Lecture 4
Filtering in the Frequency Domain

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2016
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
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 Analytique 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...
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

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

\[
c_0 = \frac{a_0}{2}, \quad \frac{a_n - jb_n}{2} = c_n, \quad \frac{a_n + jb_n}{2} = d_n
\]

Then,

\[
f(t) = c_0 + \sum_{n=1}^{\infty} \left( c_n e^{jn\omega t} + d_n e^{-jn\omega t} \right) \quad (1)
\]
From Fourier Series to Fourier Transforms

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

\[ c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) (\cos n\omega t - j \sin n\omega t) \, dt = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-j n\omega t} \, dt \quad (2) \]

\[ d_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) (\cos n\omega t + j \sin n\omega t) \, dt = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{j n\omega t} \, dt \]

We can see that

\[ d_n = c_{-n} \]

Thus,

\[ \sum_{n=1}^{\infty} d_n e^{-j n\omega t} = \sum_{n=1}^{\infty} c_{-n} e^{-j n\omega t} = \sum_{n=-\infty}^{-1} c_n e^{j n\omega t} \quad (3) \]
From Fourier Series to Fourier Transforms

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

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

\[ = c_0 e^{j0\omega t} + \sum_{n=1}^{\infty} c_n e^{jn\omega t} + \sum_{n=-\infty}^{-1} c_n e^{jn\omega t} \] (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 $T$

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

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

According to Fourier series

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

Let $s_n = n \omega$

$$f_T(t) = \sum_{n=-\infty}^{+\infty} \left( \frac{1}{T} \int_{-T/2}^{T/2} f_T(t) e^{-j s_n t} dt \right) e^{j s_n t} = \frac{1}{T} \sum_{n=-\infty}^{+\infty} \left( \int_{-T/2}^{T/2} f_T(t) e^{-j s_n t} dt \right) e^{j s_n t}$$
From Fourier Series to Fourier Transforms

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

when \( T \to +\infty \)

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

\[ \Delta s = s_n - s_{n-1} = \omega = \frac{2\pi}{T} \rightarrow T = \frac{2\pi}{\Delta s} \]

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

\[ = \frac{1}{2\pi} \lim_{\Delta s \to 0} \sum_{n=-\infty}^{+\infty} \left( \int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(t) e^{-js_n t} \, dt \right) e^{js_n t} \Delta s \]
From Fourier Series to Fourier Transforms

\[ f(t) = \frac{1}{2\pi} \lim_{\Delta s \to 0} \sum_{n=-\infty}^{+\infty} \left( \int_{-T/2}^{T/2} f_T(t) e^{-jsn t} \, dt \right) e^{jst} \Delta s \]

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

\[ s_n \rightarrow s, \quad \Delta s \rightarrow ds, \quad \sum \rightarrow \int \]

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

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

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

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 \( \mu \) 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

\[
|F(\mu)| = (R^2(\mu) + I^2(\mu))^{1/2}, \phi(\mu) = \text{atan} 2 \frac{I(\mu)}{R(\mu)}
\]

$P(\mu) = |F(\mu)|^2 = R^2(\mu) + I^2(\mu)$ 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
# Symmetry Properties of the Fourier Transform

<table>
<thead>
<tr>
<th>Real $f(t)$</th>
<th>Fourier transform $F(\mu)$</th>
<th>Symmetry of $F(\mu)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>complex</td>
<td>conjugate symmetric $\left(F(-\mu) = F^*(\mu)\right)$</td>
</tr>
<tr>
<td>even $f(t) = f(-t)$</td>
<td>only real</td>
<td>even $\left(F(\mu) = F(-\mu)\right)$</td>
</tr>
<tr>
<td>odd $-f(t) = f(-t)$</td>
<td>only imaginary</td>
<td>odd $\left(-F(\mu) = F(-\mu)\right)$</td>
</tr>
</tbody>
</table>

*Proof?*
Impulse Function

• Impulse (Dirac) function
  • Considered as an infinitely high, infinitely thin spike at the origin, with total area one under the spike
  • It physically represents an idealized point mass or point charge

Paul Adrien Maurice Dirac
(Aug. 08, 1902—Oct. 20, 1984)
Impulse Function

- Impulse (Dirac) function

Definition 1:
\[
\delta(t) = \begin{cases} 
0, & t \neq 0 \\
\int_{-\infty}^{+\infty} \delta(t) dt = 1 
\end{cases}
\]

Definition 2:
\[
\int_{-\infty}^{+\infty} f(t) \delta(t) dt = f(0)
\]

Sift property:
\[
\int_{-\infty}^{+\infty} f(t) \delta(t - t_0) dt = f(t_0)
\]

Cannot be computed using normal integral methods!
Impulse Function

- Impulse (Dirac) function

\[ \delta(x) = \lim_{a \to 0} \frac{1}{a\sqrt\pi} e^{-\frac{x^2}{a^2}} \]

The Dirac delta function can be seen as the limit of the sequence of zero-centered normal distributions.
Impulse Function

- The Fourier transform of $\delta$-function

$$\mathcal{F}(\delta(t)) = 1$$

Proof:

$$F(\mu) = \int_{-\infty}^{+\infty} \delta(t)e^{-j2\pi \mu t} dt = e^{-j2\pi \mu t} \bigg|_{t=0} = e^{-j2\pi \mu 0} = 1$$

Similarly, we have

$$\mathcal{F}(\delta(t - t_0)) = e^{-j2\pi \mu t_0}$$
Fourier Transform of $f(t) = 1$

• The Fourier transform of the function 1 is

$$\mathcal{F}(1) = \delta(\mu)$$

Proof:

$$f(t) = \int_{-\infty}^{+\infty} \delta(\mu) \cdot e^{j2\pi\mu t} \, d\mu = e^{j2\pi\mu t} \bigg|_{\mu=0} = 1$$
Fourier Transform of $e^{jat}$

• The Fourier transform of the function $e^{jat}$ is

$$\mathcal{F} (e^{jat}) = \delta \left( \mu - \frac{a}{2\pi} \right)$$

Proof:

$$f(t) = \int_{-\infty}^{+\infty} \delta \left( \mu - \frac{a}{2\pi} \right) e^{j2\pi ut} \, d\mu = e^{j2\pi at} \bigg|_{\mu = \frac{a}{2\pi}} = e^{jat}$$
Fourier Transform of $\sin at$

- The Fourier transform of the function $\sin at$ is

$$\mathcal{F}(\sin at) = \frac{\delta\left(\mu - \frac{a}{2\pi}\right) - \delta\left(\mu + \frac{a}{2\pi}\right)}{2j}$$

Proof:

$$e^{jat} = \cos at + j \sin at, \quad e^{-jat} = \cos at - j \sin at$$

$$\sin at = \frac{e^{jat} - e^{-jat}}{2j}$$
Fourier Transform of $\sin at$

- The Fourier transform of the function $\sin at$ is

$$\mathcal{F}(\sin at) = \frac{\delta\left(\mu - \frac{a}{2\pi}\right) - \delta\left(\mu + \frac{a}{2\pi}\right)}{2j}$$

Proof:

$$F(\mu) = \int_{-\infty}^{+\infty} \frac{e^{jat} - e^{-jat}}{2j} e^{-j2\pi\mu t} dt$$

$$= \frac{1}{2j} \left( \int_{-\infty}^{+\infty} e^{jat} e^{-j2\pi\mu t} dt - \int_{-\infty}^{+\infty} e^{-jat} e^{-j2\pi\mu t} dt \right)$$

$$= \frac{\delta\left(\mu - \frac{a}{2\pi}\right) - \delta\left(\mu + \frac{a}{2\pi}\right)}{2j}$$
Fourier Transform of $\cos at$

- The Fourier transform of the function $\cos at$ is

$$\mathcal{F}(\cos at) = \frac{\delta\left(\mu - \frac{a}{2\pi}\right) + \delta\left(\mu + \frac{a}{2\pi}\right)}{2}$$

Can you work it out?
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
  \[ \mathcal{F}(f(t)) = F(\mu), \mathcal{F}(h(t)) = H(\mu) \]
  then \[ \mathcal{F}(f(t) * h(t)) = F(\mu) \cdot H(\mu) \]

Proof:
Convolution Theorem

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

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

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

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

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

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

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

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

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

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

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

= \( F(\mu) \cdot \)
Revisit Gaussian Filter

original

smoothed (5x5 Gaussian)

Why does this work?

smoothed – original

Lin ZHANG, SSE, 2016
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
Discrete Fourier Transform (DFT) in 1D Case

Given a discrete sequence with $M$ points

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

Regard it as a periodic signal, thus its basis frequency is $\frac{1}{M}$

For its frequency components, the frequencies are,

$$\frac{1}{M}, \frac{2}{M}, \ldots, \frac{M}{M} \equiv \frac{1}{M} (1, 2, \ldots, M)$$

Its DFT is computed as

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

Usually, we write it as,

$$F(u) = \sum_{x=0}^{M-1} f(x)e^{-j2\pi \frac{1}{M} ux}, u = 0, 1, 2, \ldots, M - 1$$
Discrete Fourier Transform (DFT) in 1D Case

Thus, \( f \)'s DFT also has \( M \) points

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

and

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

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

\[ F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)e^{-j2\pi(ux+vy)} \, dx \, dy \]

\[ f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v)e^{j2\pi(ux+vy)} \, du \, dv \]

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 \( u = 0,1,...,M-1; v = 0,1,...,N-1 \)

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

where \( x = 0,1,...,M-1; y = 0,1,...,N-1 \)
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
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.

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

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

Lin ZHANG, SSE, 2016
DFT Visualization Samples

• An example

Original image

Spectrum without using fftshift

Spectrum using fftshift

Lin ZHANG, SSE, 2016
DFT Visualization Samples

DFT

Any relationship?

Lin ZHANG, SSE, 2016
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
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
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

\begin{align*}
\mathbf{F}(u,v) &= \mathbf{F}(u,v) \\
\mathbf{G}(u,v) &= \mathbf{G}(u,v) \\
\mathbf{H}(u,v) &= \mathbf{H}(u,v)
\end{align*}

Source codes are available on our course website

Lin ZHANG, SSE, 2016
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

Lin ZHANG, SSE, 2016
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
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:

\[ H(u, v) = \begin{cases} 
1 & \text{if } D(u, v) \leq D_0 \\
0 & \text{if } D(u, v) > D_0 
\end{cases} \]

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

\[ D(u, v) = \left[u^2 + v^2\right]^{1/2} \]
Above we show an image, its 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 80

Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 230

Lin ZHANG, SSE, 2016
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:

$$H(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)}{D_0} \right]^{2n}}$$
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 80

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 230

Lin ZHANG, SSE, 2016
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 85

Result of filtering with Gaussian filter with cutoff radius 230

Result of filtering with Gaussian filter with cutoff radius 15
Low-pass Filtering Examples

A low pass Gaussian filter is used to connect broken text

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.
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
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,

\[ H_{HP} = 1 - H_{LP}(u, v) \]
The ideal high pass filter is given as:

\[ H(u, v) = \begin{cases} 
0 & \text{if } D(u, v) \leq D_0 \\
1 & \text{if } D(u, v) > D_0 
\end{cases} \]

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:

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

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 $D_0 = 30$

Results of Butterworth high pass filtering of order 2 with $D_0 = 80$
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 $D_0 = 15$
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