Image Enhancement II

BE 244 Lecture 2

Kio Kim

Image Enhancement II

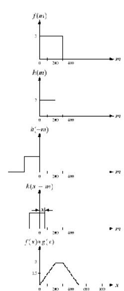
- Spatial Filter (continued)
- Fourier Transforms (1D and 2D)
- Discrete Fourier Transform (1D and 2D)
- Properties of Fourier Transform
- Sampling
- Low-pass and High-pass Filters

Spatial Filtering—continued

Filtering = Convolution

Convolution: $f * g(x) = \int_{-\infty}^{\infty} f(\alpha)\, g(x-\alpha)\, d\alpha$

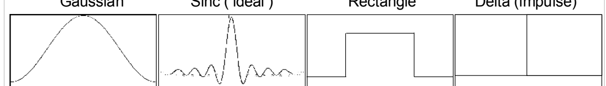
(Opposite of cross-correlation)



Spatial Filtering—continued

- Famous linear kernel functions:

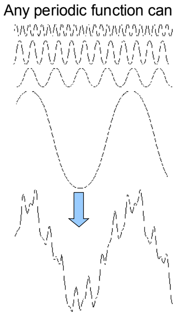
Gaussian | Sinc ('ideal') | Rectangle | Delta (impulse)

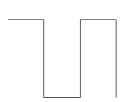


  - And Kalman filter, Wiener filter
- Nonlinear filters:
  - Median filter
  - Particle filter etc.

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?

-->You can figure it out
by Fourier transformation.

Fourier Transforms

Fourier Transform: $F(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux]\, dx$

Inverse Fourier Transform: $f(x) = \int_{-\infty}^{\infty} F(u) \exp[j2\pi ux]\, du$

FT using Euler's Formula:

$$F(u) = \int_{-\infty}^{\infty} f(x)[\cos 2\pi ux - j \sin 2\pi ux]\, dx$$
$$= \int_{-\infty}^{\infty} f(x)[\cos 2\pi ux]\, dx - j \int_{-\infty}^{\infty} f(x)[\sin 2\pi ux]\, dx$$
$$= R(u) - jI(u)$$

$$F(u) = |F(u)| \exp[j\phi(u)]$$

$|F(u)| = \left[R^2(u) + I^2(u)\right]^{\frac{1}{2}}$ — Fourier Spectrum (magnitude)

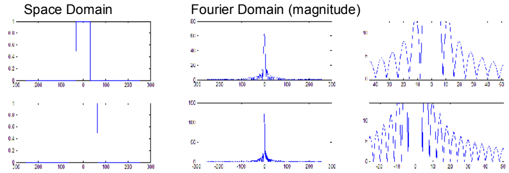
$|\phi(u)| = \tan^{-1} \left|\frac{I(u)}{R(u)}\right|$ — Phase Angle (phase)

$|P(u)| = |F(u)|^2$ — Spectral Density (power spectrum)

Fourier Transform (1D Example)

$$f(x) = \begin{cases} A & \text{for } -32 \le x < 32 \\ 0 & \text{otherwise} \end{cases} \qquad F(u) = \begin{cases} A & \text{for } u = 0 \\ \dfrac{\sin u}{u} & \text{otherwise} \end{cases}$$

Space Domain   Fourier Domain (magnitude)



Fourier Transform (2D)

1D Fourier Transform: $F(u) = \int_{-\infty}^{\infty} f(x)\exp[-j2\pi ux]\,dx$

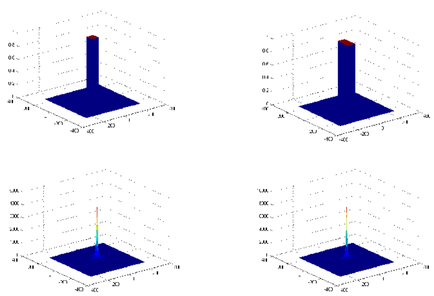
1D Inverse Fourier Transform: $f(x) = \int_{-\infty}^{\infty} F(u)\exp[j2\pi ux]\,du$

2D Fourier Transform: $F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp[-j2\pi(ux+vy)]\,dx\,dy$

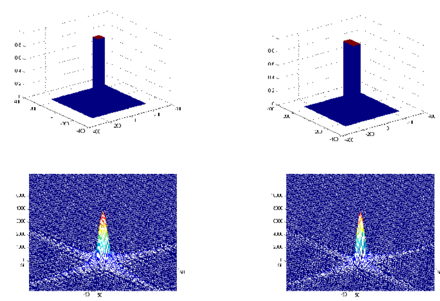
2D Inverse Fourier Transform: $f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\exp[j2\pi(ux+vy)]\,du\,dv$

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

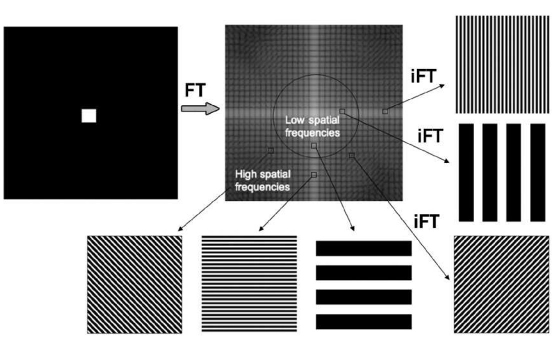
Fourier Transform (2D Example)



Fourier Transform (2D Example)



Fourier Transform (2D Example)



http://www.ajronline.org/cgi/content/figsonly/190/5/1396

Discrete Fourier Transform

Sampled Continuous (Discrete) Function: $f(x) = \{ f(x_0), f(x_o + \Delta x), f(x_o + 2\Delta x), \ldots, f(x_o + (N-1)\Delta x) \}$

$f(x) = f(x_o + x\Delta x)$ wh $x = 0, 1, 2, \ldots, N-1$

Fourier Transform: $F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j\frac{2\pi ux}{N}]$ for $u = 0, 1, 2, \ldots, N-1$

Inverse Fourier Transform: $f(x) = \sum_{x=0}^{N-1} F(u)\exp[j\frac{2\pi ux}{N}]$ for $x = 0, 1, 2, \ldots, N-1$

(continuous) Fourier Transform: $F(u) = \int_{-\infty}^{\infty} f(x)\exp[-j2\pi ux]\,dx$

(continuous) Inverse Fourier Transform: $f(x) = \int_{-\infty}^{\infty} F(u)\exp[j2\pi ux]\,du$

$N\Delta u = \frac{1}{\Delta x}$ wh Δx and Δu are sampling intervals in the spatial and frequency domains, resp.

Discrete Fourier Transform (2D)

2D FT: $$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$ for $u = 0,1,2,\ldots,M-1$, $v = 0,1,2,\ldots,N-1$

2D IFT: $$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\exp\left[j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$ for $x = 0,1,2,\ldots,M-1$, $y = 0,1,2,\ldots,N-1$

$$M\Delta u=\frac{1}{\Delta x} \qquad N\Delta v=\frac{1}{\Delta y}$$

The sampling intervals have the same relationship as in the 1D case

2D Fourier Transform: $$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp\left[-j2\pi(ux+vy)\right]dxdy$$

2D Inverse Fourier Transform: $$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\exp\left[j2\pi(ux+vy)\right]dudv$$

13

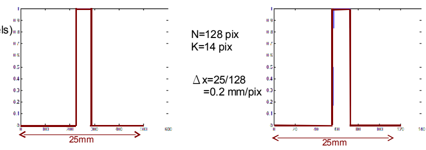
Discrete Fourier Transform

N=512 samples (pixels)
K=64 pixels

Δx = 25/512
= 0.05 mm/pixel

N=128 pix
K=14 pix

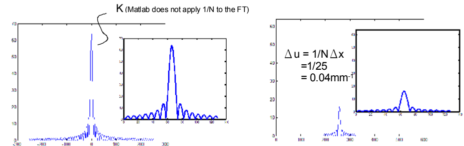
Δx=25/128
=0.2 mm/pix



K (Matlab does not apply 1/N to the FT)

$\Delta u = 1/N\Delta x$
= 1/25
= 0.04 mm^{-1}

$\Delta u = 1/N\Delta x$
=1/25
= 0.04mm^{-1}



Subsampling changes peak height (DC value) of FT (Matlab only) but the rate of zero crossings is the same for both.
Functions are the same, but represented by a different number of frequency points => loss of high frequency information.

14

Properties

Translation:

$$F(u-u_0,v-v_0)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)\right]\exp\left[-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$

$$f(x,y)\exp\left[j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)\right]\Leftrightarrow F(u-u_0,v-v_0)$$ Shifting Property

$$f(x-x_0,y-y_0)\Leftrightarrow F(u,v)\exp\left[-j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)\right]$$

Special Case: for $u_0 = M/2$, $v_0 = N/2$ $$\exp\left[j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)\right]=(-1)^{x+y}$$

Useful for centering FT or IFT at the origin

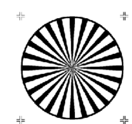
$$f(x,y)(-1)^{x+y}\Leftrightarrow F\left(u-\frac{M}{2},v-\frac{N}{2}\right)$$

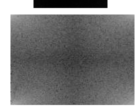
$$f\left(x-\frac{M}{2},y-\frac{N}{2}\right)\Leftrightarrow F(u,v)(-1)^{x+y}$$

15

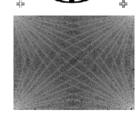
Properties

2D Function (Image)





FT of Image



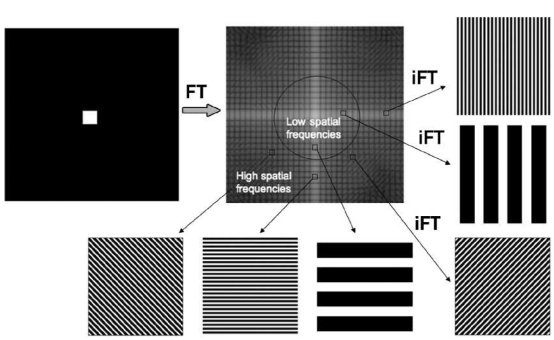
Shifted FT of Image

16

(back of 'moon blade' shown in FT)

(Spokes in spatial domain and wrapping (aliasing; Moire pattern) shown.)

Properties



http://www.ajronline.org/cgi/content/figsonly/190/5/1396

Properties

Separability: $$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$ for $u = 0,1,2,\ldots,M-1$, $v = 0,1,2,\ldots,N-1$

$$=\frac{1}{M}\sum_{x=0}^{M-1}\exp\left[-\frac{j2\pi ux}{M}\right]\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp\left[-\frac{j2\pi vy}{N}\right]$$

$$=\frac{1}{M}\sum_{x=0}^{M-1}F(x,v)\exp\left[-\frac{j2\pi ux}{M}\right]$$ 1D row FT

where $$F(x,v)=\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp\left[-\frac{j2\pi vy}{N}\right]$$ 1D column FT

18

Properties

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

$$F(u-M,v-N)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)\exp j2\pi\left(\frac{Mx}{M}+\frac{Ny}{N}\right)\right]\exp\left[-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$

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

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

Magnitude Symmetry: $|F(u,v)|=|F(-u,-v)|$

Properties

Rotation: $x=r\cos\theta$ $\quad u=\omega\cos\phi$ $\quad$ polar coordinates $\quad f(r,\theta-\theta_0)\Leftrightarrow F(\omega,\phi-\theta_0)$

$y=r\sin\theta$ $\quad v=\omega\sin\phi$

Distributivity: $\Im[f_1(x,y)+f_2(x,y)]=\Im[f_1(x,y)]+[f_2(x,y)]$ Addition YES!

$\Im[f_1(x,y)\cdot f_2(x,y)]\neq\Im[f_1(x,y)]\cdot[f_2(x,y)]$ Multiplication NO!

Scaling: $af(x,y)\Leftrightarrow aF(u,v)$ $\quad f(ax,by)\Leftrightarrow\frac{1}{|ab|}F\left(\frac{u}{a},\frac{v}{b}\right)$

Average Value: $F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$

for *u = v = 0 (origin)* $F(0,0)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$

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

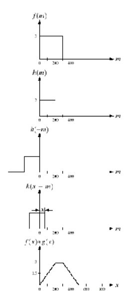
(Think what 'zero frequency' means---it's flat!)

Convolution in Fourier Domain

Convolution: $f(x)*g(x)=\int_{-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha$

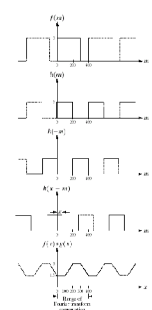
Convolution Theorem: $f(x)*g(x)\Leftrightarrow F(u)G(u)$

$f(x)g(x)\Leftrightarrow F(u)*G(u)$



Convolution in Fourier Domain

Recall, if the FT is used to compute the convolution (ie 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

Discrete Functions: $f(x)=|f(0),f(1),f(2),\ldots,f(A-1)|$

$g(x)=|g(0),g(1),g(2),\ldots,g(B-1)|$

To perform a convolution, we assume f(x) and g(x) are periodic with a period $P\geq A+B-1$
The resulting convolution will have the same period (P) and the individual periods will not overlap, ie no aliasing

$f'(x)=\begin{cases}f(x); & 0\leq x\leq A-1\\ 0; & A\leq x\leq P-1\end{cases}$ $\quad g'(x)=\begin{cases}g(x); & 0\leq x\leq B-1\\ 0; & B\leq x\leq P-1\end{cases}$ $\quad$ Zero-padded functions

$f'(x)*g'(x)=\frac{1}{P}\sum_{p=0}^{P-1} f'(p)g'(x-p)$ for *x = 0,1,2,…,P-1*

Convolution Theorem: $f'(x)*g'(x)\Leftrightarrow F(u)G(u)$

$f'(x)g'(x)\Leftrightarrow F(u)*G(u)$

Convolution in Fourier Domain

Continuous 2D Convolution: $f(x,y)*g(x,y)=\int_{-\infty}^{\infty}\int f(\alpha,\beta)g(x-\alpha,y-\beta)d\alpha d\beta$

Discrete 2D Convolution:

$f'(x,y)=\begin{cases}f(x,y); & 0\leq x\leq A\text{-}1 \;\&\; 0\leq y\leq B\text{-}1\\ 0; & A\leq x\leq P \text{ or } B\leq y\leq Q\end{cases}$

$g'(x,y)=\begin{cases}g(x,y); & 0\leq x\leq C\text{-}1 \;\&\; 0\leq y\leq D\text{-}1\\ 0; & C\leq x\leq P \text{ or } D\leq y\leq Q\end{cases}$

$f'(x,y)*g'(x,y)=\frac{1}{PQ}\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} f'(p,q)g'(x-p,y-q)$ for *x = 0,1,2,…,P-1*, *y = 0,1,2,…,Q-1*

Convolution Theorem: $f'(x,y)*g'(x,y)\Leftrightarrow F(u,v)G(u,v)$

$f'(x,y)g'(x,y)\Leftrightarrow F(u,v)*G(u,v)$

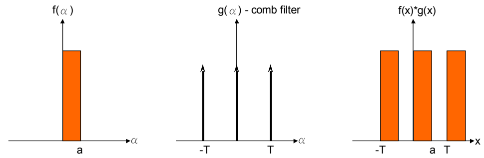
Convolution in Fourier Domain

Dirac Delta Function, δ (x): $\int_{-\infty}^{\infty}\delta(x-x_0)dx = \int_{-x_0}^{x_0^+}\delta(x-x_0)dx = 1$

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

$$\int_{-\infty}^{\infty} f(x)\,\delta(x-x_0)\,dx = f(x_0)$$

Sifting Property

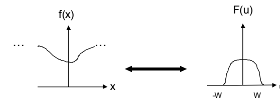


Under periodic boundary,
pixel size (Δx, space) determines bandwidth (N Δu=1/Δx, Fourier)
# of pixels (A,B, space) = # of samples (C,D, Fourier)

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: $\Delta x \leq 1/(2W)$



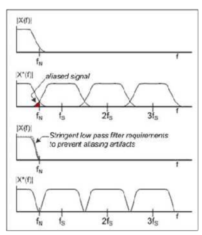
Sampling Distortion

Finite sampling with a window function:

- Window function that is finite in space (eg rect function) has infinite frequency components
- Infinite window function, H(u), and finite sampled function, s(u)*F(u), introduces 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: $N\Delta x = X$

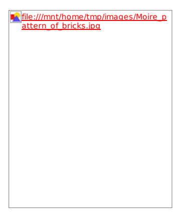
Frequency Domain: $N\Delta u = \frac{1}{\Delta x_0} \longrightarrow \Delta u = \frac{1}{N\Delta x_0}$



Narrow down the bandwidth (reduce sample #) too much --> Aliasing!

Sampling Distortion

- How can we avoid aliasing?
  - By suppressing high-frequency structures. (or, by smoothing the image)-->Gaussian filtering

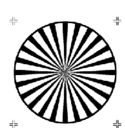


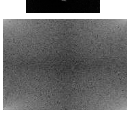
Moire Pattern

http://upload.wikimedia.org/wikipedia/commons/3/31/Moire_pattern_of_bricks.jpg

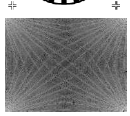
Properties

2D Function (Image)

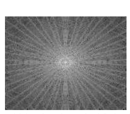




FT of Image

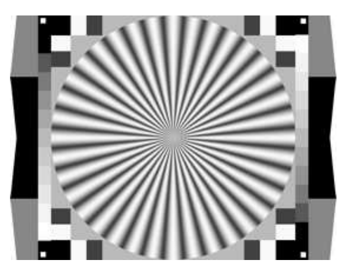


Shifted FT of Image



(back of 'moon blade' shown in FT)

(Spokes in spatial domain and wrapping (aliasing: Moire pattern) shown.)



'Siemens Star' http://www.imatest.com/docs/testcharts.html

(Or, http://www.youtube.com/watch?v=Dyisg9-Mwjw to see how sampling can distort what you see.)

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

$$f'(x,y) * h'(x,y) \Leftrightarrow F(u,v)H(u,v)$$ where the prime indicates that the images are padded appropriately

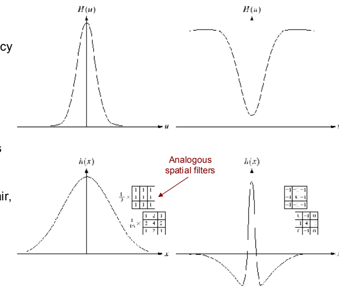
- PRO: Filtering in the frequency domain is often more intuitive,
  Faster if your kernel image is big (O(N^2) vs O($N \log N$) for FFT)
- CON: Slower if your kernel image is small
- Decide on the filter characteristics in the frequency domain but perform the filtering in the spatial domain

Frequency Domain Filters

Gaussian-based filters

$$H(u) = A\exp\left(-\frac{u^2}{2\sigma^2}\right) \longleftrightarrow h(x) = \sqrt{2\pi}\sigma A\exp\left(-2u^2\sigma^2x^2\right)$$

- Gaussian filters are real in both the frequency and spatial domains, ie computation is simpler than other filters
- We use Gaussian functions often in analyzing biological data because it best describes the normal distribution of observations that scientists make during an experiment. Because of this, the properties of the Gaussian are more intuitive to us (as scientists) than other functions.
- Because these two functions form an FT pair, their behavior is reciprocal



Frequency Domain Filters

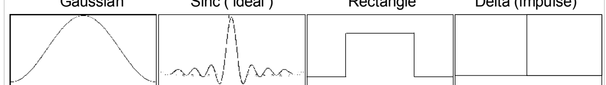
Basic Matlab steps in DFT filtering to image f

1. Compute the transform F(u,v) : fft2(f)
2. Generate a filter function H(u,v) of size A+C-1, B+D-1 using fspecial and freqz2. Use fftshift, if necessary, to be consistent with position of F(u,v)
   (*. freqz2 is not available in OCTAVE.)
3. Multiply the transform by the filter G(u,v) = H(u,v)F(u,v): G = H.*F
4. Obtain the real part of the inverse FT of G(u,v): g = real(ifft2(G))
5. Crop the top left rectangle to the original size of f: g = g(1:size(f,1), 1:size(f,2))

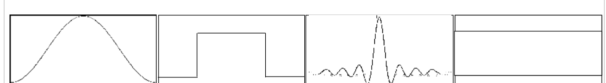
Correspondences

- Famous linear kernel functions:

Gaussian | Sinc ('ideal') | Rectangle | Delta (impulse)



- Correspondences in Fourier domain:



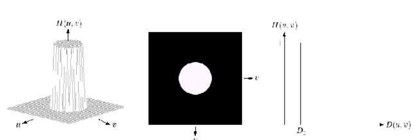
(Imagine squeezing sinc to 0)

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/or reduced noise

$$G(u,v) = F(u,v)H(u,v)$$

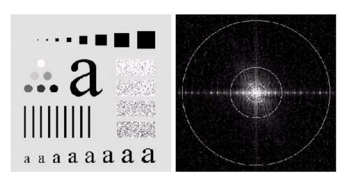
Ideal low-pass filter $H(u,v) = \begin{cases} 1; & D(u,v) \le D_0 \\ 0; & D(u,v) > D_0 \end{cases}$ where D_o is a specified cutoff frequency



a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

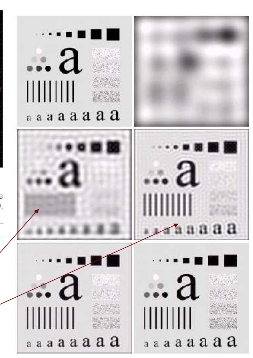
'Ideal' Low Pass Filter



a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Recall reciprocal behavior with h(x,y).
Narrowing in frequency is widening in space (eg larger smoothing kernel).
Both result in increased blur.



ILPFs are not ideal!
Ringing occurs in the spatial domain due to truncation artifacts

‘Ideal’ LPF

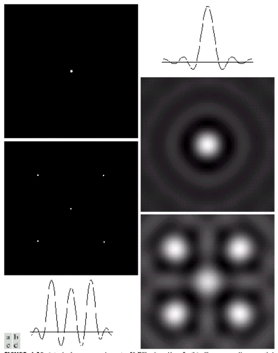


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Source of ringing in images filter with ILPF

Butterworth LPF

Butterworth LPF of order *n*

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

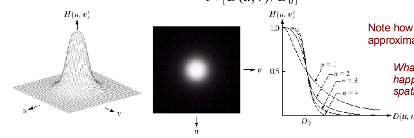


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Note how higher orders approximate ILPF.

What do you think is happening in the spatial domain?

- Smoother transition band
- Typically define cutoff (D_0) as the value when the function is a certain fraction of the maximum value (eg D_0 is defined at H(u,v) = 0.5 in the figure)

Butterworth LPF

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

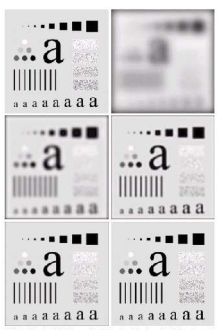


FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

No ringing!

Butterworth LPF

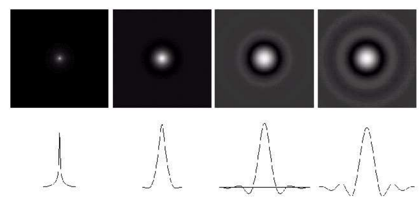


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Gaussian LPF

Gaussian LPF

$$H(u,v) = \exp\left[-\frac{D(u,v)^2}{2\sigma^2}\right] = \exp\left[-\frac{D(u,v)^2}{2D_0^2}\right]$$

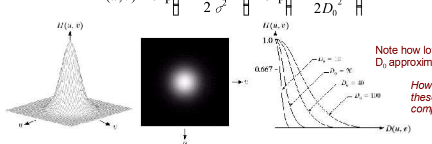


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Note how lower values of D_0 approximate ILPF.

How do you expect these filters to compare with BLPF?

D_0 is the standard deviation of the Gaussian distribution, therefore, it is defined at 67% of maxH(u,v)

Gaussian LPF

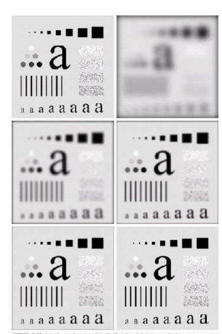


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

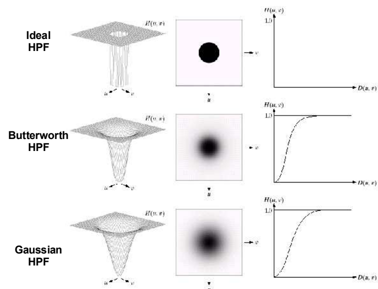
High Pass Filters

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)

$$G(u,v) = F(u,v)H(u,v)$$

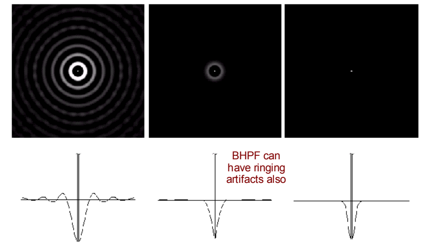
A common method for generating a HPF is to take the inverse of a LPF

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



HPF - Spatial Representations

IHPF still has ringing artifacts



a b c

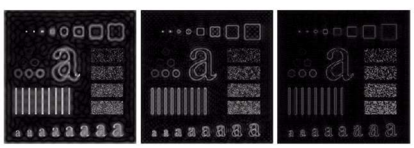
FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

Ideal HPF

Ideal HPF

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

Ringing at small D_0



a b c

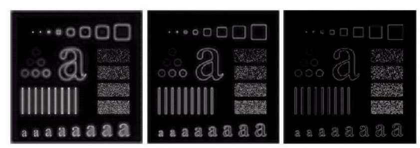
FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

Small spot size in frequency means large spot size in space

Butterworth HPF

Butterworth HPF of order *n*

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



a b c

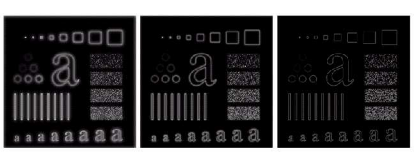
FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

Smoother behavior (less ringing) than IHPF but edges are still blurred at smaller D_0

Gaussian HPF

Gaussian HPF

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



a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Minimal ringing and cleaner edges at small D_0 than BHPF