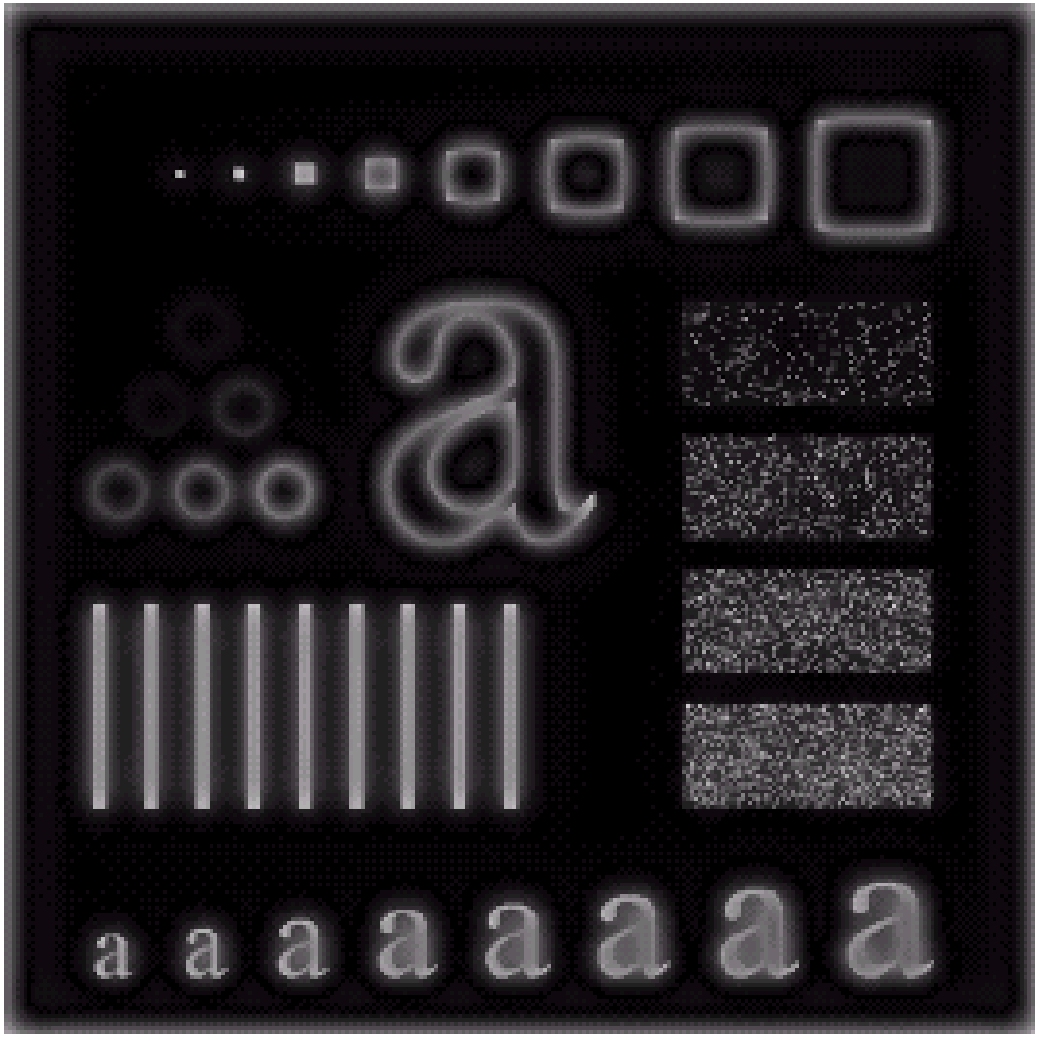
Results of Gaussian high pass filtering with D_0 = 30

Highpass Filter Comparison



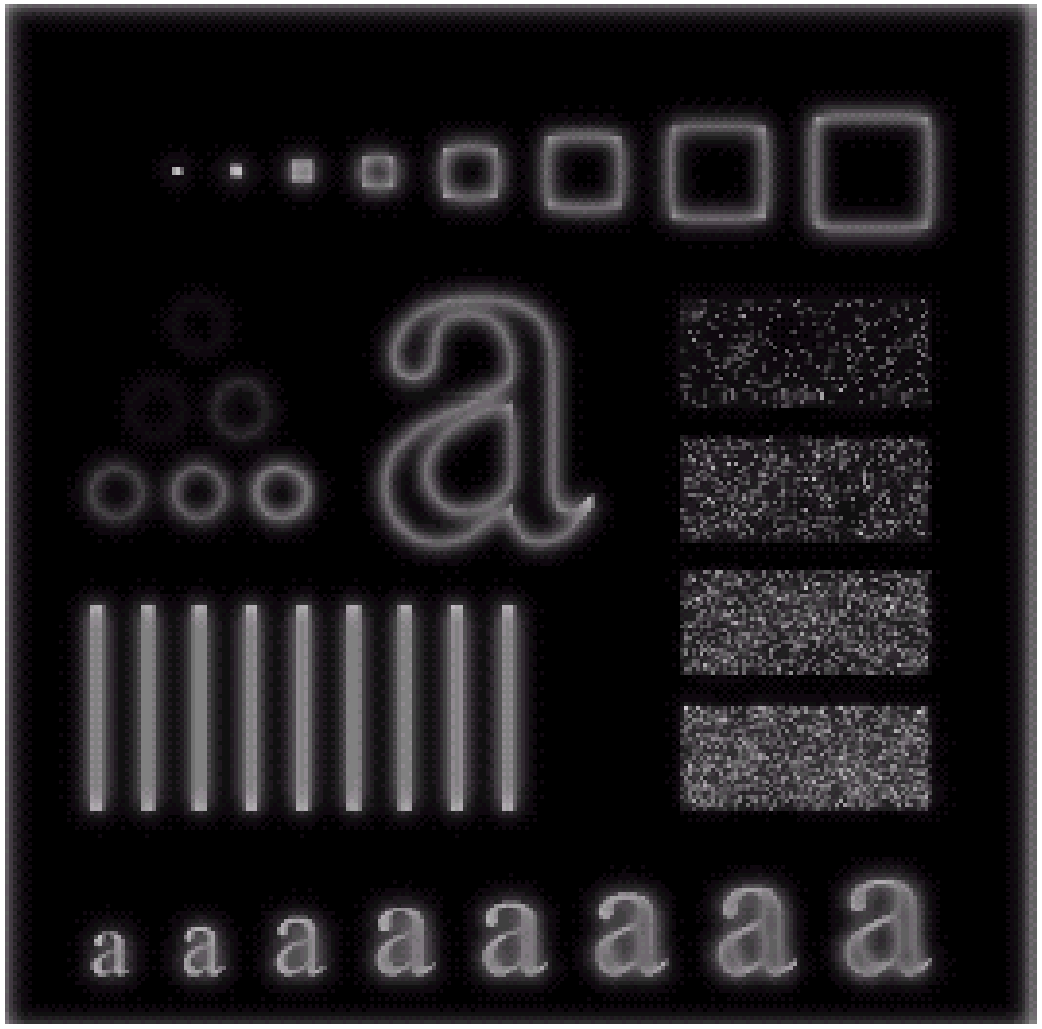
Results of ideal high pass filtering with D_0 = 15

Highpass Filter Comparison



Results of Butterworth high pass filtering of order 2 with D_0 = 15

Highpass Filter Comparison



Results of Gaussian high pass filtering with D_0 = 15

Outline

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters
- Periodic noise reduction
  - Selective filtering
  - Periodic noise reduction by selective filters

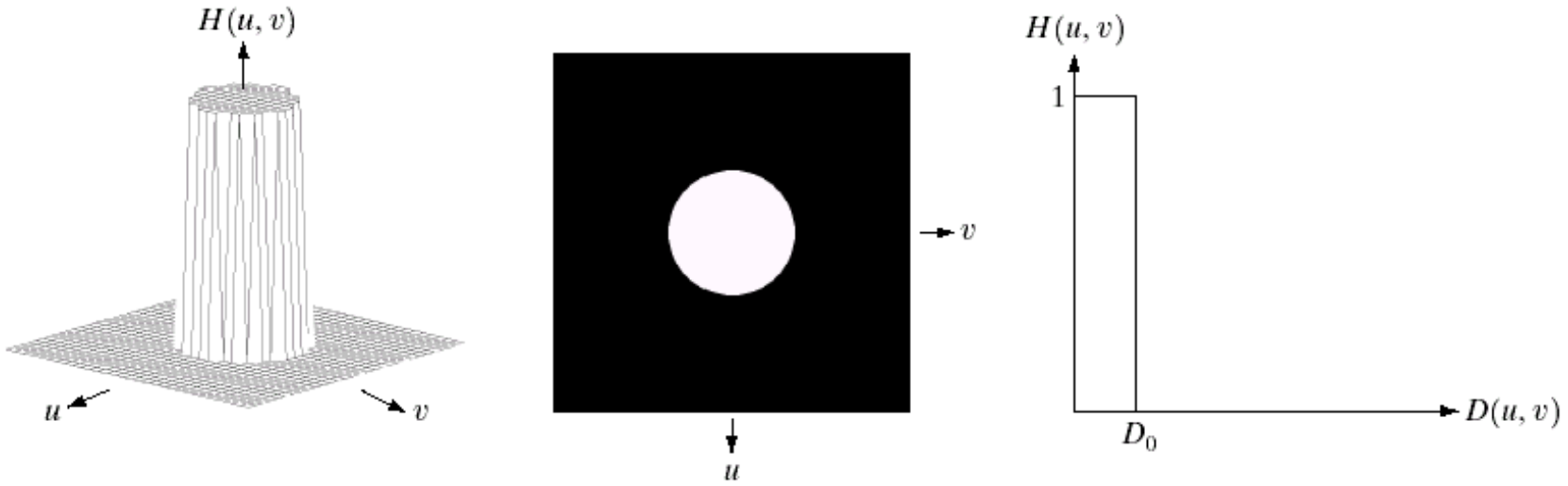
Selective Filtering

- Review: low pass filters (lecture 4)

The transfer function for the ideal low pass filter can be given as:

$$H(u,v) = \begin{cases} 1 \text{ if } D(u,v) \le D_0 \\ 0 \text{ if } D(u,v) > D_0 \end{cases}$$

where $D(u,v)$ is the distance of (u, v) to the frequency centre (0, 0) and it is given as: $D(u,v) = [u^2 + v^2]^{1/2}$



Selective Filtering

- Bandreject filters (*remove frequency components within a specific range*)

Ideal
$$H(u,v)=\begin{cases}0 & if\ D_0-\frac{W}{2}\le D(u,v)\le D_0+\frac{W}{2}\\ 1 & otherwise\end{cases}$$

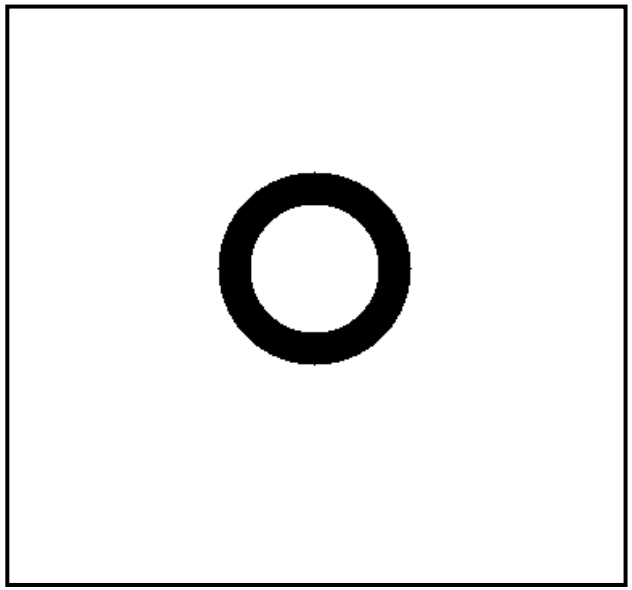
Butterworth
$$H(u,v)=\frac{1}{1+\left[\frac{WD(u,v)}{D^2(u,v)-D_0^2}\right]^{2n}}$$

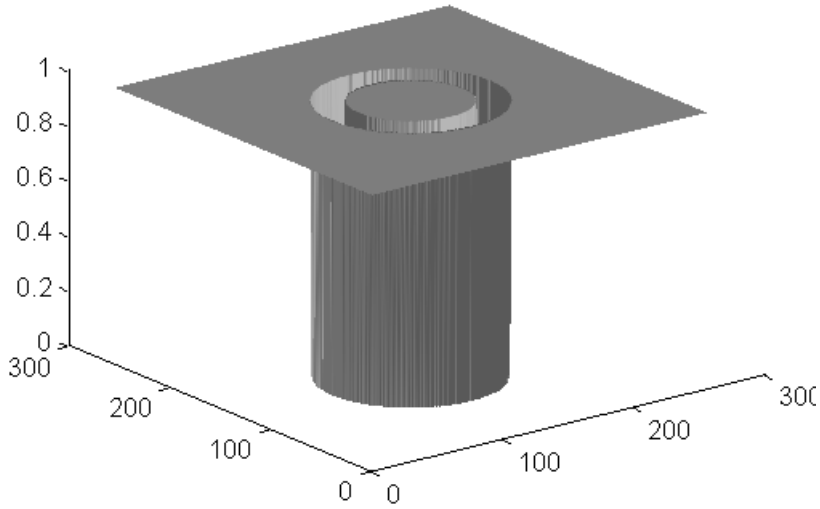
Gaussian
$$H(u,v)=1-e^{-\left[\frac{D^2(u,v)-D_0^2}{WD(u,v)}\right]^2}$$

where W is the width of the band

Selective Filtering

- Bandreject filters (*remove frequency components within a specific range*)

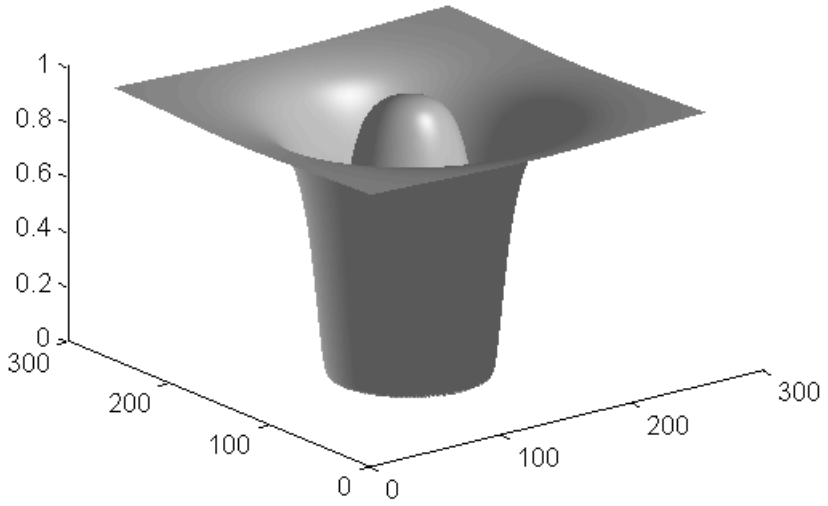




ideal band reject filter

Selective Filtering

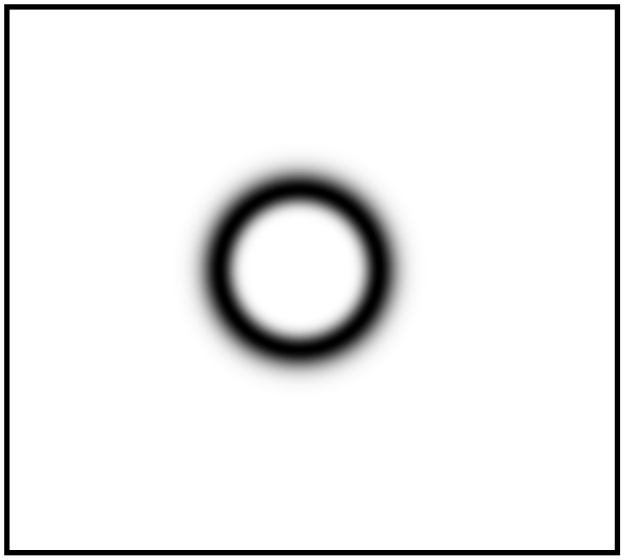
- Bandreject filters (*remove frequency components within a specific range*)

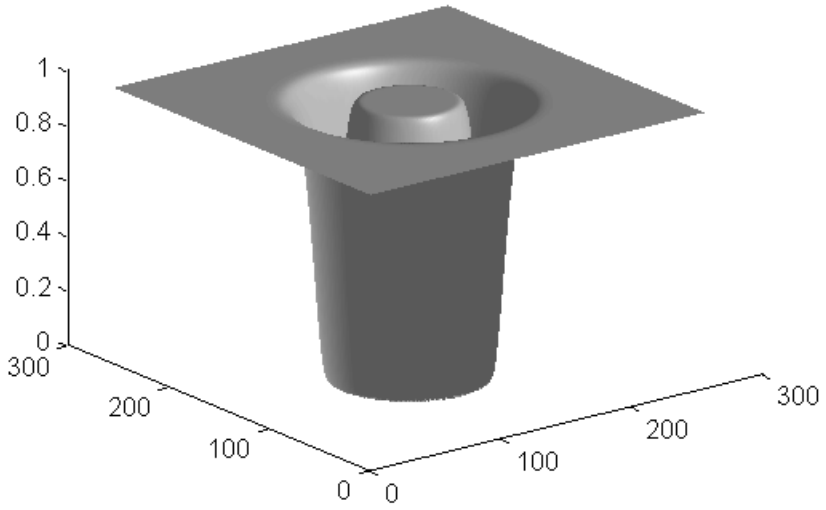


Butterworth band reject filter (of order 1)

Selective Filtering

- Bandreject filters (*remove frequency components within a specific range*)





Gaussian band reject filter

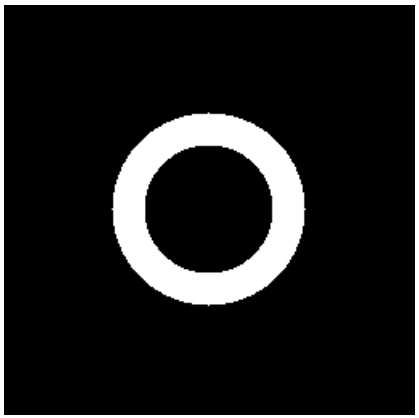
Selective Filtering

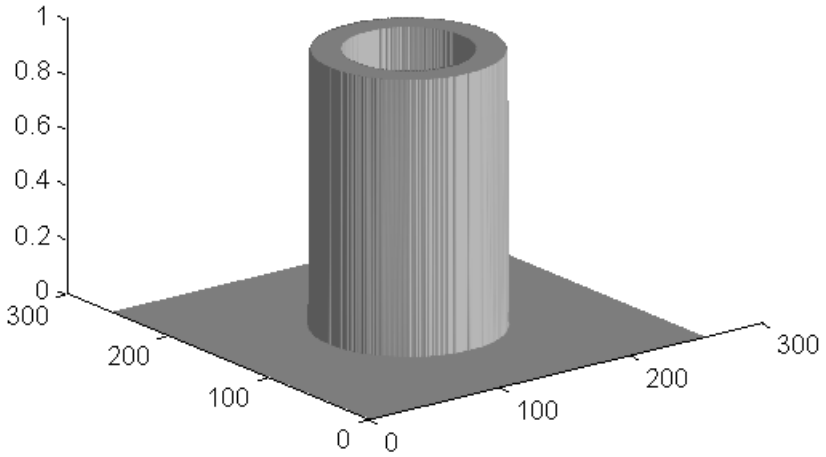
- Bandpass filters
  - Let only a portion of the frequency components pass
  - They can be constructed by

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Selective Filtering

- Bandpass filters
  - Let only a portion of the frequency components pass

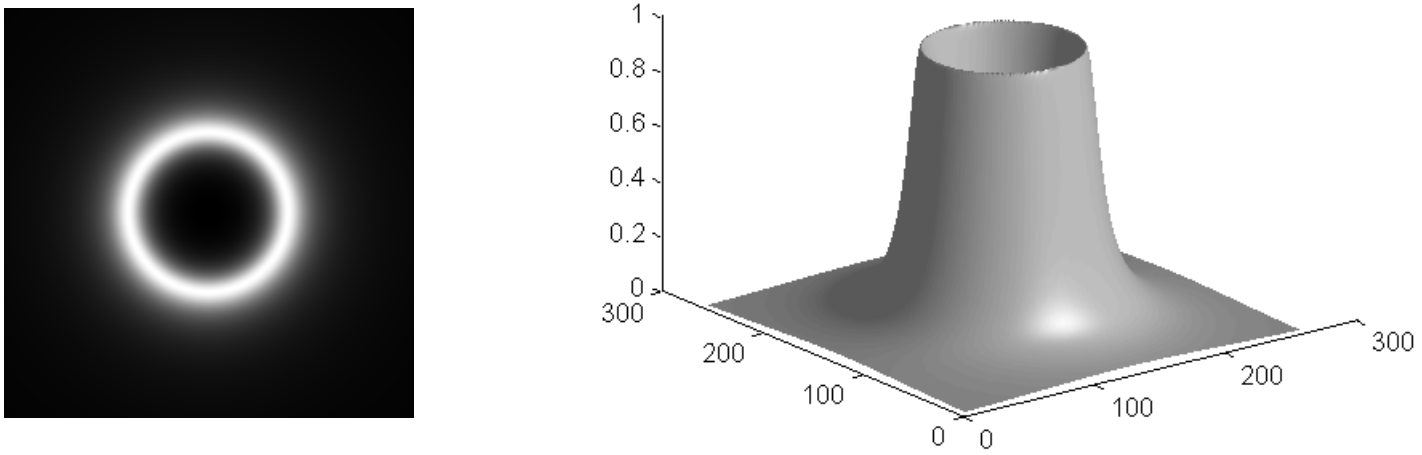




ideal band pass filter

Selective Filtering

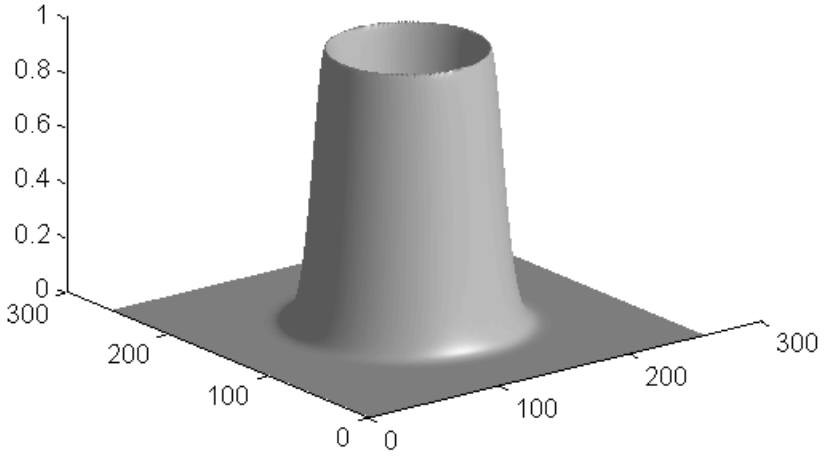
- Bandpass filters
  - Let only a portion of the frequency components pass



Butterworth band pass filter (of order 1)

Selective Filtering

- Bandpass filters
  - Let only a portion of the frequency components pass



Gaussian band pass filter

Selective Filtering

- Notch filters
  - They are the most useful of the selective filters
  - A notch filter rejects (or passes) frequencies in a predefined neighborhood
  - Zero-phase-shifted filters must be symmetric about the origins, so a notch with center at (u_0, v_0) must have a corresponding notch at location $(-u_0, -v_0)$

Selective Filtering

- Notch filters
  - Notch reject filters are constructed as products of highpass filters whose centers are translated to the centers of the notches

$$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v)$$

where $H_k(u,v), H_{-k}(u,v)$ are highpass filters whose centres are at $(u_k, v_k), (-u_k, -v_k)$, respectively

The distance computations for each filter are

$$D_k(u,v) = \left(u - u_k\right)^2 + \left(v - v_k\right)^2$$

$$D_{-k}(u,v) = \left(u + u_k\right)^2 + \left(v + v_k\right)^2$$

Selective Filtering

- Notch filters—An example

Butterworth notch reject filter of order n, containing three notch pairs,

$$H_{NR}(u,v) = \prod_{k=1}^{3}\left[\frac{1}{1+\left[D_{k0}/D_k(u,v)\right]^{2n}}\right]\left[\frac{1}{1+\left[D_{k0}/D_{-k}(u,v)\right]^{2n}}\right]$$

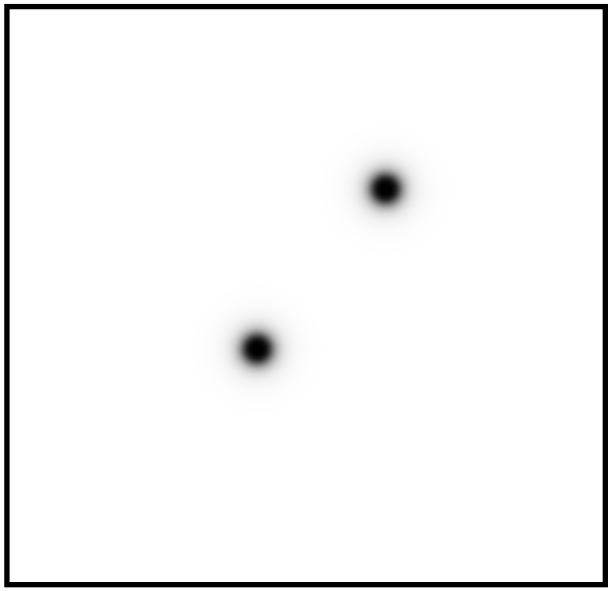
The constant D_{k0} is the same for each pair of notches, but it can be different for different pairs

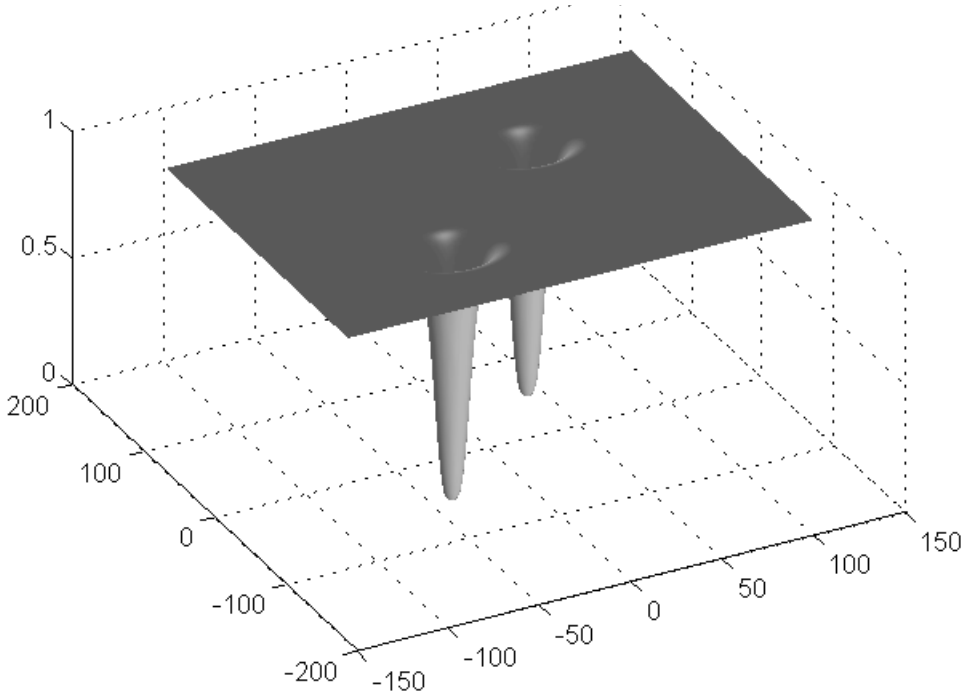
A ***notch pass filter*** is obtained from a notch reject filter by

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$

Selective Filtering

- Notch filters—An example

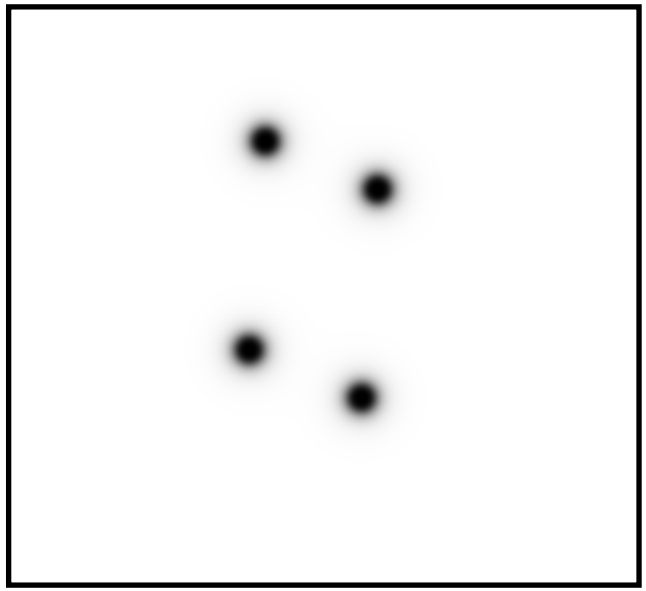


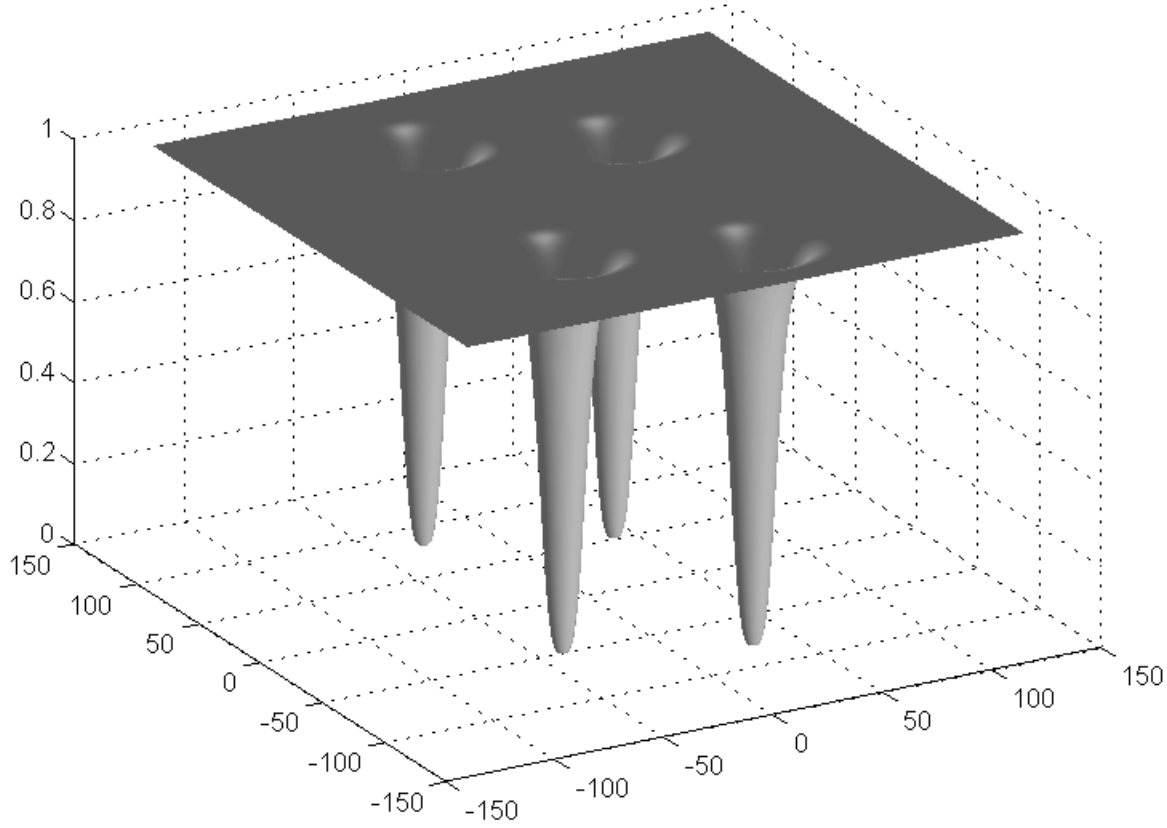


A Butterworth notch reject filter, containing one notch pair

Selective Filtering

- Notch filters—An example





A Butterworth notch reject filter, containing two notch pairs

Source code for this demo is available on our course website

Contents

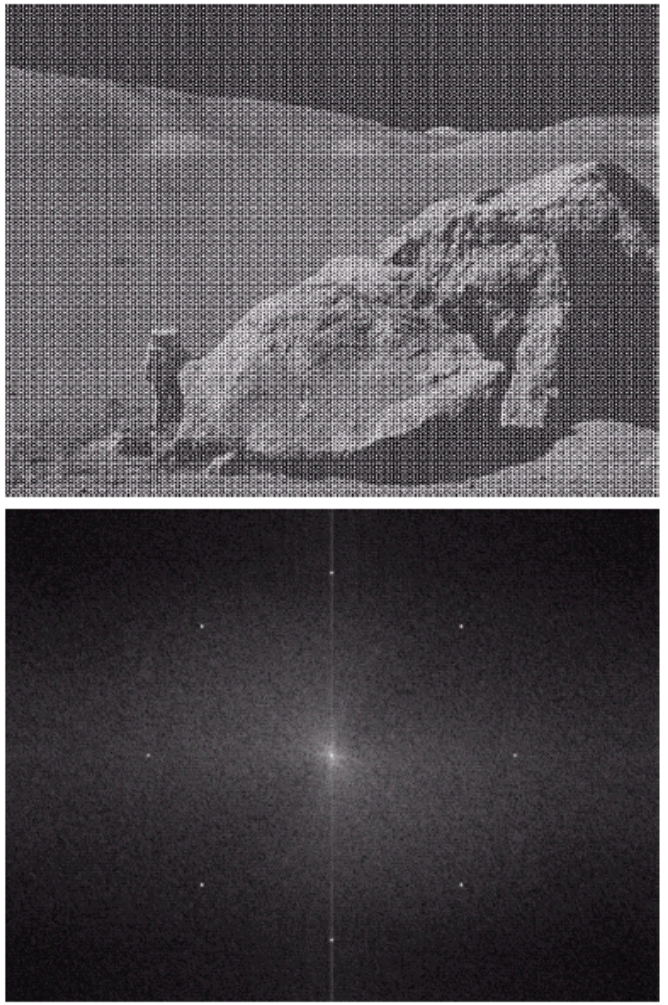
- What is image restoration?
- A model of the image degradation/restoration process
- Noise models
- Additive random noise reduction
- Periodic noise reduction
  - Selective filtering
  - Periodic noise reduction by selective filters

Periodic noise reduction by selective filters

Typically arises due to electrical or electromagnetic interference

Gives rise to regular noise patterns in an image

Frequency domain techniques in the Fourier domain are most effective at removing periodic noise



Periodic noise reduction by selective filters

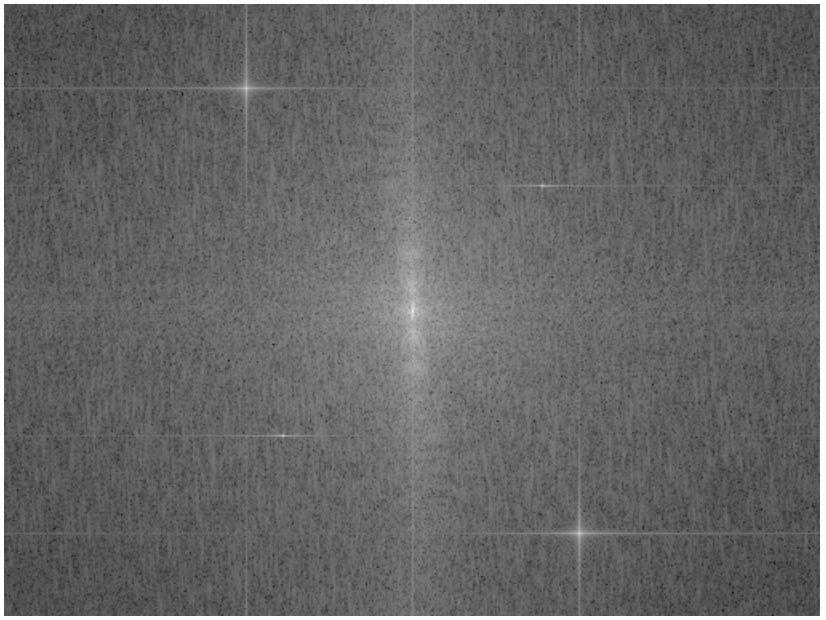
- Notch filters can efficiently remove the periodic noise
  - Step 1: analyze the Fourier spectrum F of the image
  - Step 2: identify the locations of the peaks in F
  - Step 3: construct a notch reject filter H in Fourier domain, whose centers are at peaks
  - Step 4: use H to filter F to get the result

Periodic noise reduction by selective filters

- Notch filters can efficiently remove the periodic noise

  Step 1: analyze the Fourier spectrum F of the image

Noisy image



Fourier spectrum F

Periodic noise reduction by selective filters

- Notch filters can efficiently remove the periodic noise

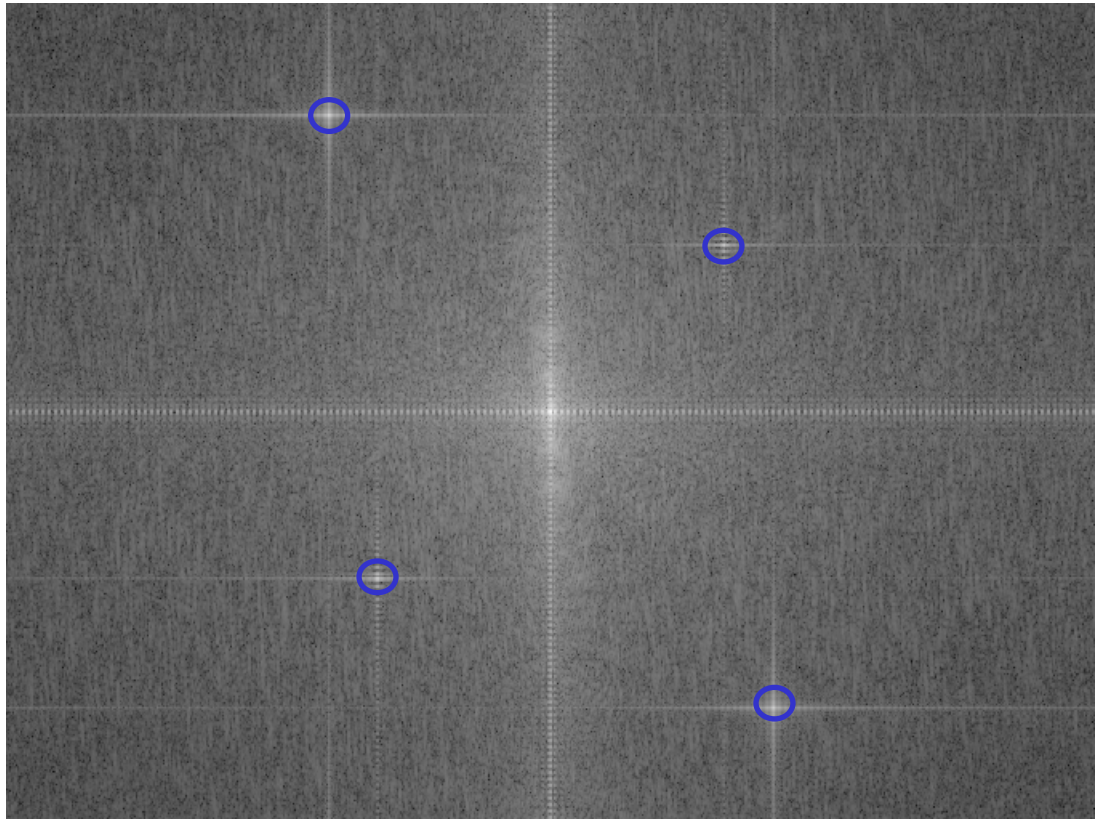
Step 2: identify the locations of the peaks in F

Padded image

Periodic noise reduction by selective filters

- Notch filters can efficiently remove the periodic noise

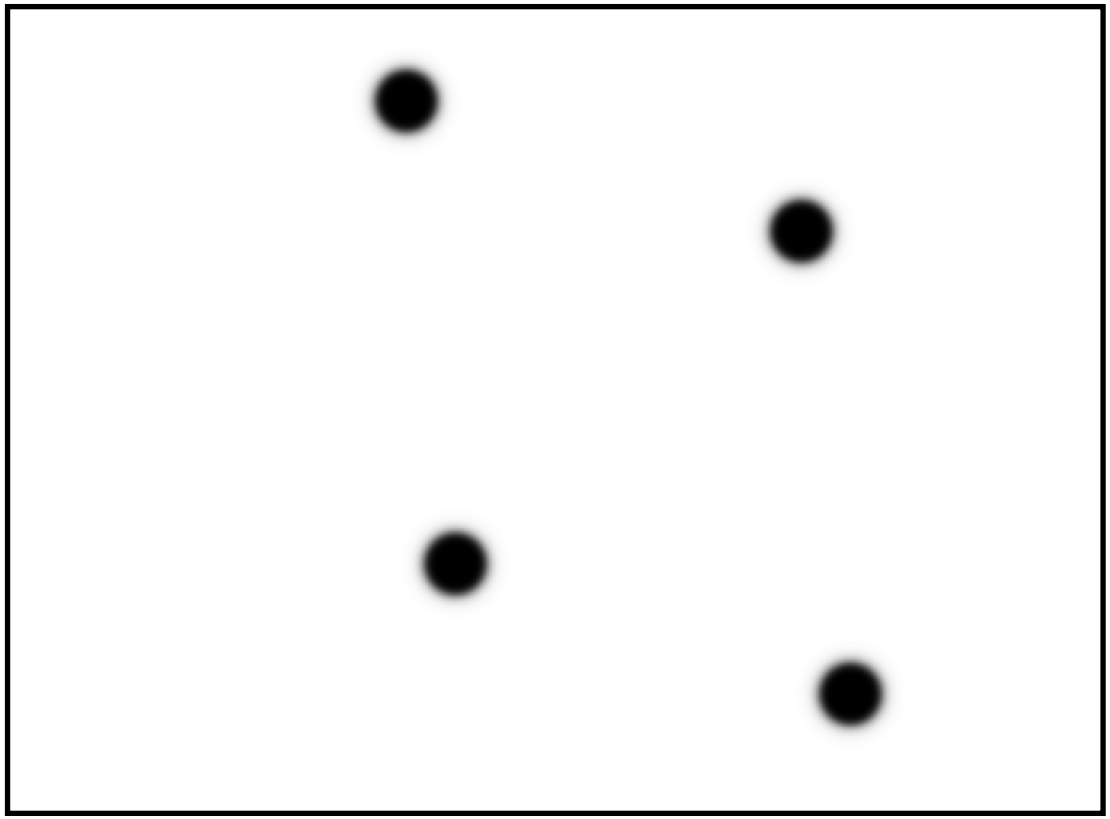
Step 2: identify the locations of the peaks in F



Fourier spectrum F

Periodic noise reduction by selective filters

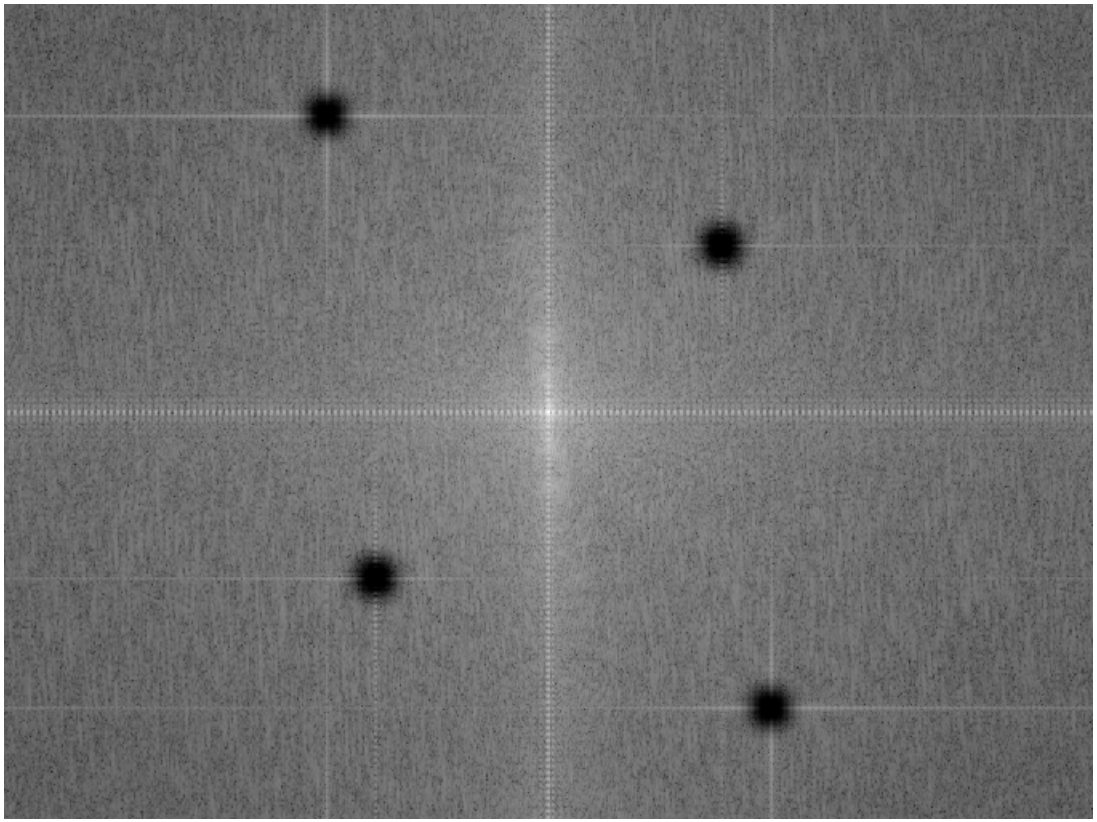
- Notch filters can efficiently remove the periodic noise
  Step 3: construct a notch reject filter H in Fourier domain, whose centers are at peaks



Notch filter H

Periodic noise reduction by selective filters

- Notch filters can efficiently remove the periodic noise
  Step 4: use H to filter F to get the result



HF in the Fourier domain

Periodic noise reduction by selective filters

- Notch filters can efficiently remove the periodic noise
  Step 4: use H to filter F to get the result

Final result in the spatial domain

Periodic noise reduction by selective filters

- Notch filters can efficiently remove the periodic noise
  Let's see the power of such a technology

Noisy image

After de-noising

Fast Fourier Transform

- The reason that Fourier based techniques have become so popular is the development of the Fast Fourier Transform (FFT) algorithm
- Allows the Fourier transform to be carried out in a reasonable amount of time
- Reduces the amount of time required to perform a Fourier transform by a factor of 100 – 600 times!

Fourier Domain Filtering & Spatial Domain Filtering

- Similar jobs can be done in the spatial and frequency domains
- Filtering in the spatial domain can be easier to understand
- Filtering in the frequency domain can be much faster – especially for large images

Summary

In this lecture we examined image filtering in the frequency domain

- Background
- From Fourier series to Fourier transform
- Properties of the Fourier transform
- Discrete Fourier transforms
- The basics of filtering in the frequency domain
- Image smoothing using frequency domain filters
- Image sharpening using frequency domain filters

Thanks for your attention