Filtering in the Fourier Domain

Ross Whitaker
SCI Institute, School of Computing
University of Utah

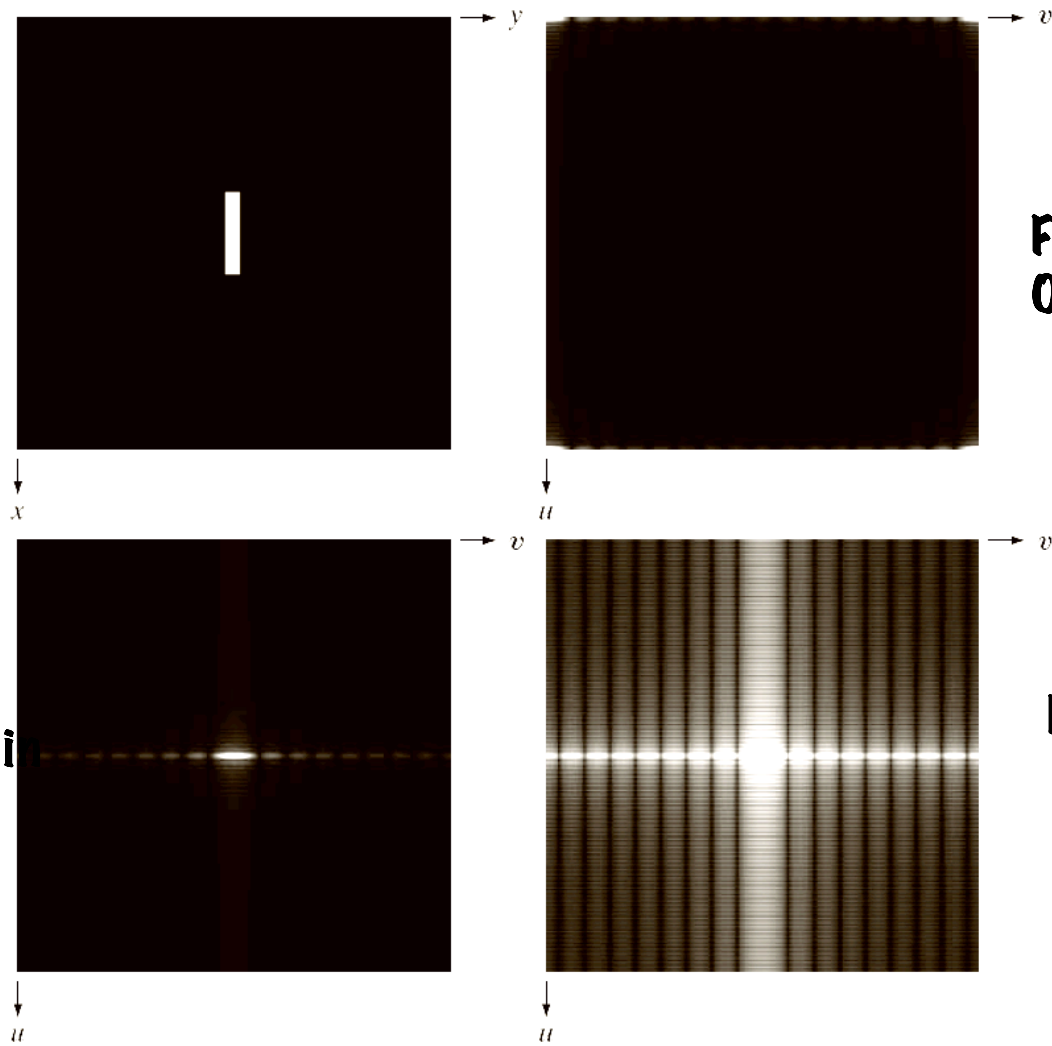
Fourier Filtering

- Low-pass filtering
- High-pass filtering
- Band-pass filtering
- Sampling and aliasing
- Tomography
- Optimal filtering and match filters

Some Identities to Remember

Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
Rectangle	$\mathrm{rect}[a, b] \Leftrightarrow ab\frac{\sin(\pi ua)}{(\pi ua)}\frac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $j\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)\right]$
Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)\right]$
Gaussian	$A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

Fourier Spectrum



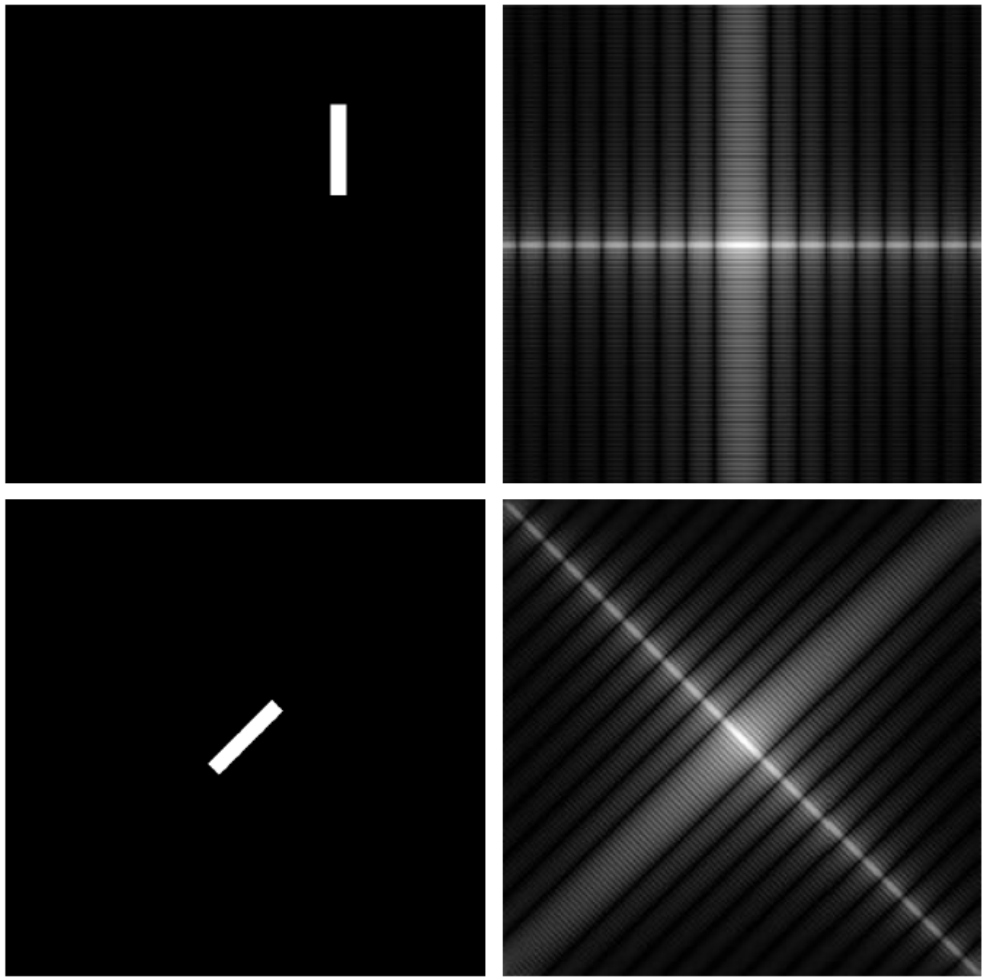
Image

Fourier spectrum
Origin in corners

Retiled with origin
In center

Log of spectrum

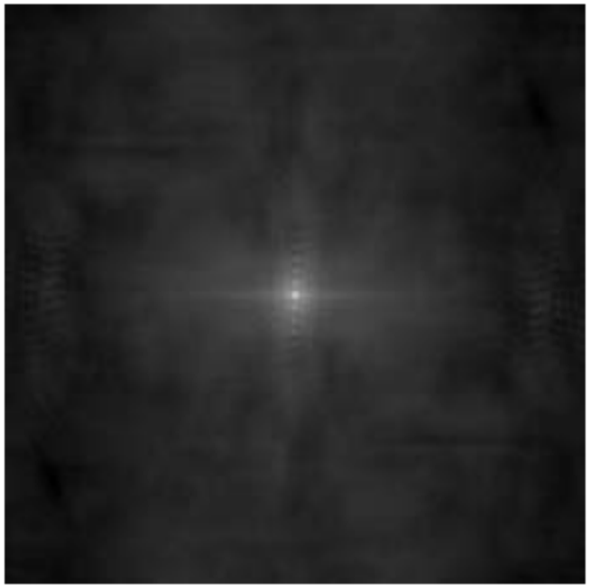
Fourier Spectrum-Rotation



Phase vs Spectrum

Image

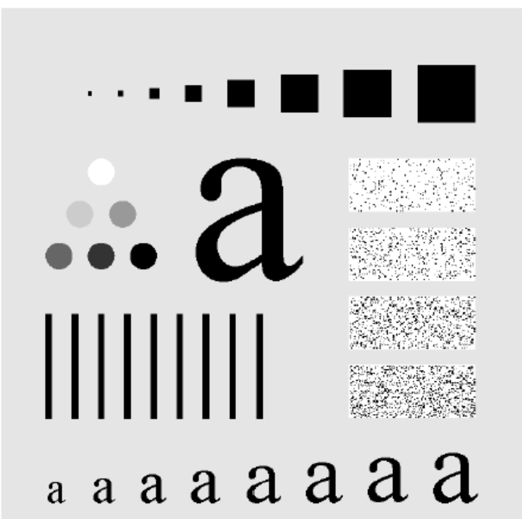
Reconstruction from <u>phase map</u>

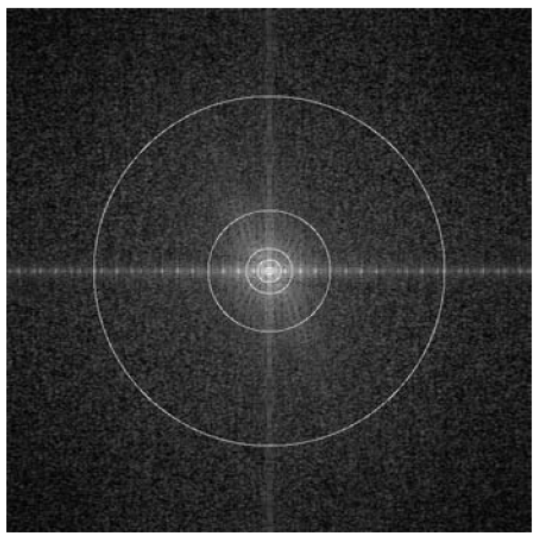


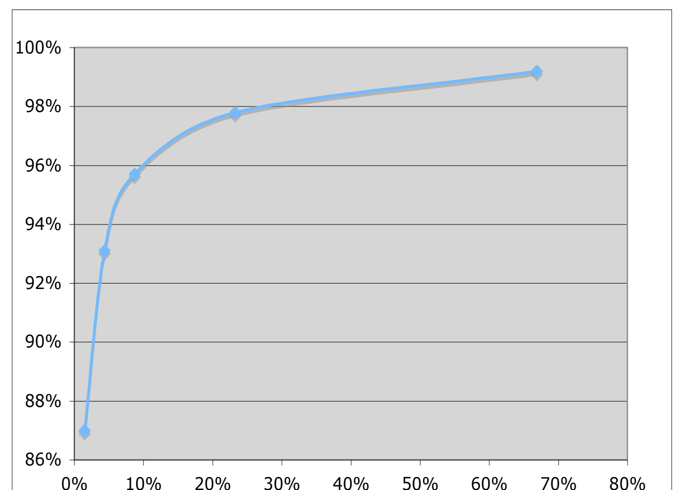
Reconstruction from <u>spectrum</u>

Low-Pass Filter

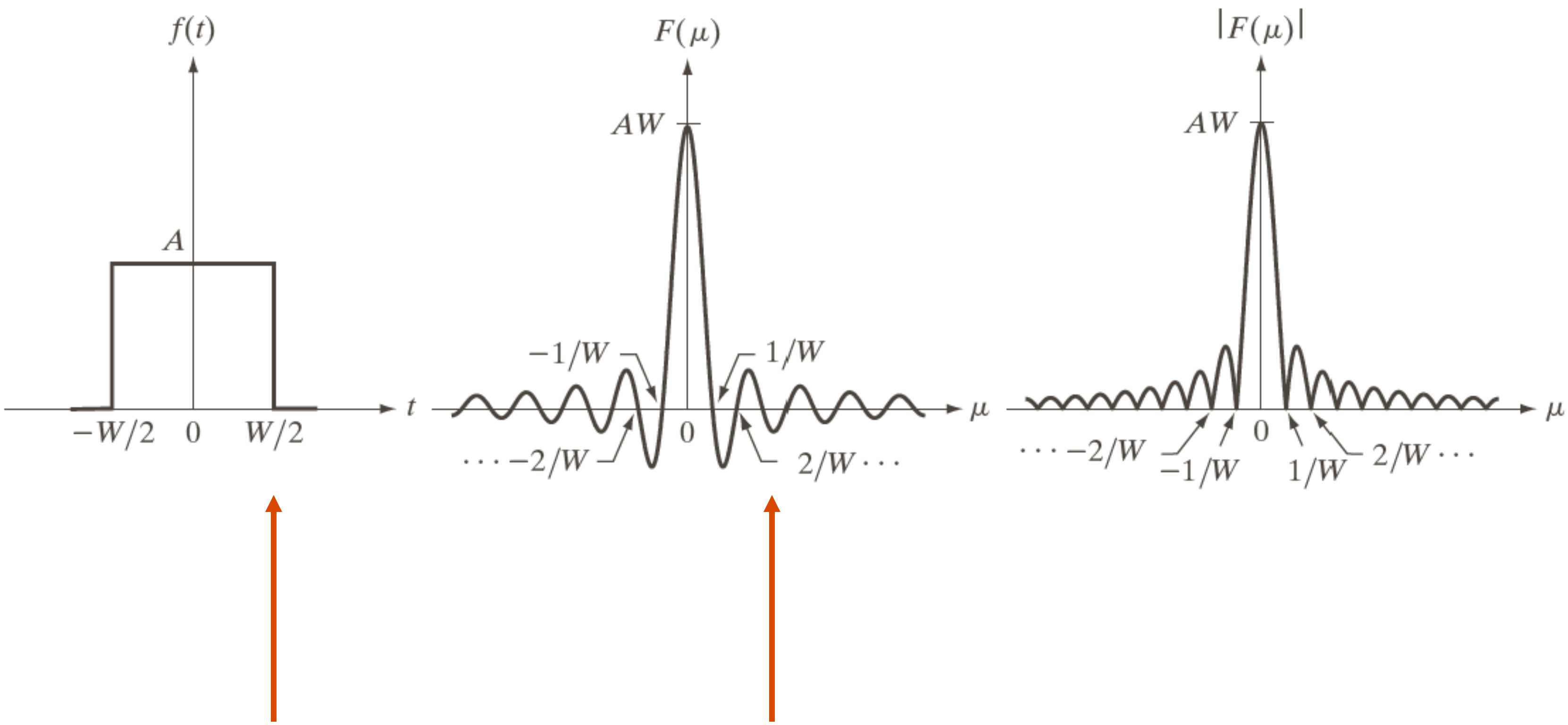
- Reduce/eliminate high frequencies
- Applications
  - Noise reduction
    - uncorrelated noise is broad band
    - Images have sprectrum that focus on low







Ideal LP Filter - Box, Rect

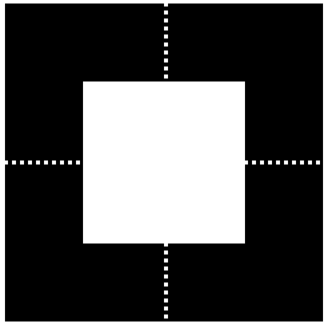


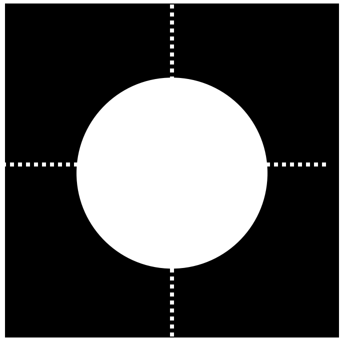
Cutoff freq

Ringing - Gibbs phenomenon

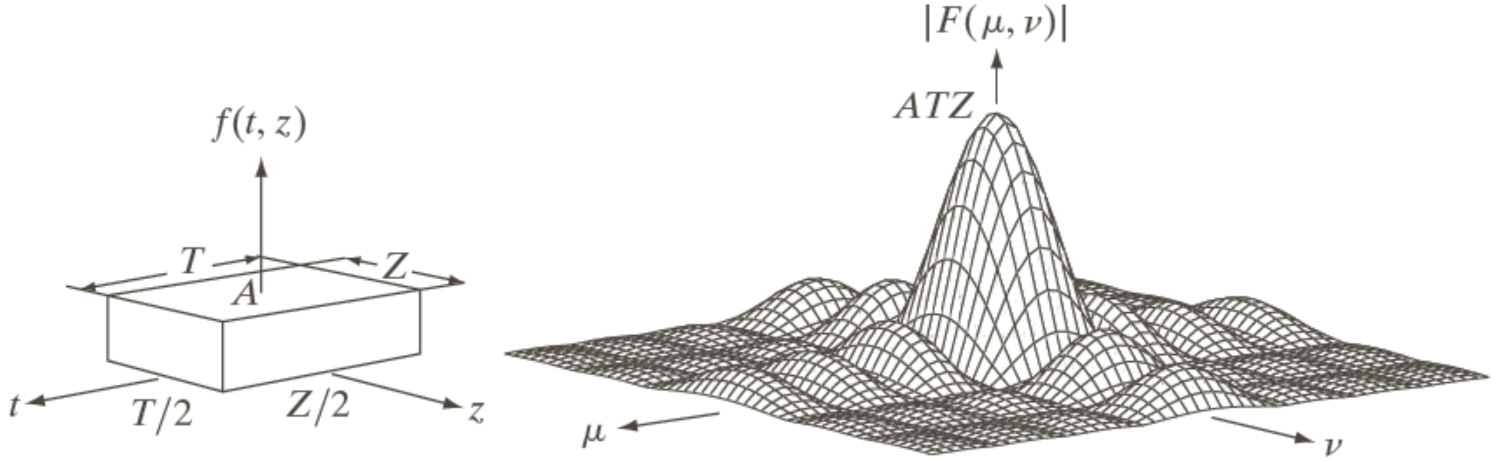
Extending Filters to 2D (or higher)

- Two options
  - Separable
    - H(s) -> H(u)H(v)
    - Easy, analysis
  - Rotate
    - H(s) -> $H((u^2 + v^2)^{1/2})$
    - Rotationally invariant





Ideal LP Filter - Box, Rect



Ideal Low-Pass
Rectangle With Cutoff of 2/3

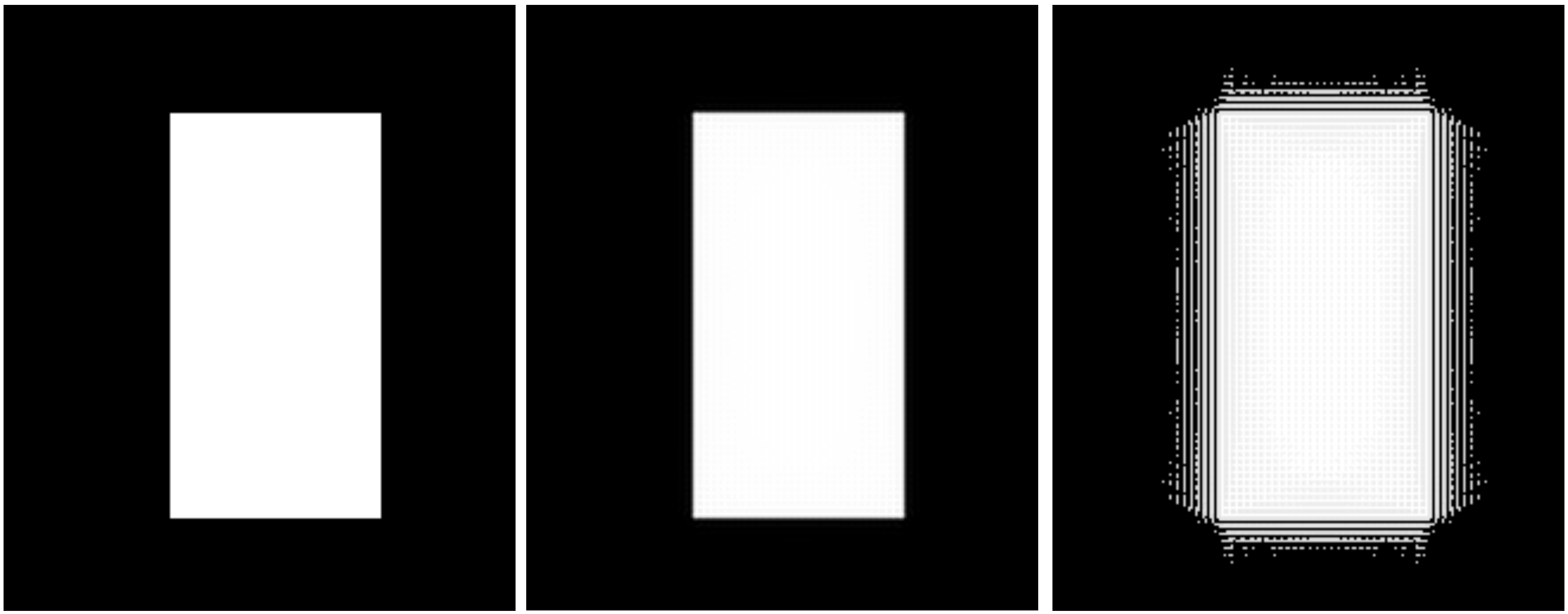


Image Filtered Filtered + HE

Ideal LP - 1/3

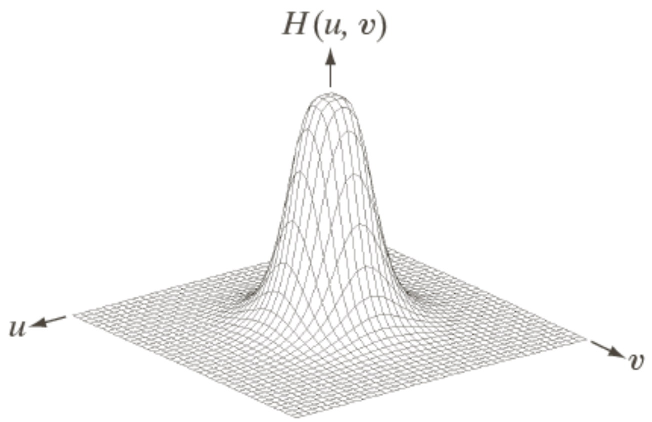
Ideal LP - 2/3

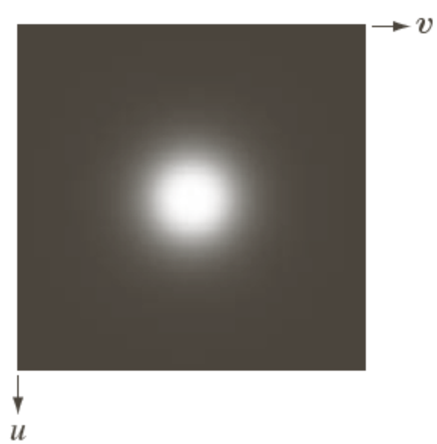
Butterworth Filter

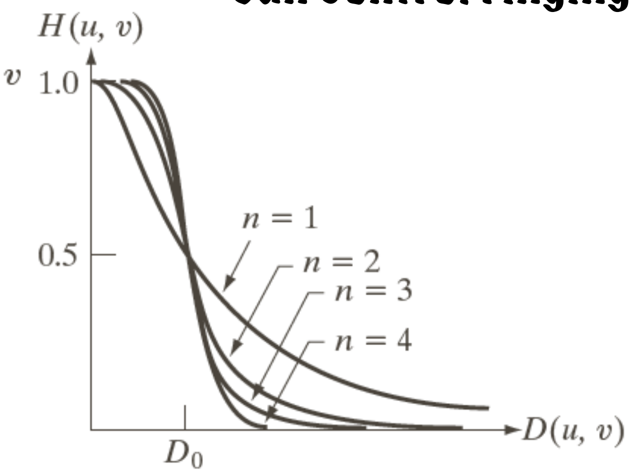
Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

Control of cutoff and slope
Can control ringing





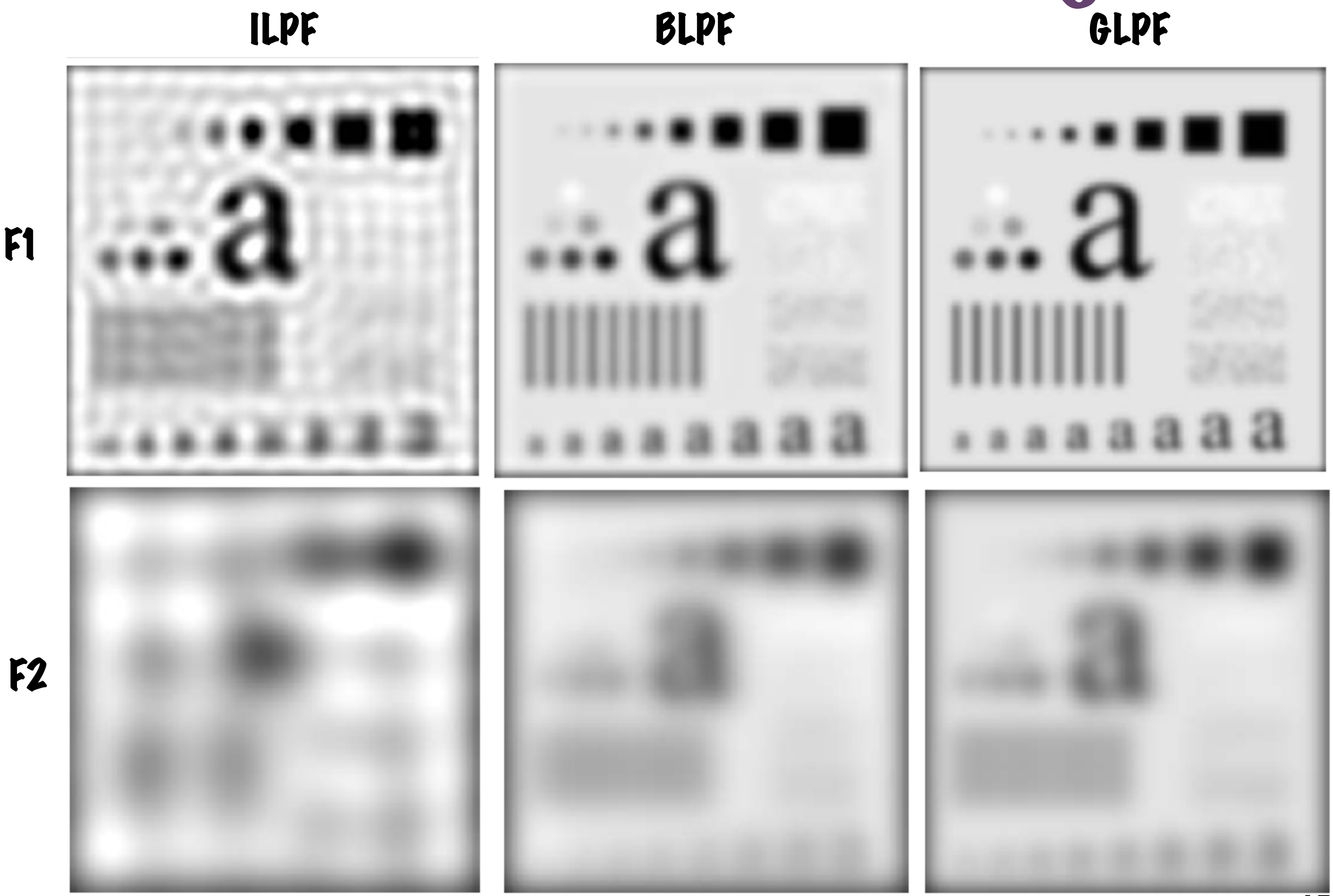


Butterworth - 1/3

Butterworth vs Ideal LP

Butterworth - 2/3

Gaussian LP Filtering



High Pass Filtering

- HP = 1 - LP
  - All the same filters as HP apply
- Applications
  - Visualization of high-freq data (accentuate)
- High boost filtering
  - HB = (1- a) + a(1 - LP) = 1 - a*LP

High-Pass Filters

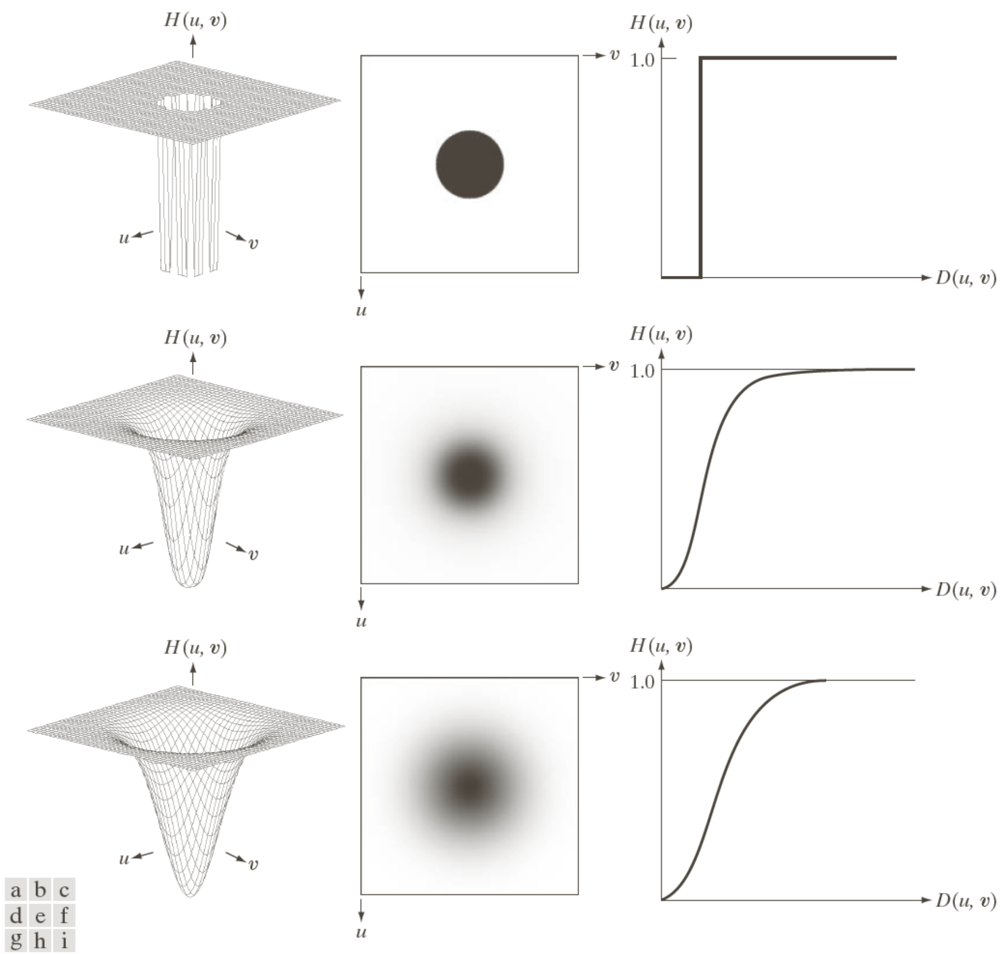
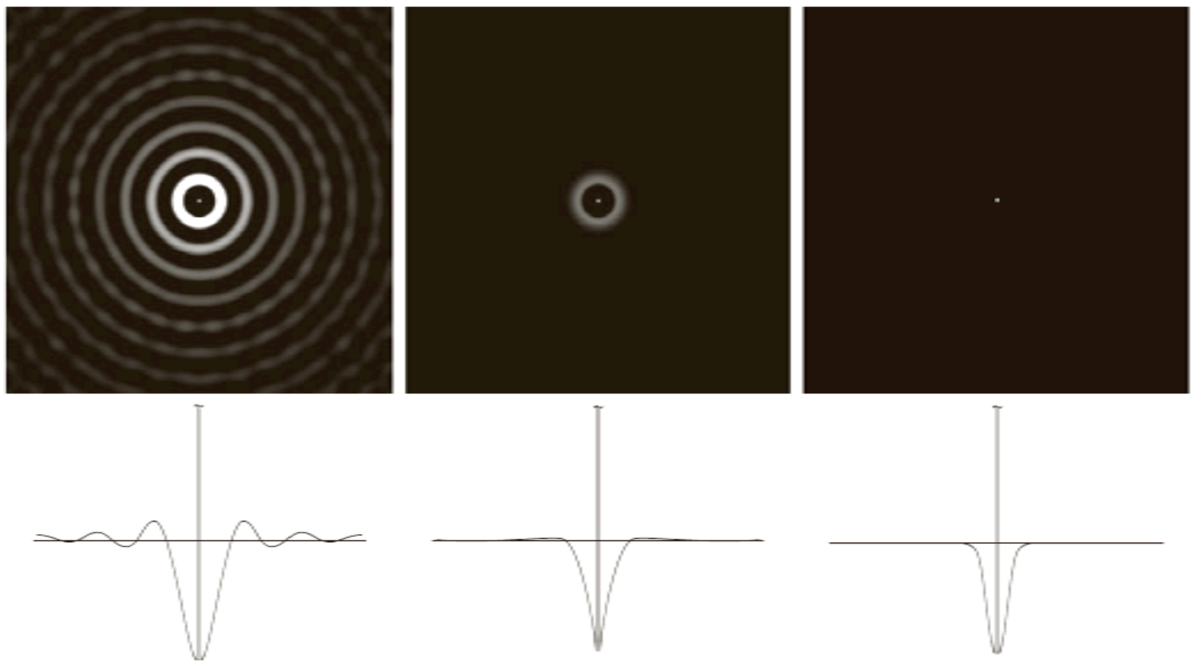


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

High-Pass Filters in Spatial Domain



a b c

FIGURE 4.53 Spatial representation of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding intensity profiles through their centers.

High-Pass Filtering with IHPF

a b c

FIGURE 4.54 Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with D_0 = 30, 60, and 160.

BHPF

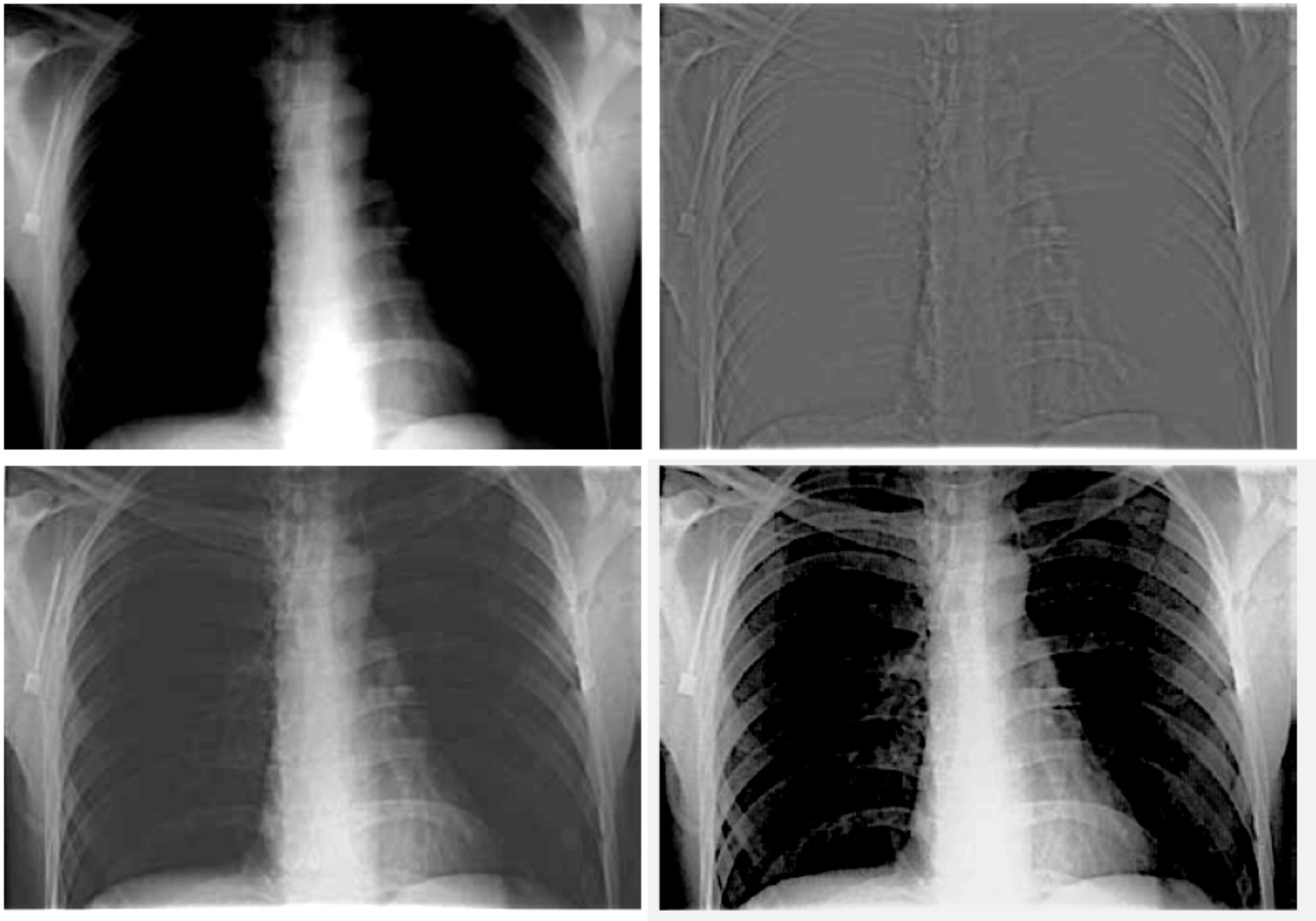
a b c

FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

GHPF

FIGURE 4.56 Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with D_0 = 30, 60, and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

HP, HB, HE



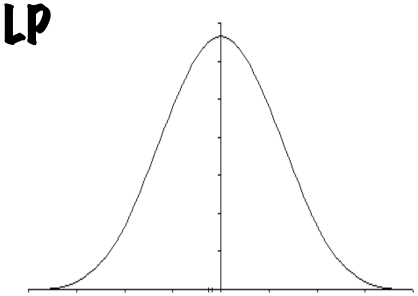
High Boost with GLPF

High-Boost Filtering

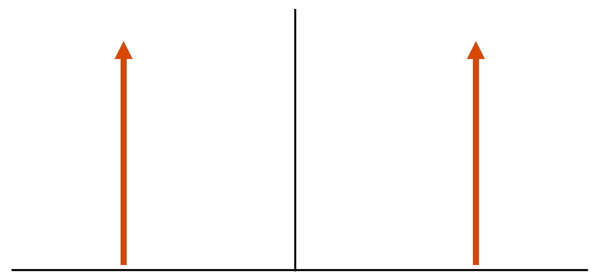
Band-Pass Filters

- Shift LP filter in Fourier domain by convolution with delta

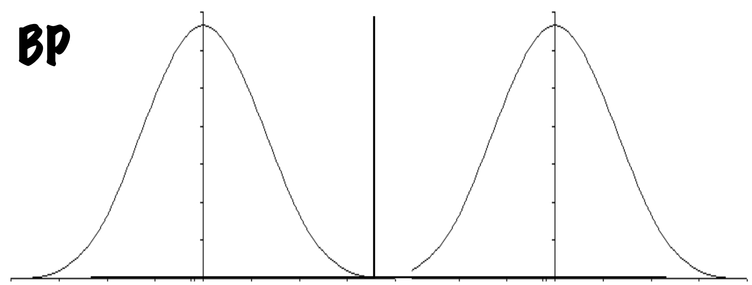
LP



$$\delta(s - s_0) + \delta(s + s_0)$$



BP

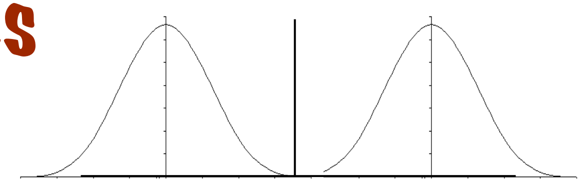


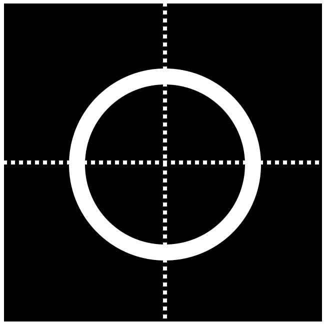
Typically 2-3 parameters

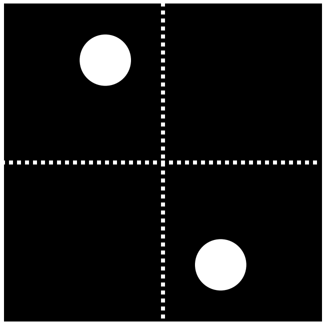
- -Width
- -Slope
- -Band value

Band Pass - Two Dimensions

- Two strategies
  - Rotate
    - Radially symmetric
  - Translate in 2D
    - Oriented filters

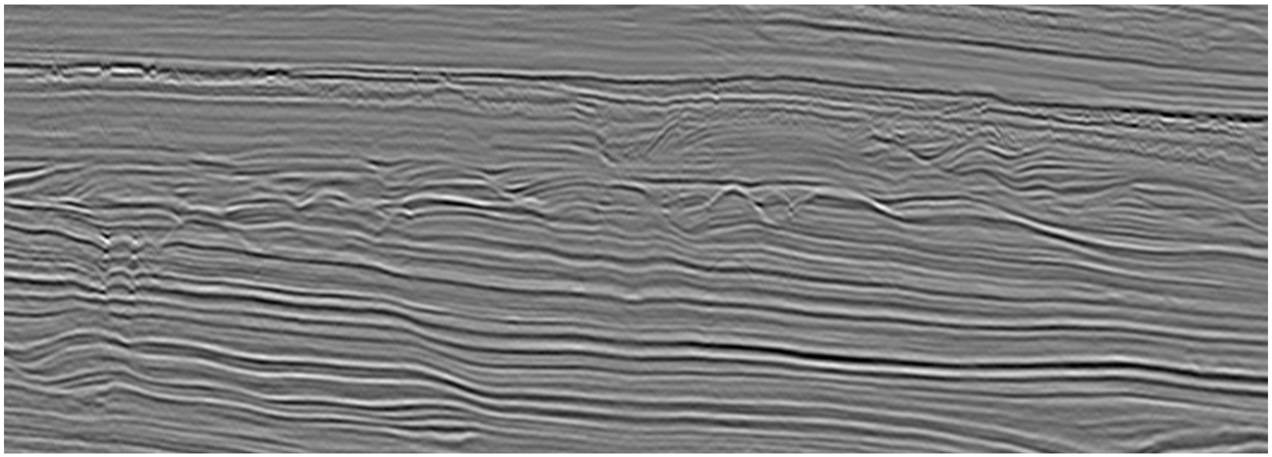


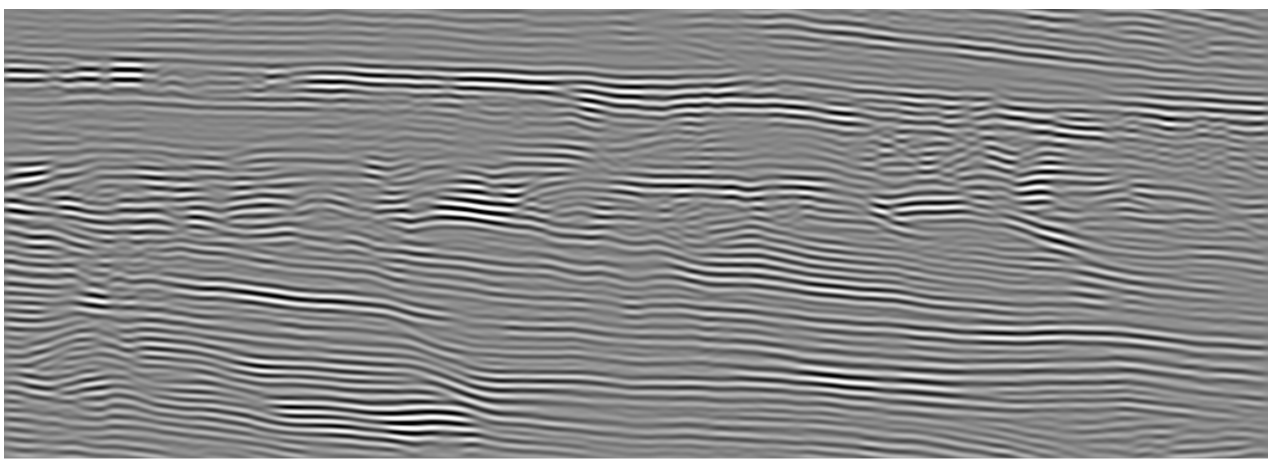




- Note:
  - Convolution with delta-pair in FD is multiplication with cosine in spatial domain

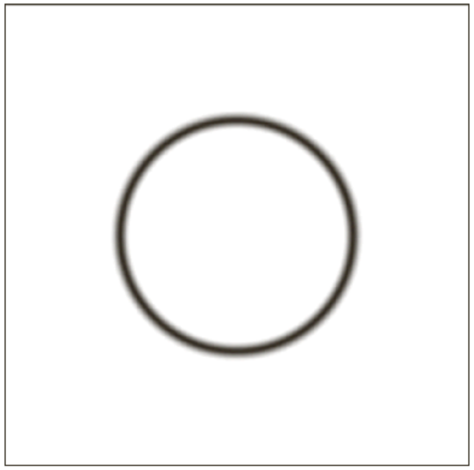
Band Bass Filtering

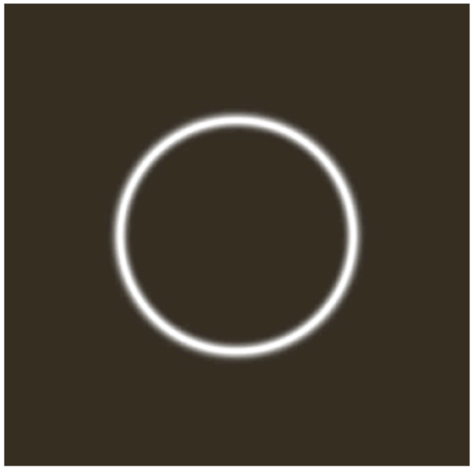




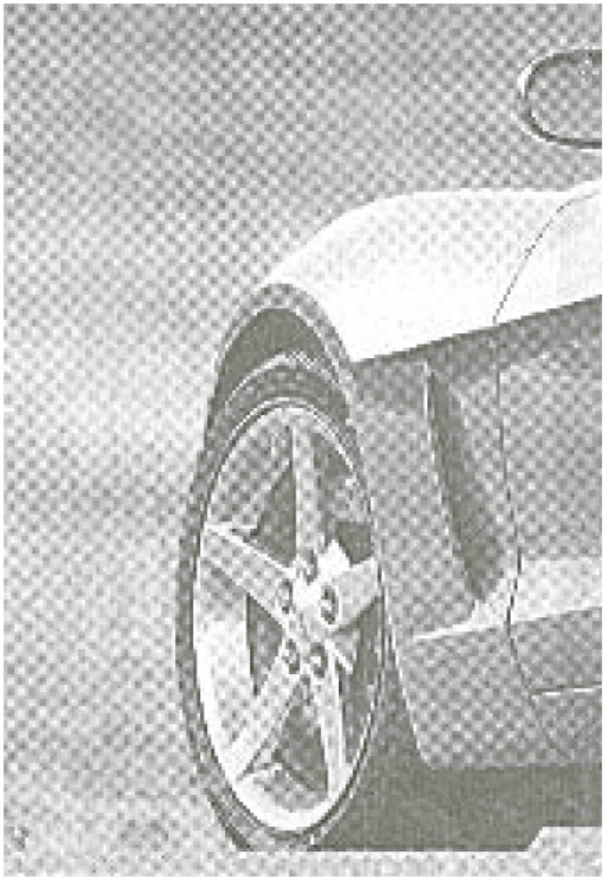
Radial Band Pass/Reject

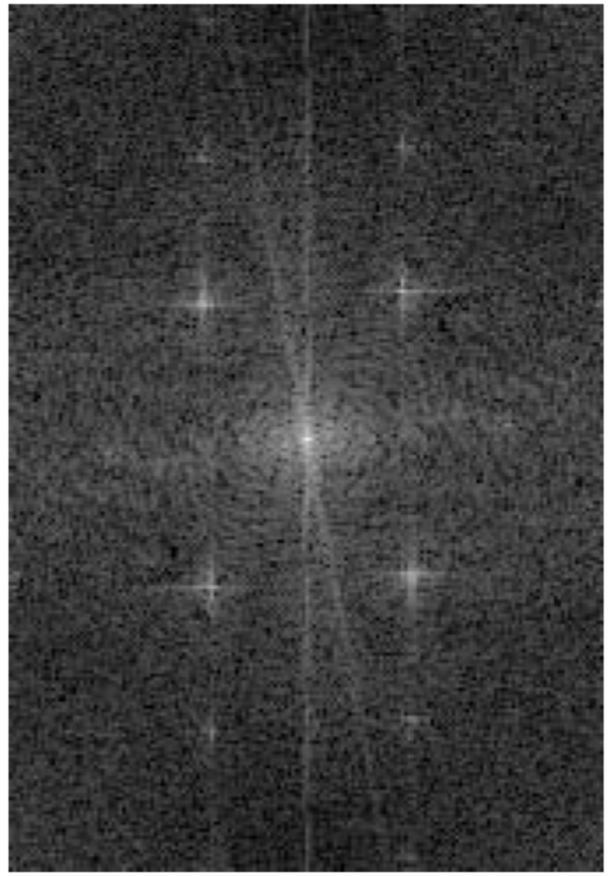
Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \frac{1}{1 + \left[\frac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$



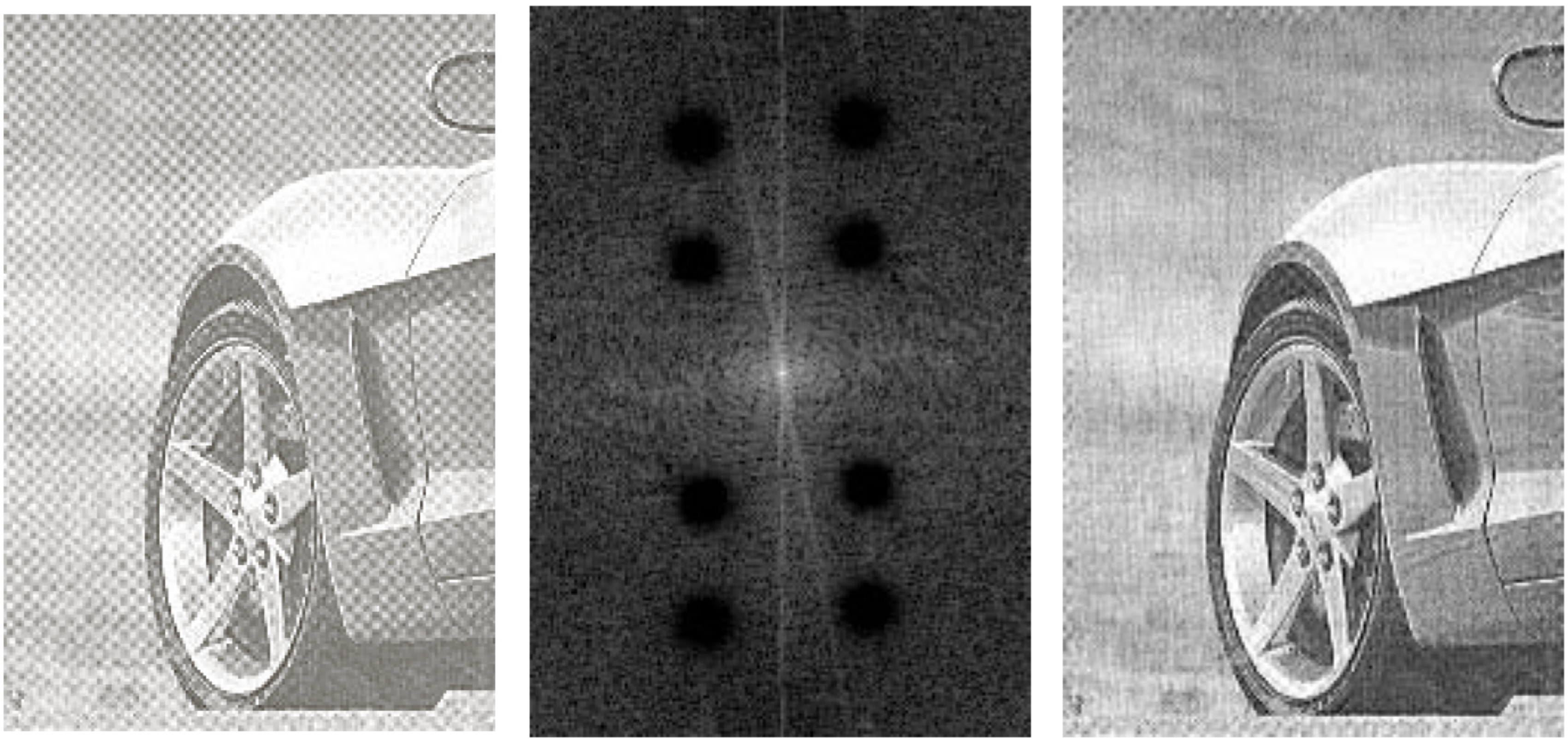


Band Reject Filtering

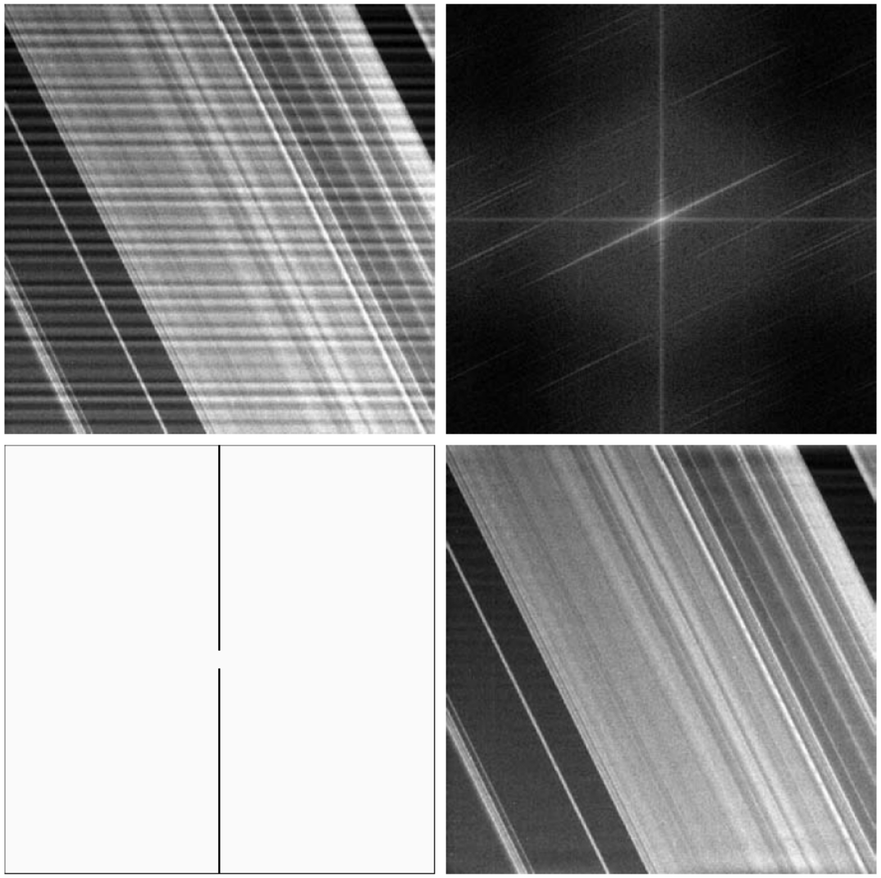




Band Reject Filtering



Band Reject Filtering

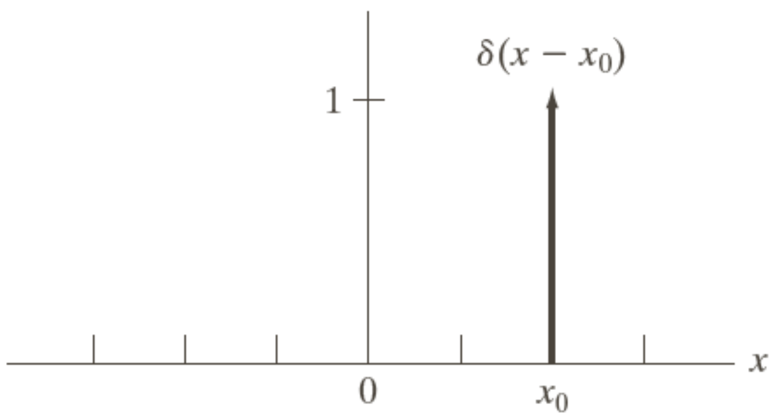


Discrete Sampling and Aliasing

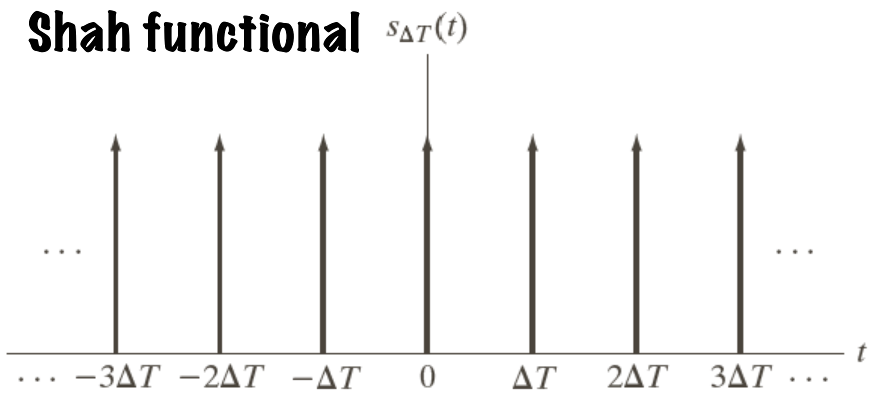
- Digital signals and images are discrete representations of the real world
  - Which is continuous
- What happens to signals/images when we sample them?
  - Can we quantify the effects?
  - Can we understand the artifacts and can we limit them?
  - Can we reconstruct the original image from the discrete data?

A Mathematical Model of Discrete Samples

Delta functional



Shah functional



$$s_{\Delta T}(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta T)$$

A Mathematical Model of Discrete Samples

- **Goal**
  - To be able to do a continuous Fourier transform on a signal before and after sampling

Discrete signal

$$f_k \quad k = 0, \pm 1, \ldots$$

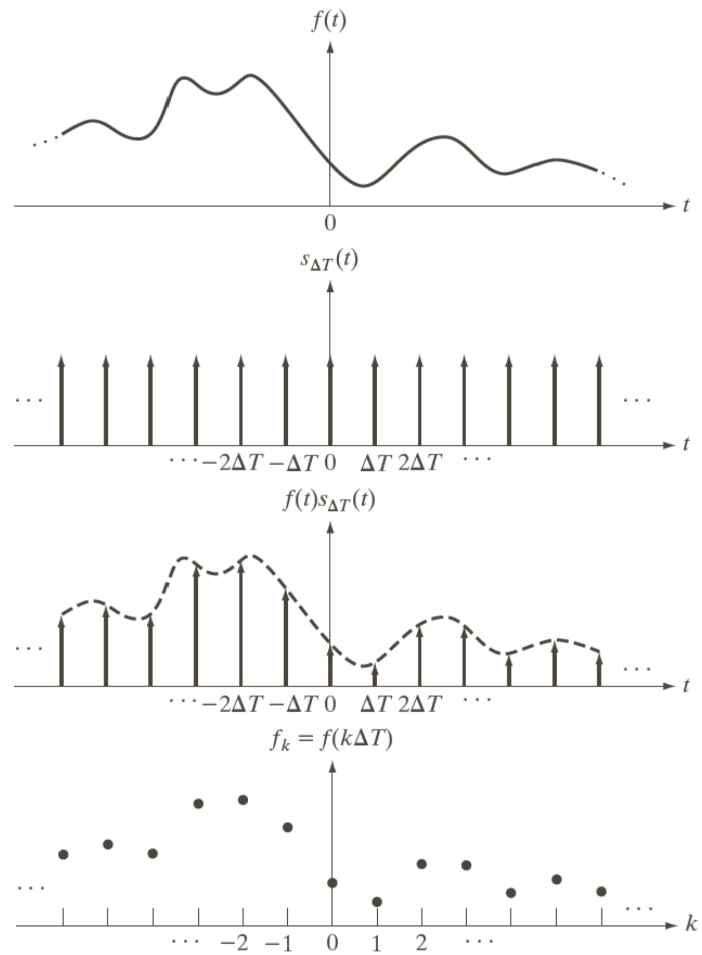
Samples from continuous function

$$f_k = f(k\Delta T)$$

Representation as a function of t

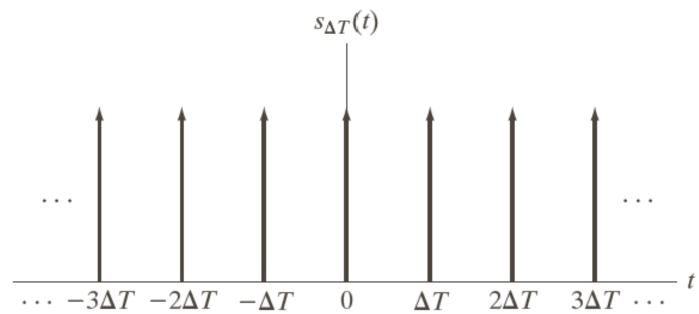
- Multiplication of f(t) with Shah

$$\tilde{f}(t) = f(t) s_{\Delta T}(t) = \sum_{k=-\infty}^{\infty} f_k \delta(t - k\Delta T)$$



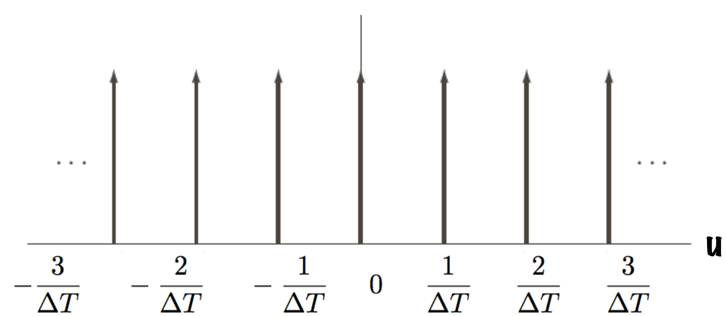
Fourier Series of A Shah Functional

$$s(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta T)$$



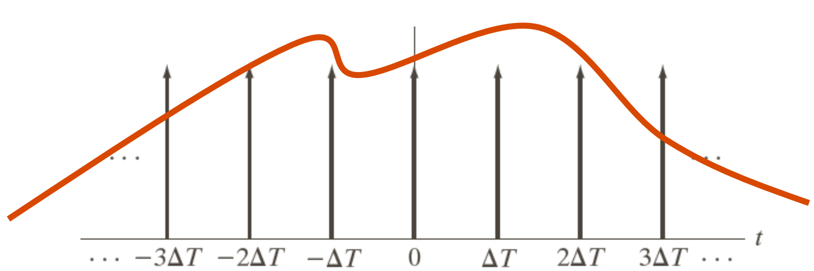
$$S(u) = \frac{1}{\Delta T} \sum_{k=-\infty}^{\infty} \delta(u - \frac{k}{\Delta T})$$

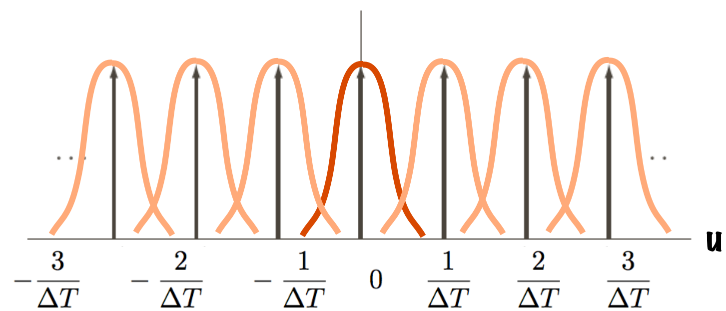
$$= \sum_{k=-\infty}^{\infty} \delta(\Delta T u - k)$$



Fourier Transform of A Discrete Sampling

$$\tilde{f}(t) = f(t)s(t) \longleftrightarrow \tilde{F}(u) = F(u) * S(u)$$

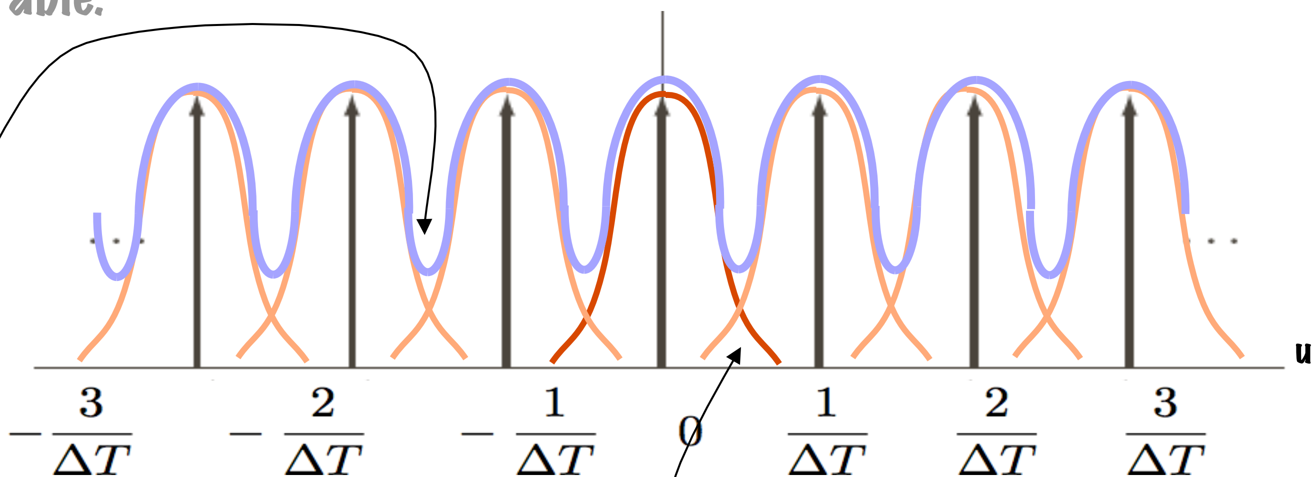




Fourier Transform of A Discrete Sampling

Frequencies get mixed. The original signal is not recoverable.

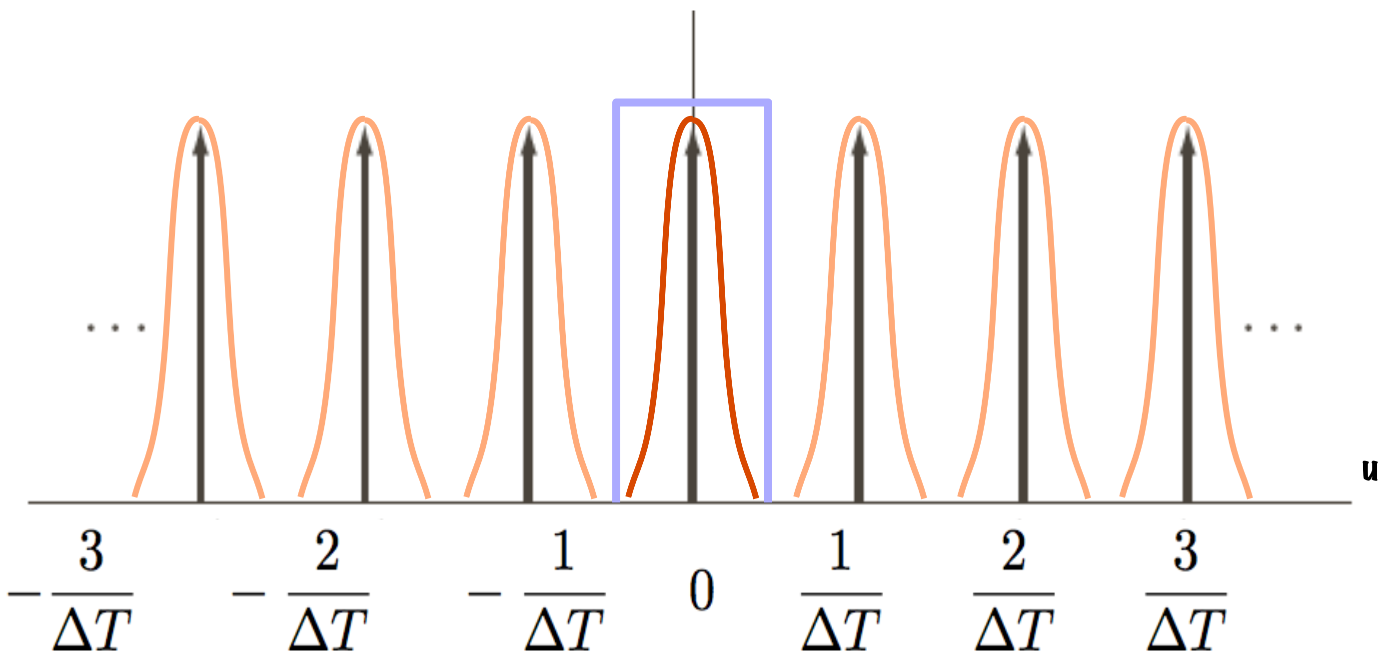
$$\tilde{F}(u) = F(u) * S(u)$$



Energy from higher freqs gets folded back down into lower freqs – Aliasing

What if F(u) is Narrower in the Fourier Domain?

- No aliasing!
- How could we recover the original signal?



What Comes Out of This Model

- Sampling criterion for complete recovery
- An understanding of the effects of sampling
  - Aliasing and how to avoid it
- Reconstruction of signals from discrete samples

Shannon Sampling Theorem

- Assuming a signal that is band limited:

$$f(t) \longleftrightarrow F(u) \qquad |F(u)| = 0 \;\; \forall \;\; |u| > B$$

- Given set of samples from that signal

$$f_k = f(k\Delta T) \qquad \Delta T \leq \frac{1}{2B}$$

- Samples can be used to generate the original signal
  - Samples and continuous signal are equivalent

Sampling Theorem

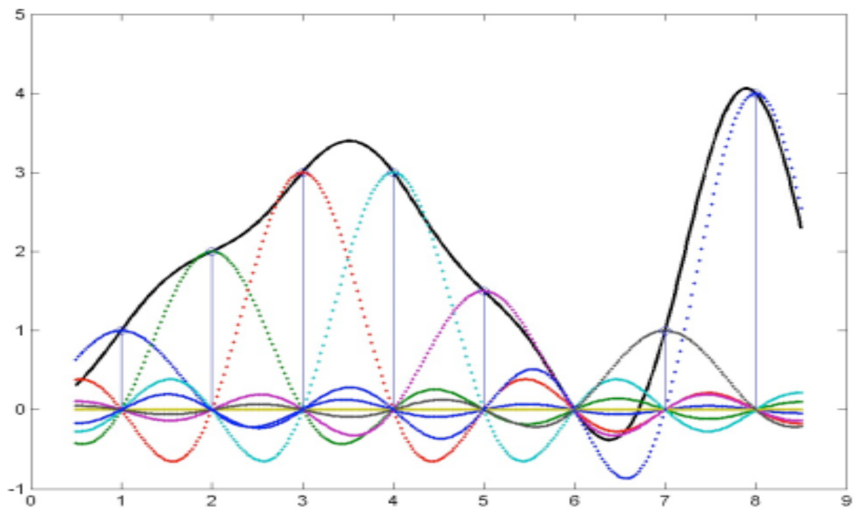
- Quantifies the amount of information in a signal
  - Discrete signal contains limited frequencies
  - Band-limited signals contain no more information then their discrete equivalents
- Reconstruction by cutting away the repeated signals in the Fourier domain
  - Convolution with sinc function in space/time

Reconstruction

- Convolution with sinc function

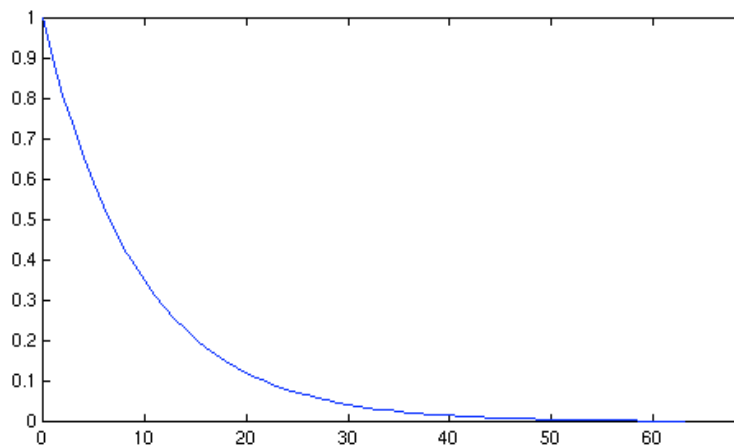
$$f(t) = \tilde{f}(t) * \mathbb{F}^{-1}\left[\mathrm{rect}\left(\Delta \mathrm{Tu}\right)\right]$$

$$= \left(\sum_k f_k \delta(t - k\Delta T)\right) * \mathrm{sinc}\left(\frac{\mathrm{t}}{\Delta \mathrm{T}}\right) = \sum_{\mathrm{k}} \mathrm{f_k}\ \mathrm{sinc}\left(\frac{\mathrm{t} - \mathrm{k}\Delta \mathrm{T}}{\Delta \mathrm{T}}\right)$$

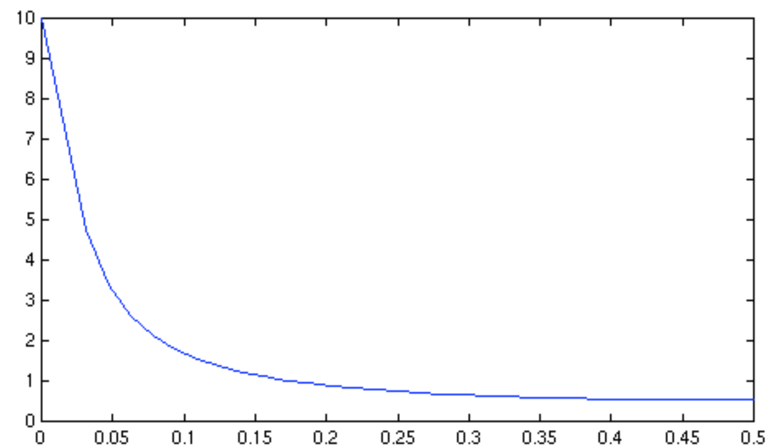


Sinc Interpolation Issues

- Must functions are not band limited
- Forcing functions to be band-limited can cause artifacts (ringing)

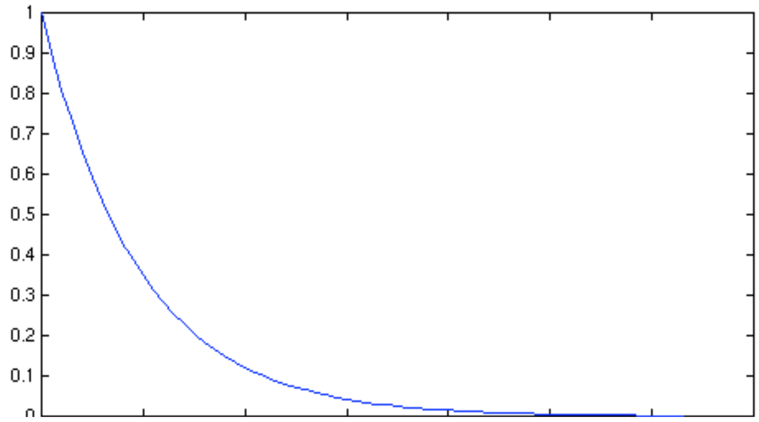


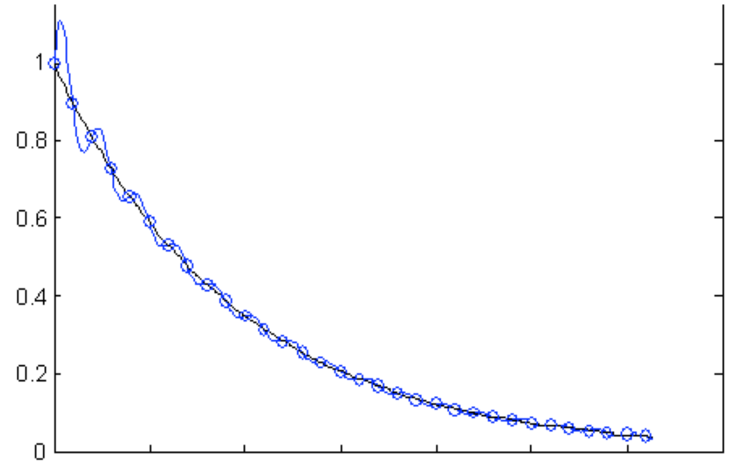
f(t)



|F(s)|

Sinc Interpolation Issues





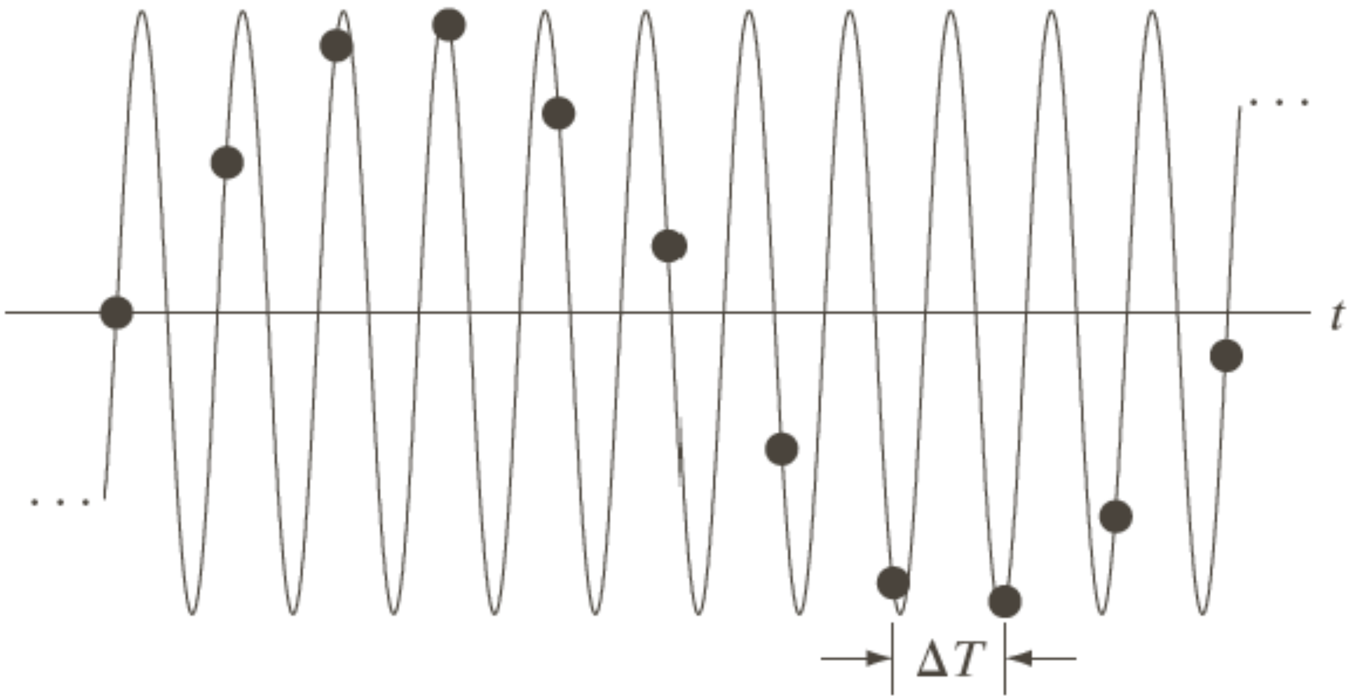
Ringing – Gibbs phenomenon

Other issues:

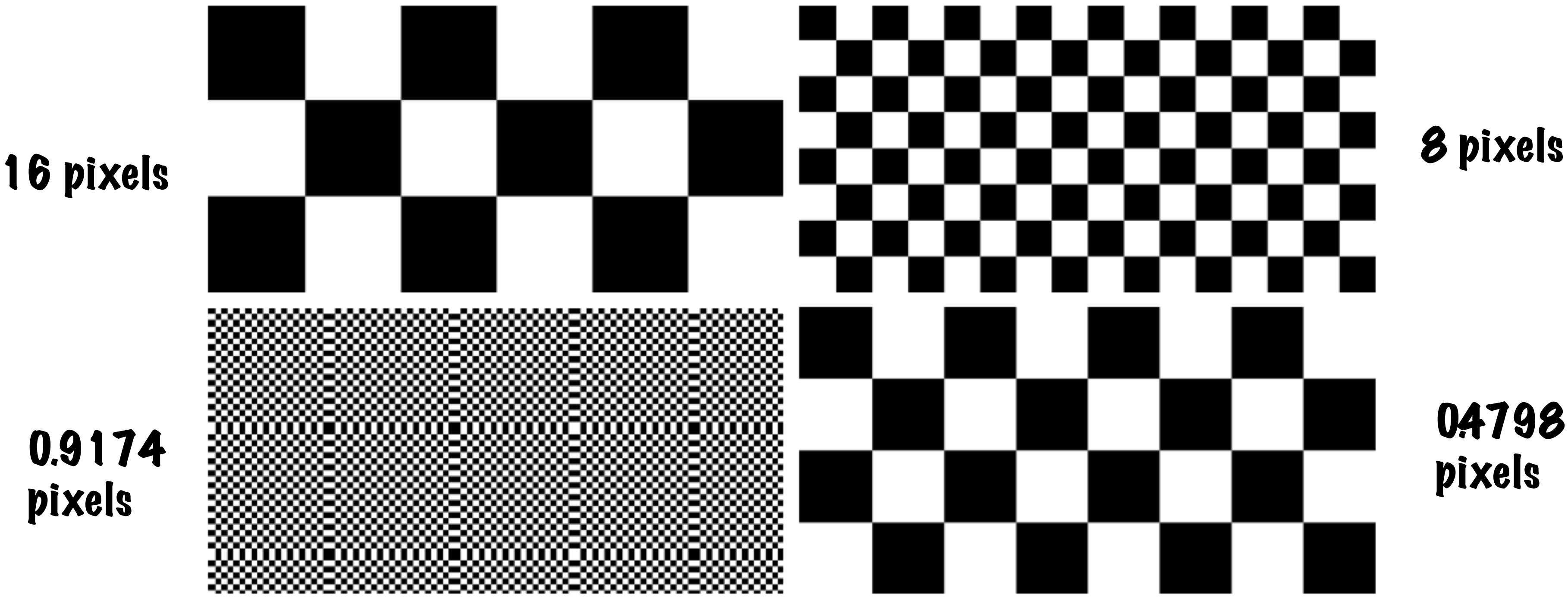
Sinc is infinite – must be truncated

Aliasing

- High frequencies appear as low frequencies when undersampled

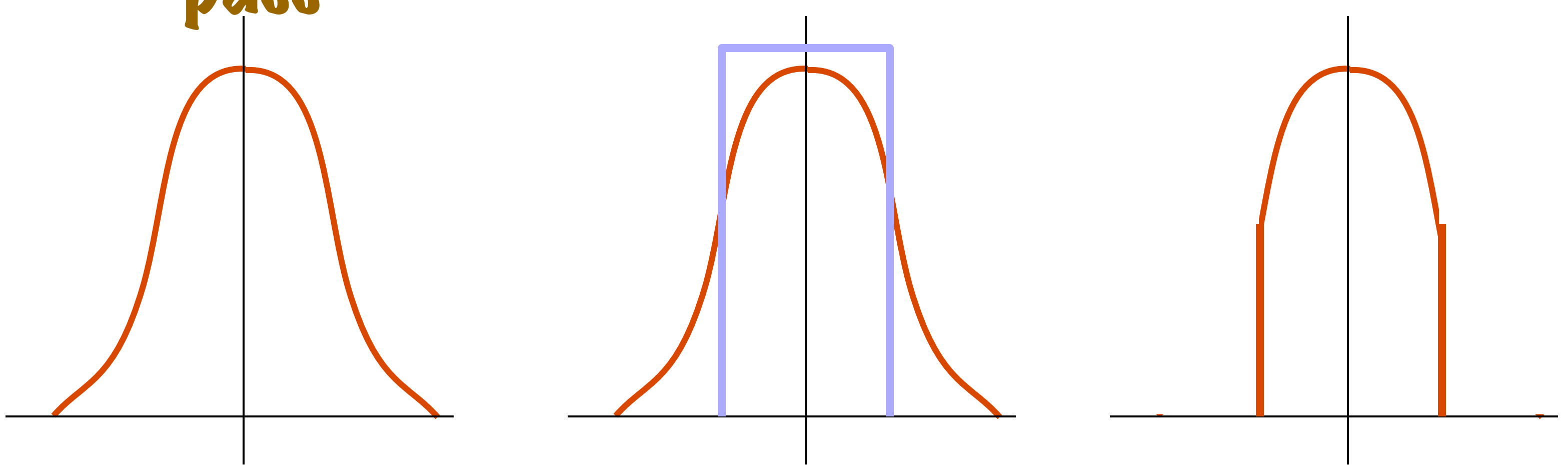


Aliasing



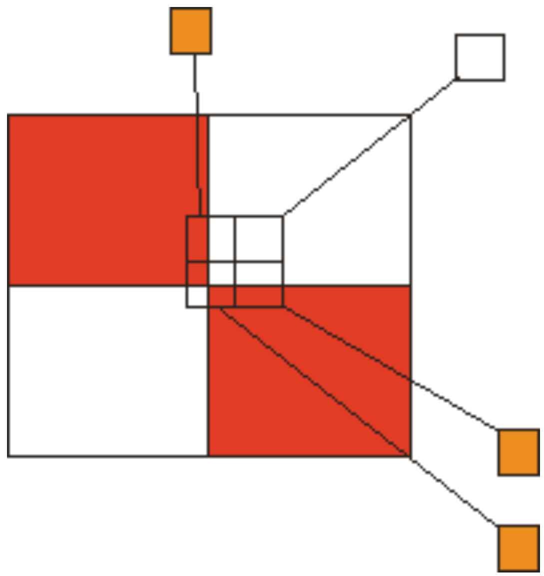
Overcoming Aliasing

- Filter data prior to sampling
  - Ideally - band limit the data (conv with sinc function)
  - In practice - limit effects with fuzzy/soft low pass



Antialiasing in Graphics

- Screen resolution produces aliasing on underlying geometry



Multiple high-res samples get averaged to create one screen sample

Antialiasing

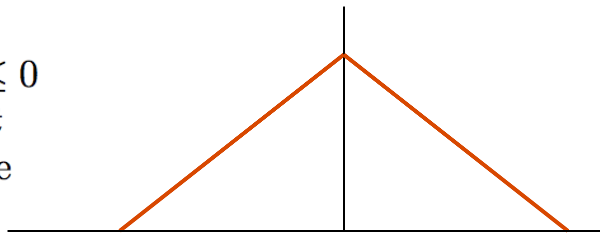
Interpolation as Convolution

- Any discrete set of samples can be considered as a functional

$$\tilde{f}(t) \;=\; \sum_k f_k \delta(t - k\Delta T)$$

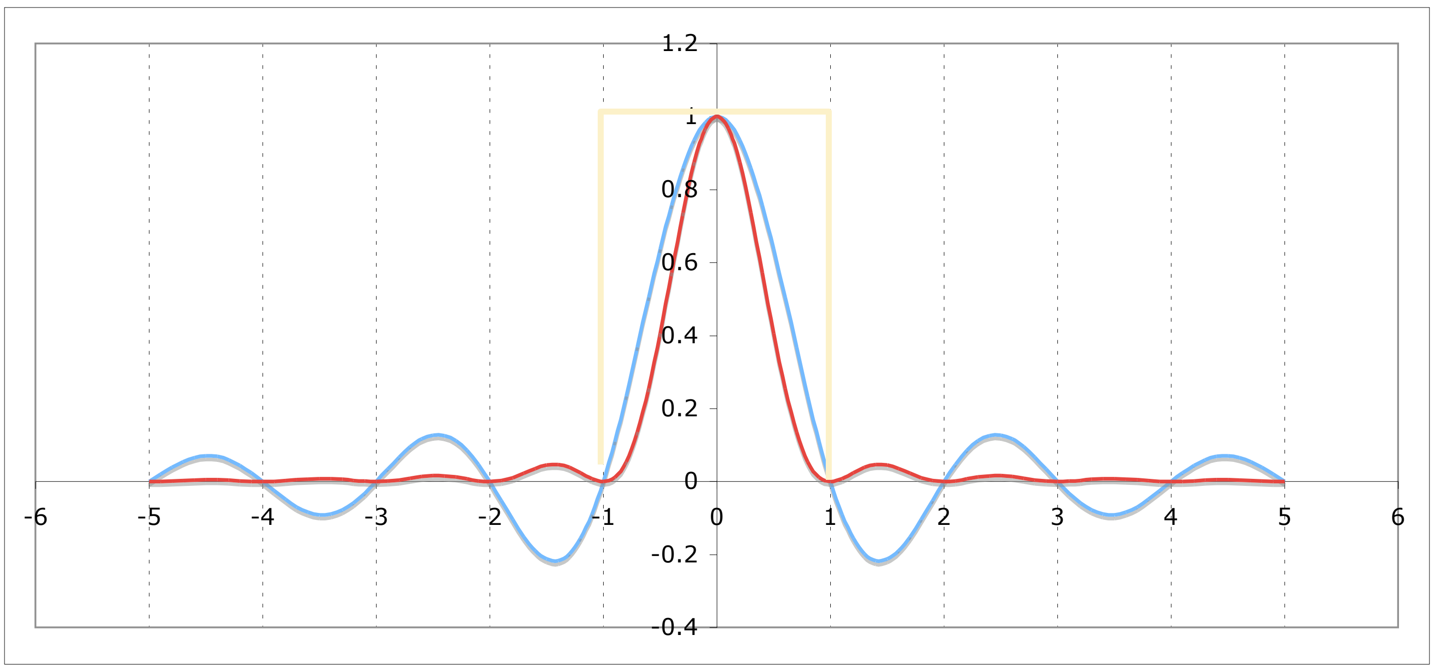
- Any linear interpolant can be considered as a convolution
  - Nearest neighbor - rect(t)
  - Linear - tri(t)

$$\mathrm{tri(t)} = \begin{cases} t+1 & -1 \leq t \leq 0 \\ 1-t & 0 \leq t \leq t \\ 0 & \text{otherwise} \end{cases}$$

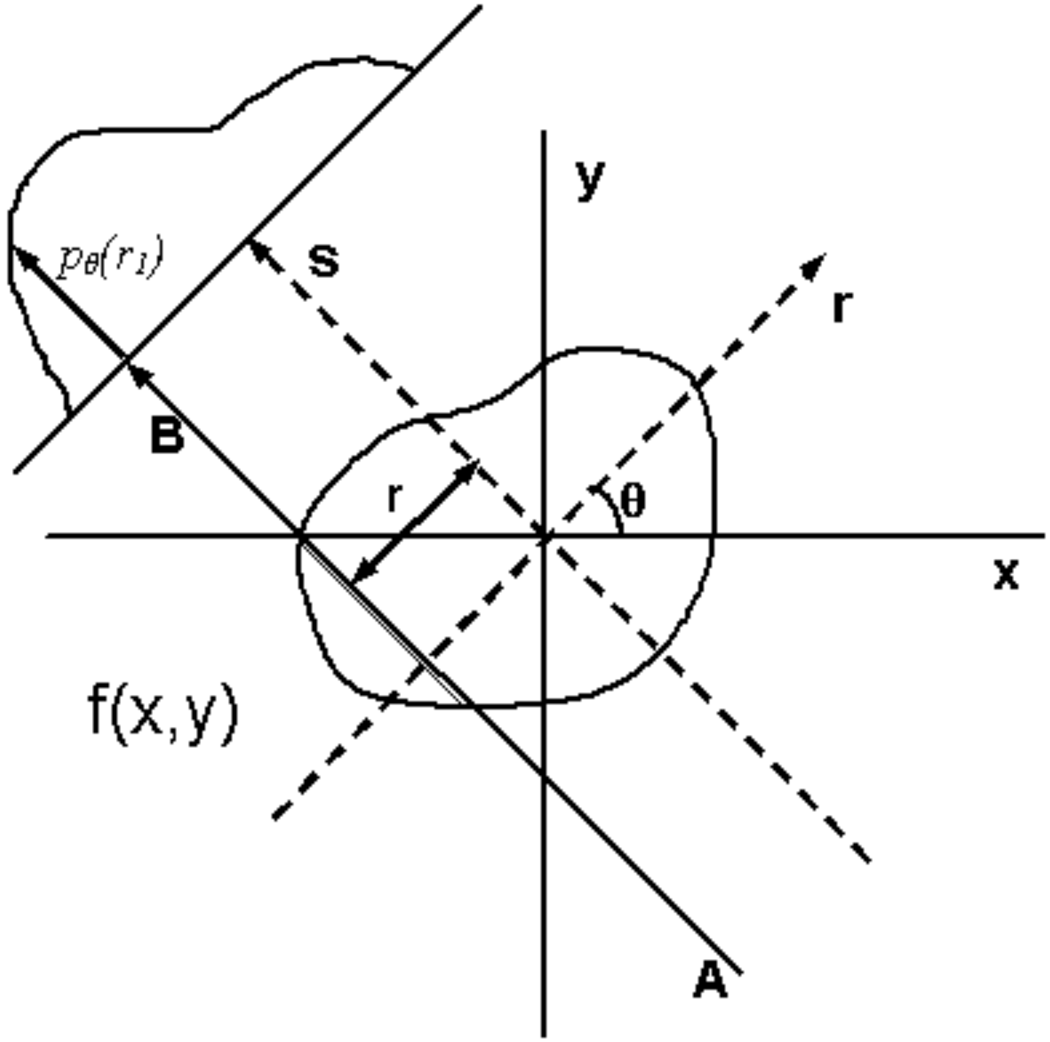


Convolution-Based Interpolation

- Can be studied in terms of Fourier Domain
- Issues
  - Pass energy (=1) in band
  - Low energy out of band
  - Reduce hard cut off (Gibbs, ringing)



Tomography



Tomography Formulation

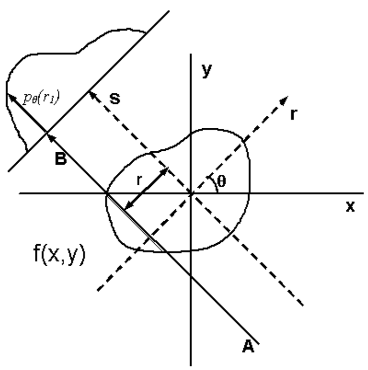
Attenuation $I = I_0 \exp\left(-\int \mu(x,y)\, ds\right)$

Log gives line integral $p(r,\theta) = \ln(I/I_0) = -\int \mu(x,y)\, ds$

Line with angle theta $x\cos\theta + y\sin\theta = r$

Volume integral $p(r,\theta) =$

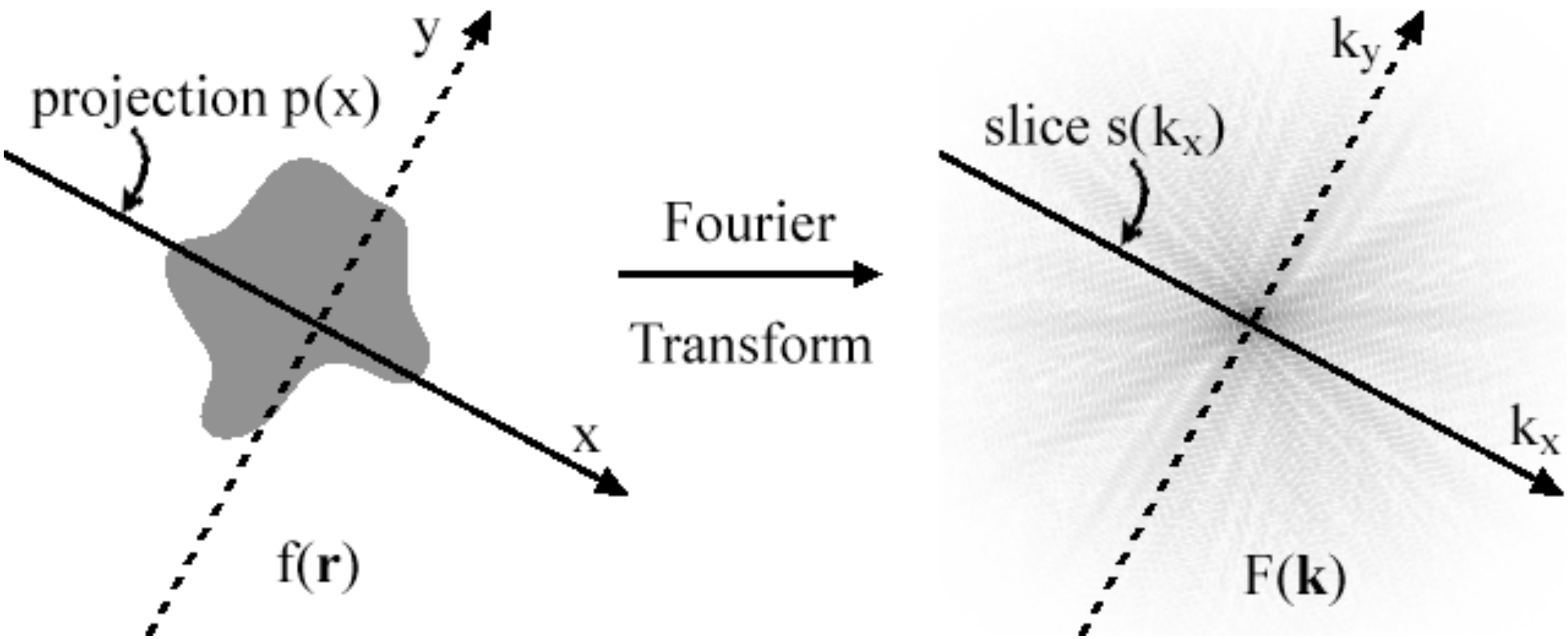
$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - r)\, dx\, dy$$



Fourier Slice Theorem

1D FT — Projection to 1D — 1D Slice — 2D FT

$$F_1 P_1 = S_1 F_2$$



Optimal Filtering

- Systems model

$$y(t) = h(t) * x(t) + v(t)$$

- Ergodic signals
  - Drawn from a ensemble
  - Average over ensemble is constant
  - Average over time is ensemble average
  - --> Expected value of power spectrum describes ensemble

Optimal/Weiner Filter

- Power spectrum of signal, noise are known
- H(u) is known
- Filter that minimizes the expected squared error of reconstruction is:

$$G(u) = \frac{H^*(u)S(u)}{|H(u)|^2 S(u) + N(u)}$$

Optimal/Weiner Filter

$$G(u) = \frac{1}{H(u)} \frac{|H(u)|^2}{|H(u)|^2 + \frac{N(u)}{S(u)}}$$

$$G(u) = \frac{1}{H(u)} \frac{|H(u)|^2}{|H(u)|^2 + \frac{1}{SNR(u)}}$$

Image Registration

- Find dx and dy that best matches two images
- Cross correlation can give the best translation between two images
- Algorithms
  - SD -> FD (mult) -> SD - look for peak
  - SD -> FD -> find the best fit for a phase shift
- Issues
  - Boundaries, overlap, intensity variations, high intensity edges

Normalized Cross Correlation

- Subtract the mean of the image and divide by the S.D.
  - This maps the image to the unit sphere
  - A single integral is the dot product of these to vectors
    - angles between the two normalized images
  - Helps alleviate intensity differences

Phase Correlation

$$\mathbf{G}_a = \mathcal{F}\{g_a\},\ \mathbf{G}_b = \mathcal{F}\{g_b\}$$

$$R = \frac{\mathbf{G}_a \mathbf{G}_b^*}{|\mathbf{G}_a \mathbf{G}_b^*|}$$

$$r = \mathcal{F}^{-1}\{R\}$$

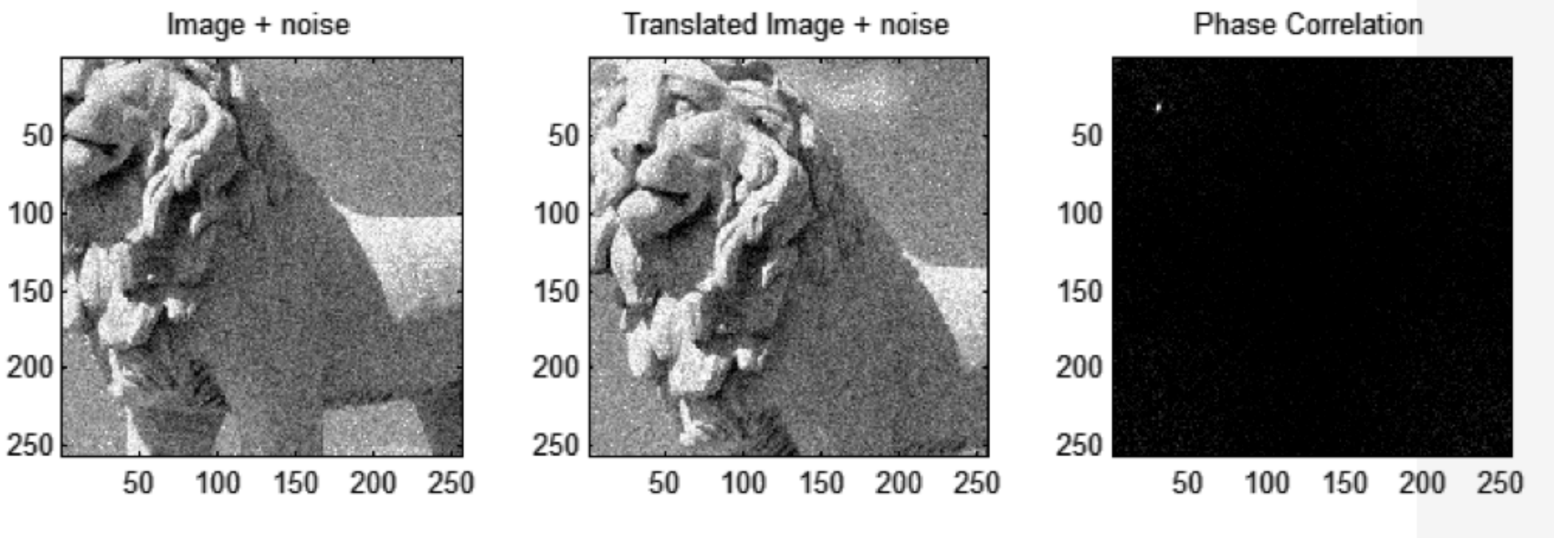
$$(\Delta x, \Delta y) = \arg\max_{(x,y)}\{r\}$$

$$g_b(x,y) \stackrel{\text{def}}{=} g_a((x - \Delta x) \bmod M, (y - \Delta y) \bmod N)$$

$$\mathbf{G}_b(u,v) = \mathbf{G}_a(u,v) e^{-2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}$$

$$\begin{aligned} R(u,v) &= \frac{\mathbf{G}_a \mathbf{G}_b^*}{|\mathbf{G}_a \mathbf{G}_b^*|} \\ &= \frac{\mathbf{G}_a \mathbf{G}_a^* e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}}{|\mathbf{G}_a \mathbf{G}_a^* e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}|} \\ &= \frac{\mathbf{G}_a \mathbf{G}_a^* e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}}{|\mathbf{G}_a \mathbf{G}_a^*|} \\ &= e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})} \end{aligned}$$

Phase Correlation



For Midterm – Pseudocode

- High level code to describe algorithm
- Make up reasonable key words
- Key idea – convey the algorithm
- Can adopt syntax from any major programming language
  - Be consistent

Psuedocode

Input: READ, OBTAIN, GET
Output: PRINT, DISPLAY, SHOW
Compute: COMPUTE, CALCULATE, DETERMINE
Initialize: SET, INIT
Add one: INCREMENT, BUMP

```
IF HoursWorked > NormalMax THEN

     Display overtime message

ELSE

     Display regular time message

ENDIF
```

```
WHILE employee.type NOT EQUAL manager AND personCount < numEmployees

      INCREMENT personCount
      CALL employeeList.getPerson with personCount RETURNING employee

ENDWHILE
```

Psuedocode

```
SET Carry to 0
FOR each DigitPosition in Number from least significant to most significant

    COMPUTE Total as sum of FirstNum[DigitPosition] and SecondNum[DigitPosition] and Carry

    IF Total > 10 THEN
        SET Carry to 1
        SUBTRACT 10 from Total
    ELSE
        SET Carry to 0
    END IF

    STORE Total in Result[DigitPosition]

END LOOP

IF Carry = 1 THEN
    RAISE Overflow exception
END IF
```