Image Enhancement II

BE 244 Lecture 2
Kio Kim

Spatial Filtering—continued

Filtering = Convolution

Convolution: \( f \ast g = \int f(x) g(x) dx \)

(Figure of cross-correlation)

Spatial Filtering—continued

• Famous linear kernel functions:
  - Gaussian
  - Sinc (‘ideal’)
  - Rectangle
  - Delta (impulse)
  - And Kalman filter, Wiener filter

• Nonlinear filters:
  - Median filter
  - Particle filter etc.

Fourier Transforms

• Basic idea:
  Any periodic function can be decomposed into sine and cosine functions.

Ideal Discrete Function

\* How much magnitude for each frequency?
\* How much phase shift for each frequency?

\rightarrow You can figure it out by Fourier transformation.

Fourier Transforms

Fourier Transform:

\[ F(u) = \int f(x) e^{-2\pi i u x} dx \]

Inverse Fourier Transform:

\[ f(x) = \int F(u) e^{2\pi i u x} du \]

FT using Euler’s Formula:

\[ F(u) = \int f(x) \cos(2\pi u x) dx \]

\[ = \frac{1}{2} \left[ \int f(x) e^{2\pi i u x} dx + \int f(x) e^{-2\pi i u x} dx \right] \]

\[ = \Re \{ F(u) \} = \mathcal{F} \{ f(x) \} \]

\[ |F(u)| = \sqrt{\Re^2 \{ F(u) \} + \Im^2 \{ F(u) \} } \]

\[ \angle \{ F(u) \} = \tan^{-1} \left( \frac{\Im \{ F(u) \}}{\Re \{ F(u) \} } \right) \]

\[ |F| = |F(u)| \]

\[ \angle F = \angle F(u) \]

\[ \mathcal{F} \{ F \} = F \]

\[ \mathcal{F} \{ F \} = F \]

\[ \mathcal{F} \{ F \} = F \]

\[ \mathcal{F} \{ F \} = F \]

\[ \mathcal{F} \{ F \} = F \]
Fourier Transform (2D)

1D Fourier Transform:
\[ F(u) = \int f(x) \exp(-j2\pi ux) \, dx \]

1D Inverse Fourier Transform:
\[ f(x) = \frac{1}{2\pi} \int F(u) \exp(j2\pi ux) \, du \]

2D Fourier Transform:
\[ F(u,v) = \int \int f(x,y) \exp(-j2\pi ux - j2\pi vy) \, dx \, dy \]

2D Inverse Fourier Transform:
\[ f(x,y) = \frac{1}{(2\pi)^2} \int \int F(u,v) \exp(j2\pi ux + j2\pi vy) \, du \, dv \]

Fourier Spectrum, Phase Angle, and Power Spectrum are all calculated in the same manner as the 1D case.

Discrete Fourier Transform

Sampled Continuous (Discrete) Function
\[ f(x) = f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \ldots, f(x_0 + (N-1)\Delta x) \]

Fourier Transform:
\[ F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-j2\pi ux/N) \]

Inverse Fourier Transform:
\[ f(x) = \sum_{u=0}^{N-1} F(u) \exp(j2\pi ux/N) \]

where \( \Delta x \) and \( \Delta u \) are sampling intervals in the spatial and frequency domains, respectively.
Discrete Fourier Transform (2D)

2D FT: \[ F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi uv/M} e^{-j2\pi vw/N} \]

2D IFT: \[ f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi uv/M} e^{j2\pi vw/N} \]

Properties

Translation:

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

Shifting Property

\[ f(x-u,y-v) = F(u,v) e^{j2\pi uv/M} e^{j2\pi vw/N} \]

Special Case: for \( u_0 = M/2 \), \( v_0 = N/2 \)

\[ f(x,y) \rightarrow 0 \rightarrow F(u,v) e^{-j2\pi uv/M} e^{-j2\pi vw/N} \]

Useful for centering FT or IFT at the origin

\[ f(x,M-x/2,\ldots,N-x/2) \rightarrow F(u,0) \rightarrow F(0,v) \]

Properties

Separability: 

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

for \( u = 0,1,\ldots,M-1 \)

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

1D row FT

\[ F(u,v) = \frac{1}{N} \sum_{v=0}^{N-1} f(x,v) e^{-j2\pi uv/M} e^{-j2\pi wv/N} \]

1D column FT
Properties

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

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

Conjugate Symmetry: \( F(u,v) = F^*(u,v) \)

Magnitude Symmetry: \( |F(u,v)| = |F^*(u,v)| \)

Rotation: \( x = u \cos \theta - v \sin \theta \), \( y = u \sin \theta + v \cos \theta \)

Polar coordinates: \( f(x,y) = f(u,v) \)

Distributivity: \( D[f(x,y) + g(x,y)] = D[f(x,y)] + D[g(x,y)] \)

Multiplication: \( \tilde{F}[f(x,y)g(x,y)] = \tilde{F}[f(x,y)] \tilde{F}[g(x,y)] \)

Scaling: \( \tilde{F}[af(x,y)] = \frac{1}{|a|} \tilde{F}[f(x,y)] \)

Average Value: \( F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp(j2\pi(u'x/M + v'y/N)) \)

The average value of \( f(x,y) \) is the value at the center of the frequency matrix.

Recall, if the FT is used to compute the convolution (as product of the FT of \( f(x,y) \) and \( g(x,y) \)), the assumption is that \( f \) and \( g \) are periodic. If the images are not padded to extend the FT computation window, wrap error will occur as shown.

Convolution in Fourier Domain

\[
\text{Convolution: } \hat{f}(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp(j2\pi(u'x/M + v'y/N))
\]

\[
\text{Convolution Theorem: } \hat{f}(u,v) = \hat{F}(u,v) \hat{G}(u,v)
\]

\[
\text{Convolution Theorem: } f(x,y) * g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x',y') g(x-x',y-y') dx' dy'
\]

\[
\text{Continuous 2D Convolution: } \hat{f}(u,v) * \hat{g}(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(u',v') \hat{g}(u-u',v-v') du' dv'
\]

\[
\text{Discrete 2D Convolution: } f(x,y) * g(x,y) = \sum_{x'=0}^{A-1} \sum_{y'=0}^{B-1} f(x',y') g(x-x',y-y')
\]

\[
\text{Convolution Theorem: } f(x,y) * g(x,y) = \hat{F}(u,v) \hat{G}(u,v)
\]

\[
\text{Convolution Theorem: } f(x,y) * g(x,y) = \hat{F}(u,v) \hat{G}(u,v)
\]
Convolution in Fourier Domain

Dirac Delta Function, \( \delta(x) \):
\[
\int_{-\infty}^{\infty} f(x) \delta(x-x_0) \, dx = f(x_0)
\]

Sampling is simply a convolution with an impulse function \( f(x-x_0) \)

\[
\tilde{f}(x) = \sum_{n=-\infty}^{\infty} f(n) e^{2\pi i nx} \]

Under periodic boundary,

pixel size \( \Delta x \) determines bandwidth \( B = \frac{1}{\Delta x}, \) Fourier

\# of pixels \( (A, B, \text{space}) \) = \# of samples \( (2D, \text{Fourier}) \)

Sampling Distortion

Band-limited Functions:
- Infinite in spatial domain
- Only defined on the frequency interval \([-W, W]\)
- Must be sampled at the Nyquist rate to avoid aliasing: \( \Delta x \leq \frac{1}{2W} \)

Sampling Distortion

Finite sampling with a window function:
- Window function that is finite in space (e.g. rect function) has infinite frequency components
- Infinite window function, \( h(n) \), and finite sampled function, \( s[n] \), introduce distortion
- Limits full recovery of original function, \( f(x) \), from a finite number of samples
- Complete recovery is only possible if \( f(x) \) is band-limited and periodic with a period \( X \) where

Spatial Domain: \( X = X \)
Frequency Domain: \( \Delta f = \frac{1}{X} \)

Properties

2D Function (Image)

FT of Image

Shifted FT of Image
Frequency Domain Filters

- Frequency filter $H(u,v)$ suppresses certain frequency components in an image
  - Low-pass filters smooth images by suppressing high frequency components (rapidly changing intensities)
  - High-pass filters highlight edges by suppressing low frequency components (near-constant intensities)
- Spatial filters are applied to the image with a 2D convolution. By the convolution theorem
  \[ f(x,y) \ast h(x,y) = F(u,v)H(u,v) \]
  where the prime indicates that the images are padded appropriately
- **PRO:** Filtering in the frequency domain is often more intuitive. Faster if your kernel image is big ($O(N^2)$ vs $O(N \log N)$ for FFT)
- **CON:** Slower if your kernel image is small
- Decide on the filter characteristics in the frequency domain but perform the filtering in the spatial domain

---

Correspondences

- Famous linear kernel functions:
  - Gaussian
  - Sinc (‘ideal’)
  - Rectangle
  - Delta (impulse)

---

Low Pass Filters

Find the appropriate $H(u,v)$ to suppress high frequency components in $F(u,v)$ and generate $G(u,v)$ with smoother intensities and reduced noise

\[ G(u,v) = F(u,v)H(u,v) \]

**Ideal low-pass filter**

\[ H(u,v) = \begin{cases} 1, & |D_u| \leq D_c \\ 0, & |D_u| > D_c \end{cases} \]

where $D_c$ is a specified cutoff frequency

---

‘Ideal’ Low Pass Filter

**Figure 1:** 1D low pass filter of 100x100 pixels of zero to 255 gray scale with cut-off frequency $0.5$. $D_c = 0.5$.

Recall reciprocal behavior with $h(x,y)$. Narrowing in frequency is widening in space (e.g., larger smoothing kernel). Both result in increased blur.
'Ideal' LPF

Source of ringing in images filter with LPF

Butterworth LPF

Butterworth LPF of order $n$

$H(u, v) = \frac{1}{1 + (\frac{R}{D})^n}$

Note how higher orders approximate LPF:

- Smoother transition band
- Typically define cutoff ($D_c$) as the value when the function is a certain fraction of the maximum value (eg $D_c$ is defined at $H(u, v) = 0.5$ in the figure)

Butterworth LPF

Effect of 2nd order BLPF with same cutoffs shown for ILPF

No ringing!

Butterworth LPF

Gaussian LPF

$H(u, v) = \exp \left( -\frac{D(u, v)^2}{2\sigma^2} \right)$

Note how lower values of $D_c$ approximate LPF:

How do you expect these filters to compare with BLPF?

$D_c$ is the standard deviation of the Gaussian distribution, therefore, it is defined at 67% of max($H(u, v)$)
High Pass Filters

Find the appropriate \( H(u,v) \) to suppress low frequency components in \( F(u,v) \) and generate \( G(u,v) \) with sharper intensity transitions (edges)

\[
G(u,v) = F(u,v) \cdot H(u,v)
\]

\[ H(u,v) = 1 - \frac{X^2 + Y^2}{D_0^2} \]

A common method for generating a HPF is to take the inverse of a LPF

\[ H_{\text{LPF}}(u,v) = \frac{X^2 + Y^2}{D_0^2} \]

\[ H_{\text{Ideal HPF}}(u,v) = \begin{cases} 
1 & \text{if } \sqrt{X^2 + Y^2} < D_1 \\
0 & \text{if } \sqrt{X^2 + Y^2} > D_2 
\end{cases} \]

Ideal HPF

Butterworth HPF

\[
H_{\text{Butterworth}}(u,v) = \frac{1}{\left(1 + \left(\frac{\sqrt{X^2 + Y^2}}{D_0}\right)^n\right)}
\]

Gaussian HPF

\[
H_{\text{Gaussian}}(u,v) = \exp \left( -\frac{X^2 + Y^2}{2\sigma^2} \right)
\]

Minimal ringing and cleaner edges at small \( D_0 \) than BHPF