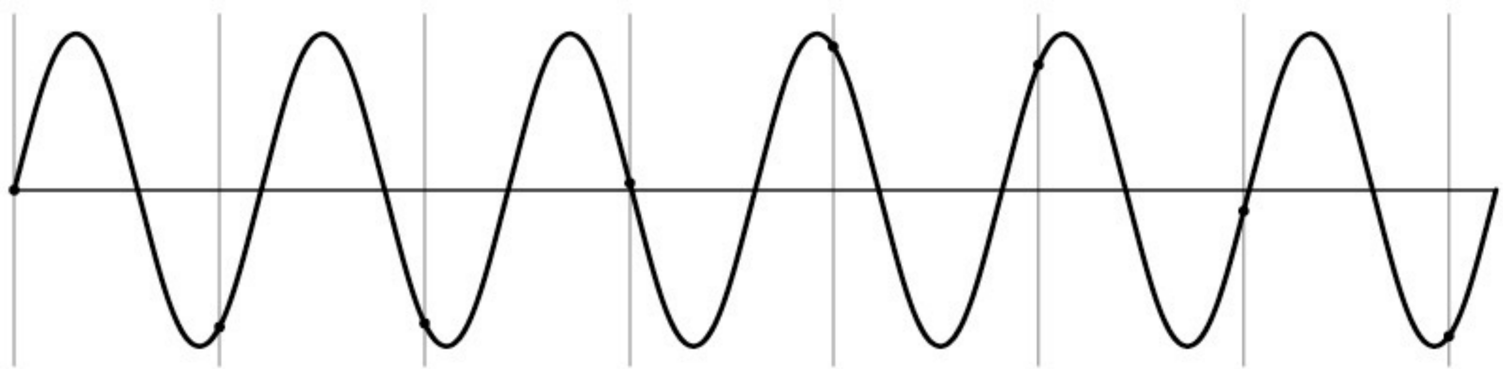
Sampling Theory

CS5625 Lecture 7

Sampling example (reminder)

- When we sample a high-frequency signal we don't get what we expect
  - result looks like a lower frequency
  - not possible to distinguish between this and a low-frequency signal



Sneak preview

- Sampling creates copies of the signal at higher frequencies
- Aliasing is these frequencies leaking into the reconstructed signal
  - frequency $f_s - x$ shows up as frequency x
- The solution is filtering
  - during sampling, filter to keep the high frequencies out so they don't create aliases at the lower frequencies
  - during reconstruction, again filter high frequencies to avoid including high-frequency aliases in the output.

Checkpoint

- Want to formalize sampling and reconstruction
  - define impulses
  - then we can talk about S&R with only one datatype
- Define Fourier transform
- Destination: explaining how aliases leak into result

Mathematical model

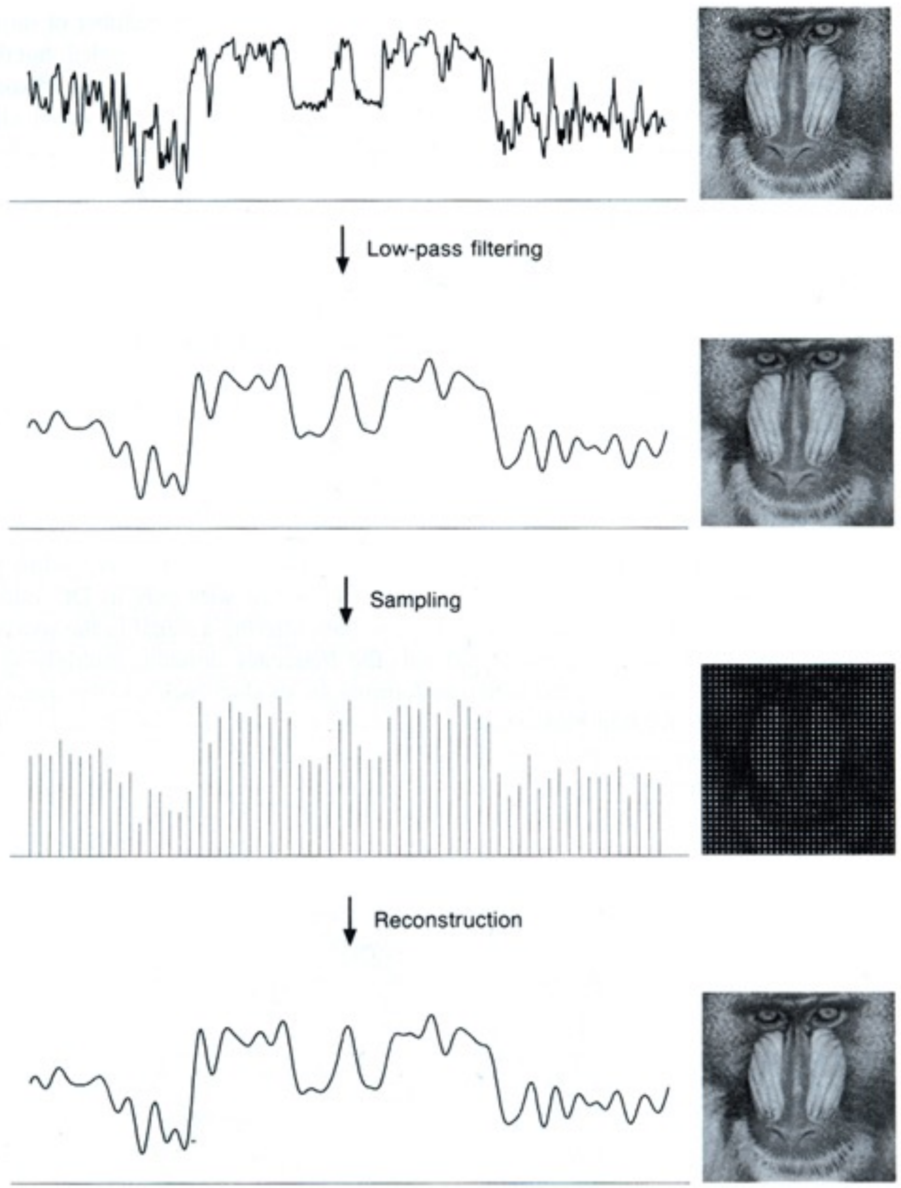
- We have said sampling is storing the values on a grid
- For analysis it's useful to think of the sampled representation in the same space as the original
  - I'll do this using *impulse functions* at the sample points

Impulse function

- A function that is confined to a very small interval
  - but still has unit integral
  - really, the limit of a sequence of ever taller and narrower functions
  - also called Dirac delta function
- Key property: multiplying by an impulse selects the value at a point
  - Defn via integral
- Impulse is the identity for convolution
  - “impulse response” of a filter

Sampling & recon. reinterpreted

- Start with a continuous signal
- Convolve it with the sampling filter
- Multiply it by an impulse grid
- Convolve it with the reconstruction filter



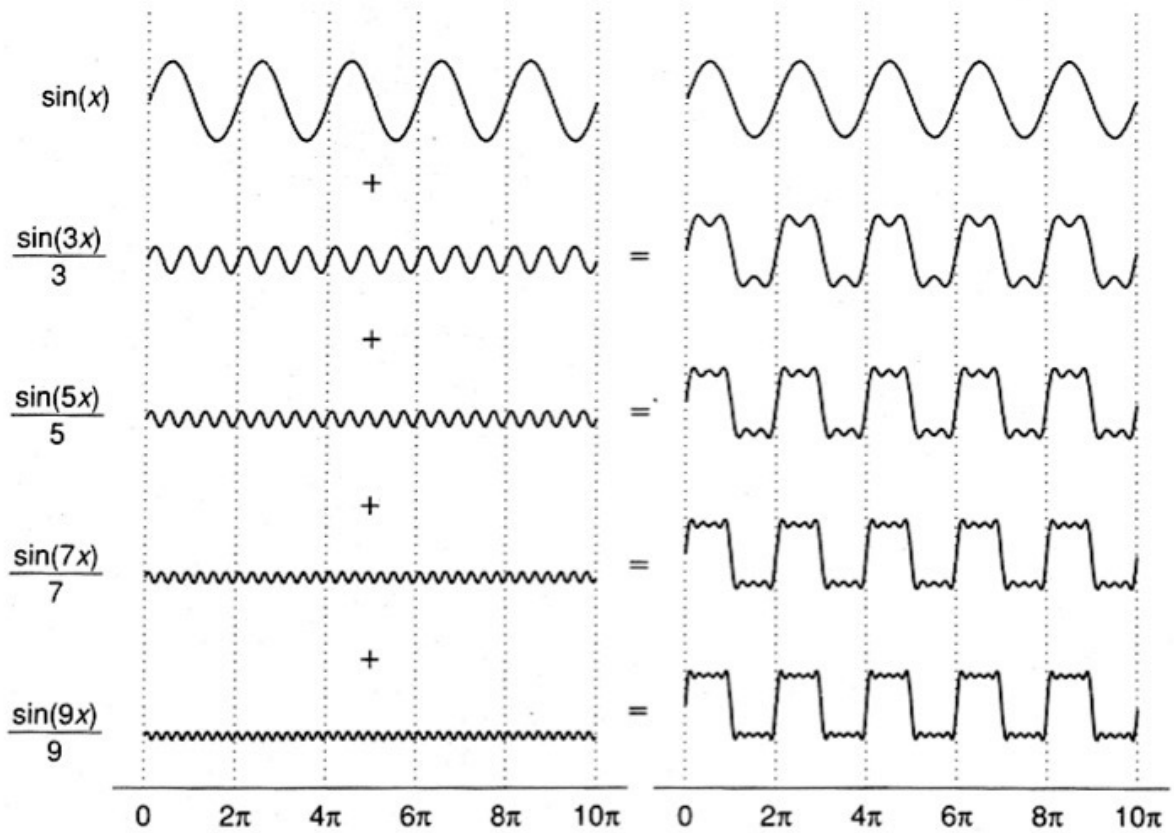
[F]

Checkpoint

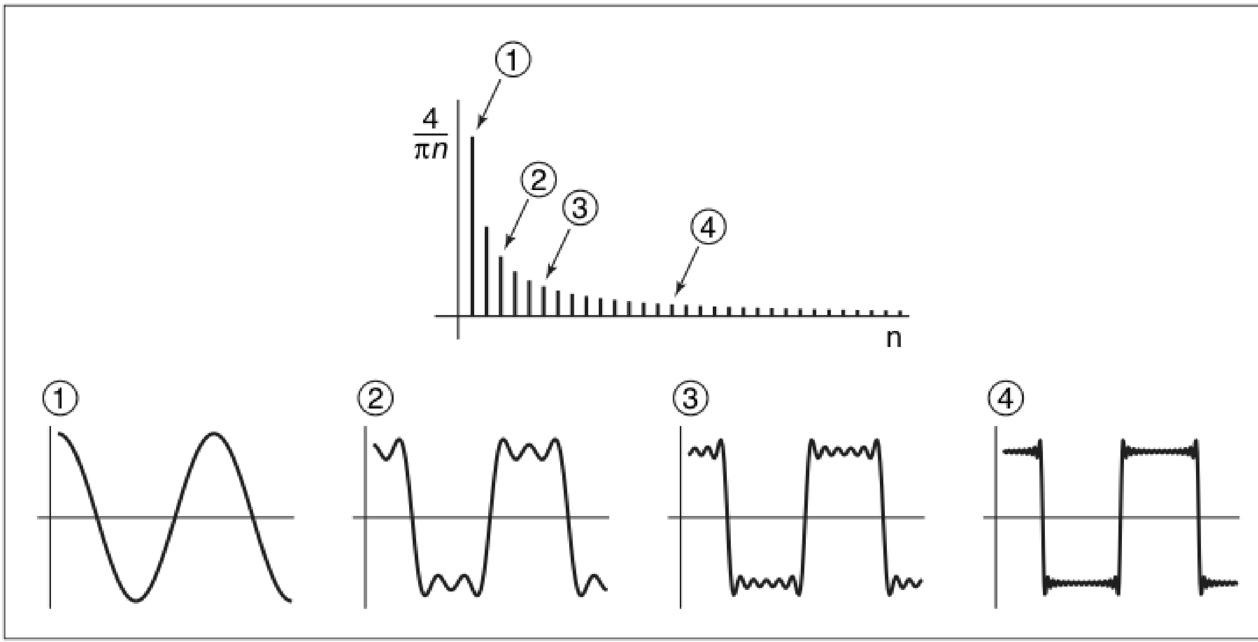
- Formalized sampling and reconstruction
  - used impulses with multiplication and convolution
  - can talk about S&R with only one datatype
- Define Fourier transform
- Destination: explaining how aliases leak into result

Fourier series

- Probably familiar idea of adding up sines and cosines to approximate a periodic function



Fourier series

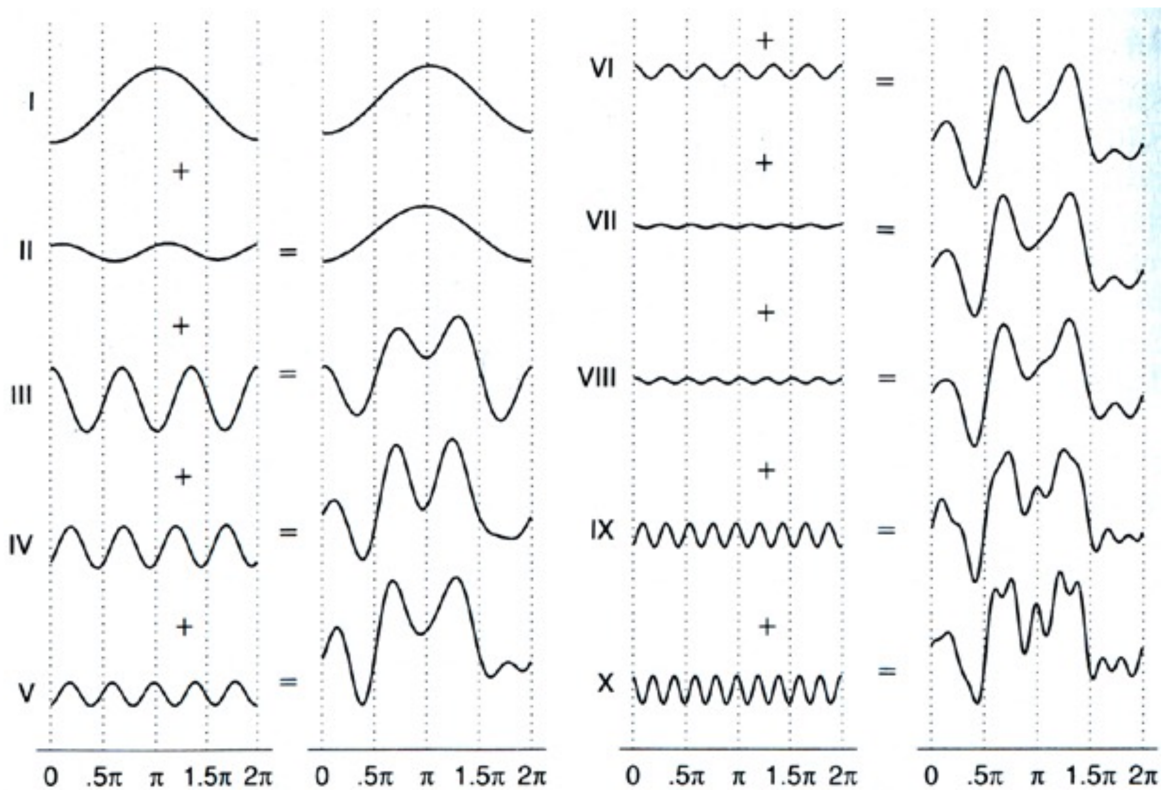


Fourier transform

- Like Fourier series but for aperiodic functions
  - Fourier series: only multiples of base frequency
- Fourier transform: let period go to infinity
  - eventually all frequencies are needed
  - result: countable sum turns into integral

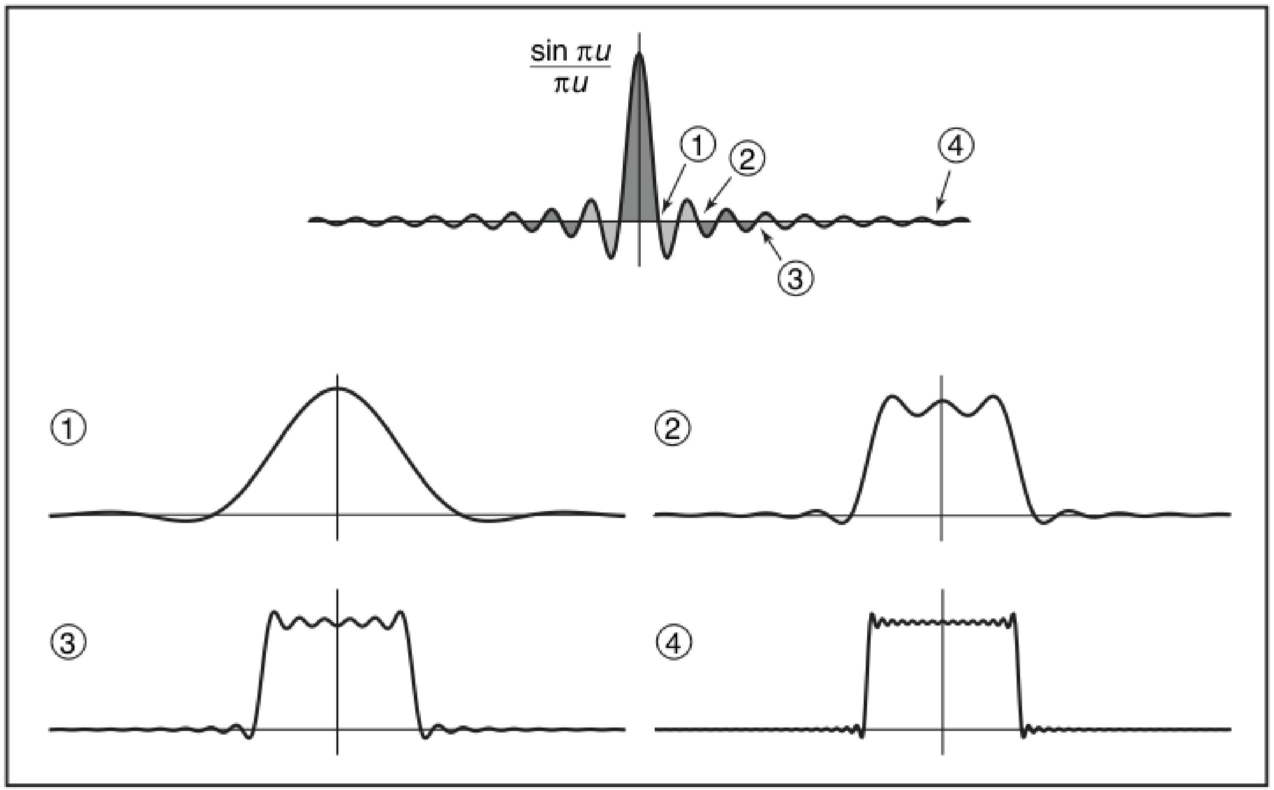
The Fourier transform

- Any function on the real line can be represented as an infinite sum of sine waves



[F]

The Fourier transform



The Fourier transform

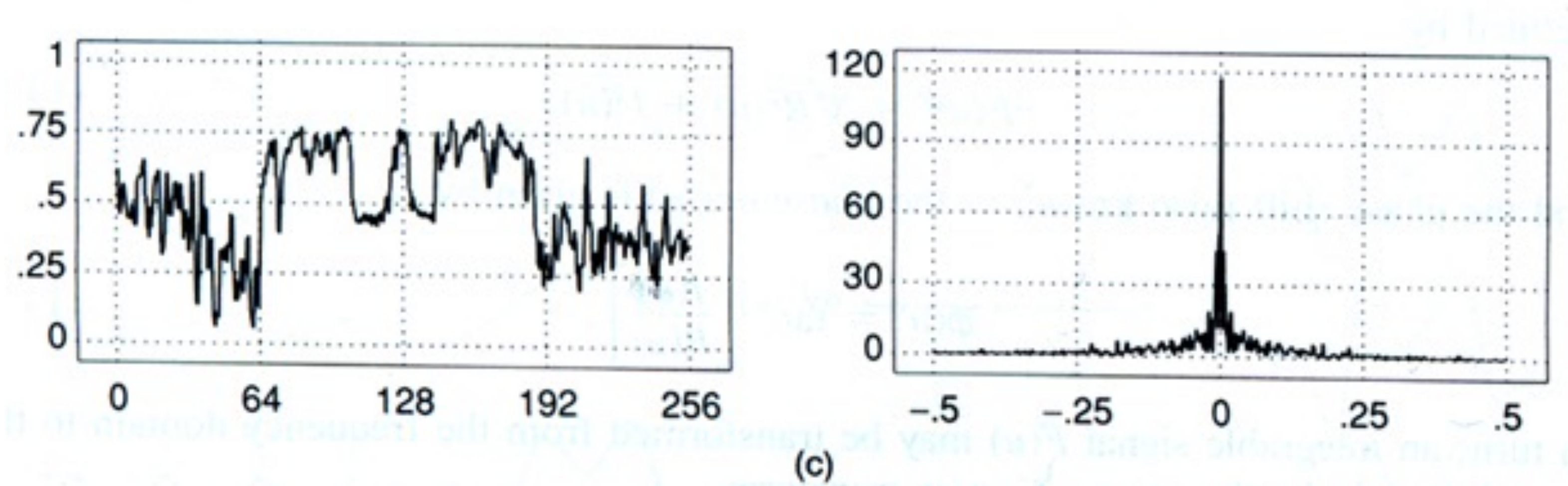
- The coefficients of those sine waves form a continuous function of frequency
- That function, which has the same datatype as the first one, is the Fourier transform.

$$F(u) = \int_{-\infty}^{\infty} f(x)(\cos 2\pi ux - i \sin 2\pi ux) dx$$

- Phase encoded in complex number

Fourier transform properties

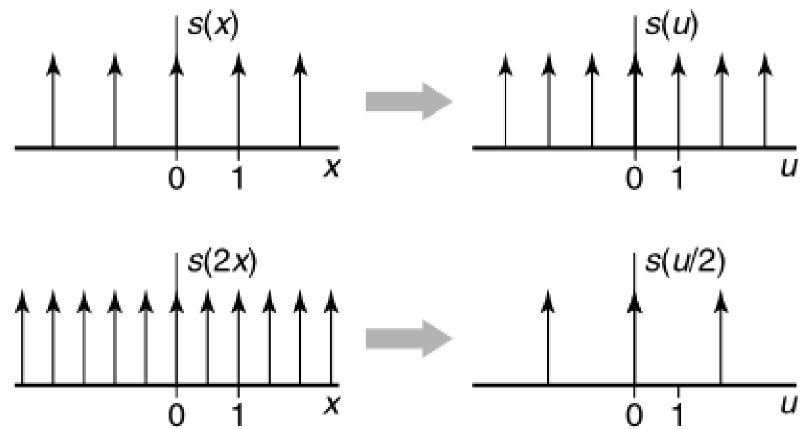
- F.T. is its own inverse (just about)
- Frequency space is a dual representation
  - amplitude known as "spectrum"



[FvDFH fig.14.15 / Wolberg]

Fourier pairs

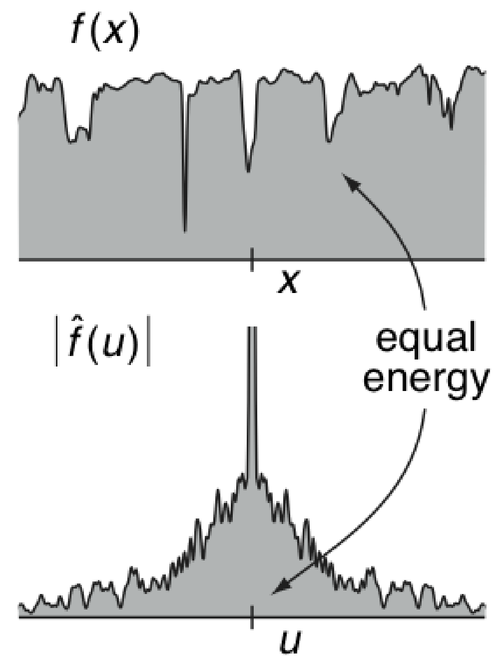
- sinusoid — impulse pair
- box — sinc
- tent — sinc^2
- bspline — sinc^4
- gaussian — gaussian
  (inv. width)
- imp. grid — imp. grid
  (1/d spacing)



[FvDFH fig.14.25 / Wolberg]

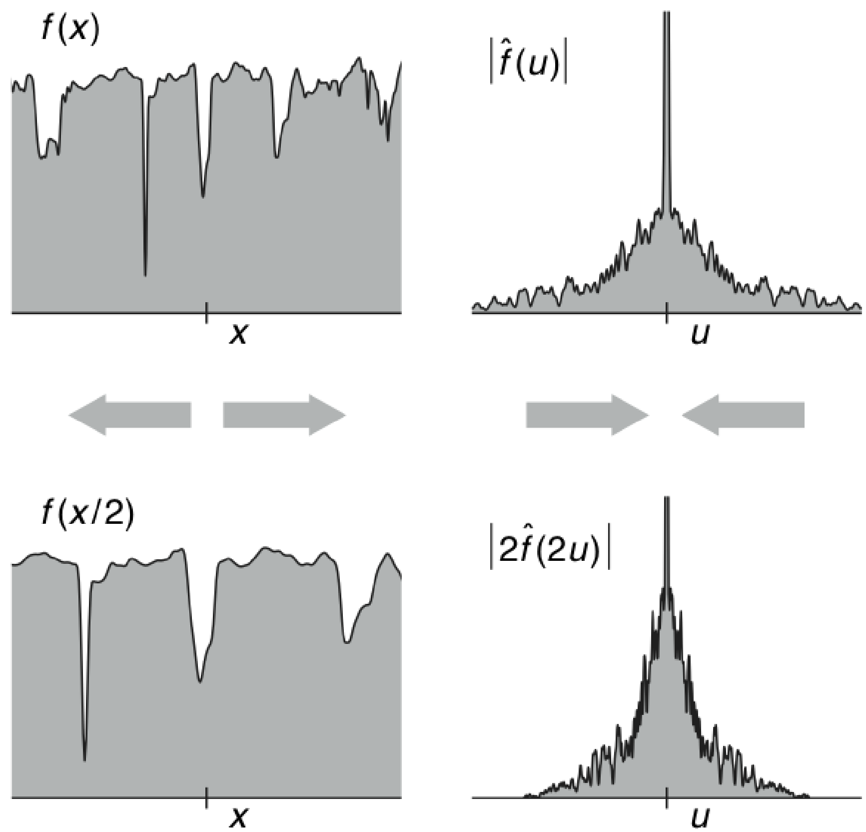
More Fourier facts

- F.T. preserves energy
  - That is, the squared integral
- DC component (average value)
  - It shows up at F(0)



More Fourier facts

- Dilation (stretching/squashing)
  - Results in inverse dilation in F.T.



Convolution and multiplication

- They are dual to one another under F.T.

$$\mathcal{F}\{f * g\}(u) = F(u)G(u)$$
$$\mathcal{F}\{fg\}(u) = (F * G)(u)$$

- Lowpass filters
  - Most of our "blurring" filters have most of their F.T. at low frequencies
  - Therefore they attenuate higher frequencies

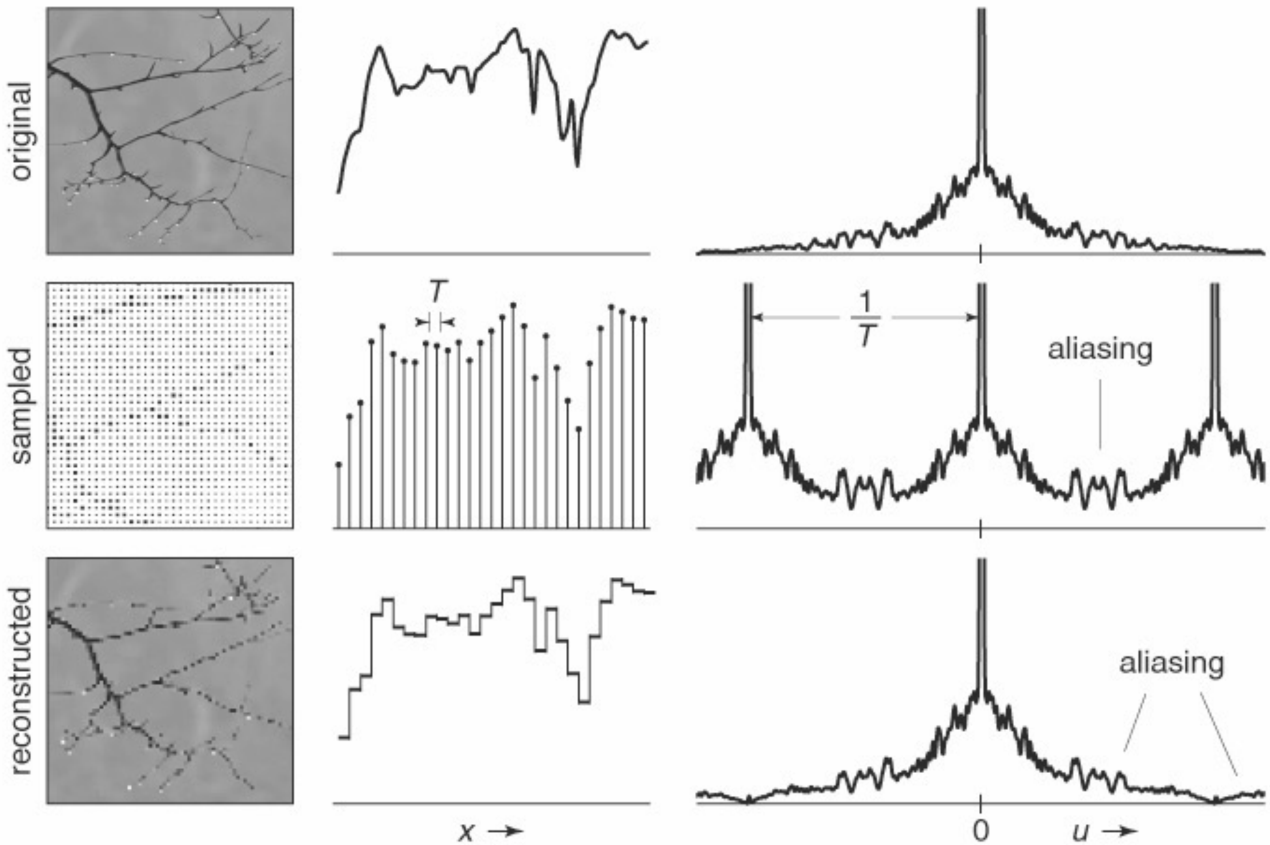
Checkpoint

- Formalized sampling and reconstruction
  - used impulses with multiplication and convolution
- Can talk about S&R with only one datatype
- Defined Fourier transform
  - alternate representation for functions
  - turns convolution, which seems hard, into multiplication, which is easy
- Destination: explaining how aliases leak into result

Sampling and reconstruction in F.T.

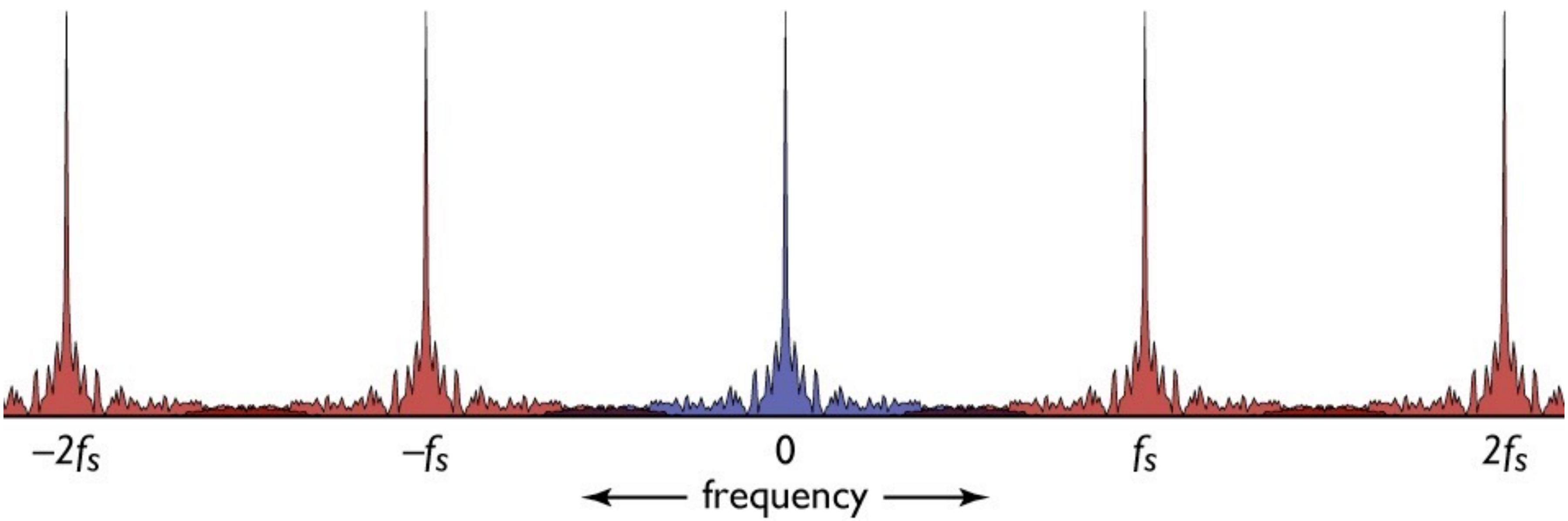
- Look at our sampling/reconstruction formulation in Fourier domain
  - Convolve with filter = remove high frequencies
  - Multiply by impulse grid = convolve with impulse grid
    - that is, make a bunch of copies
  - Convolve with filter = remove extra copies
  - Left with approximation of original
    - but filtered a couple of times

Aliasing in sampling/reconstruction



Aliasing in sampling

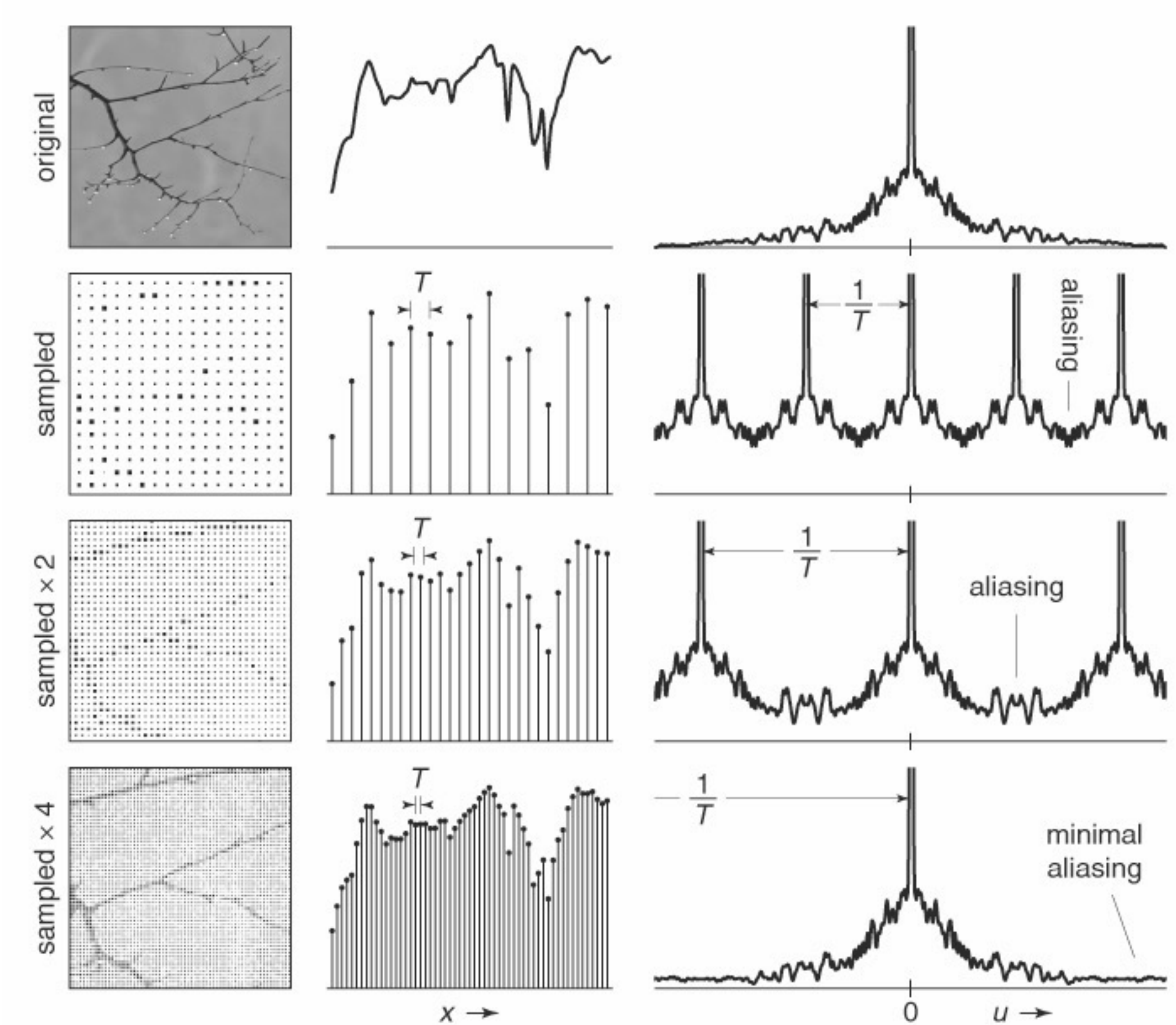
- If sampling filter is not adequate, spectra will overlap
- No way to fix once it's happened
  - can only use drastic reconstruction filter to eliminate
- Nyquist criterion



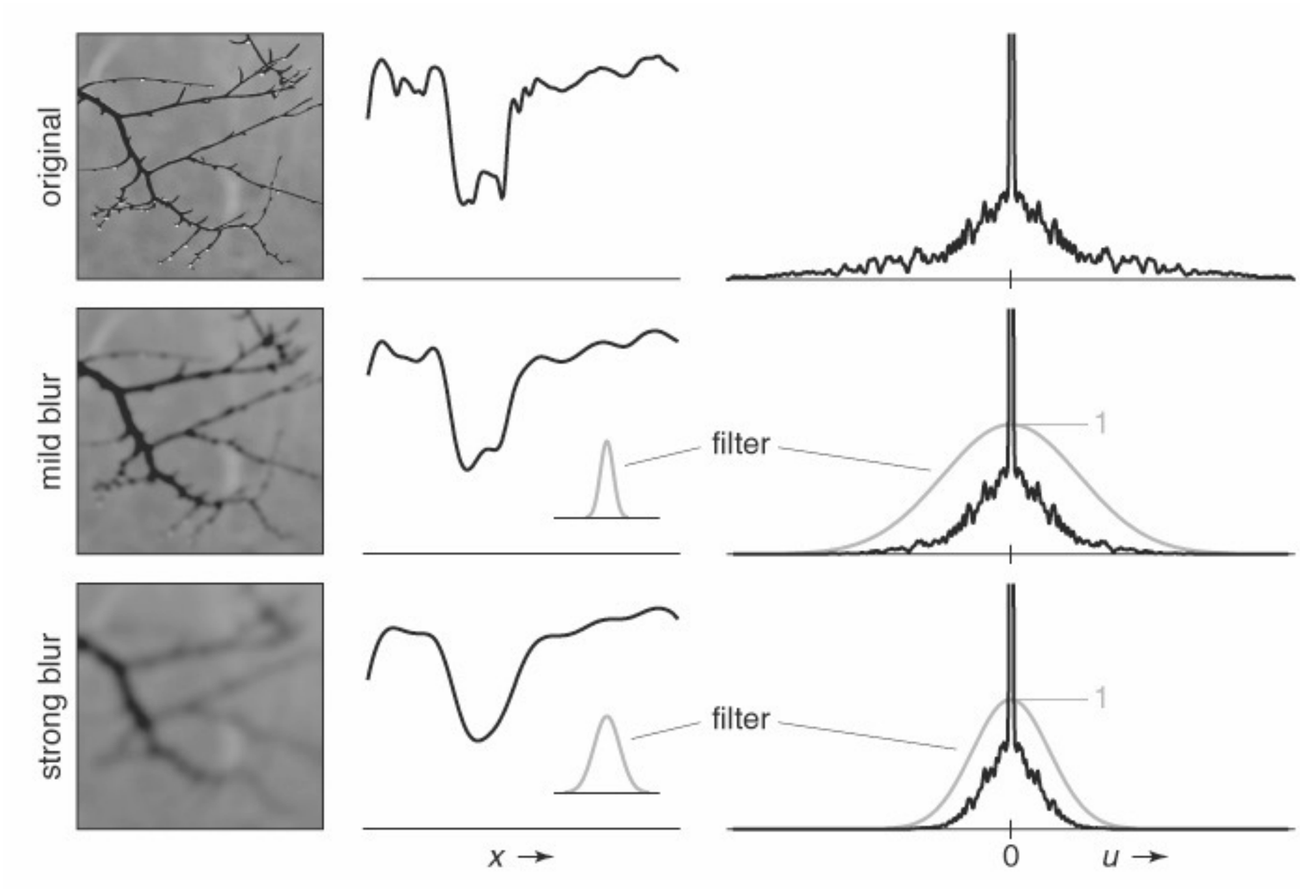
Preventing aliasing in sampling

- Use high enough sample frequency
  - works when signal is *band limited*
  - sample rate 2 * (highest freq.) is enough to capture all details
- Filter signal to remove high frequencies
  - make the signal band limited
  - remove frequencies above 0.5 * (sample freq.) (Nyquist)

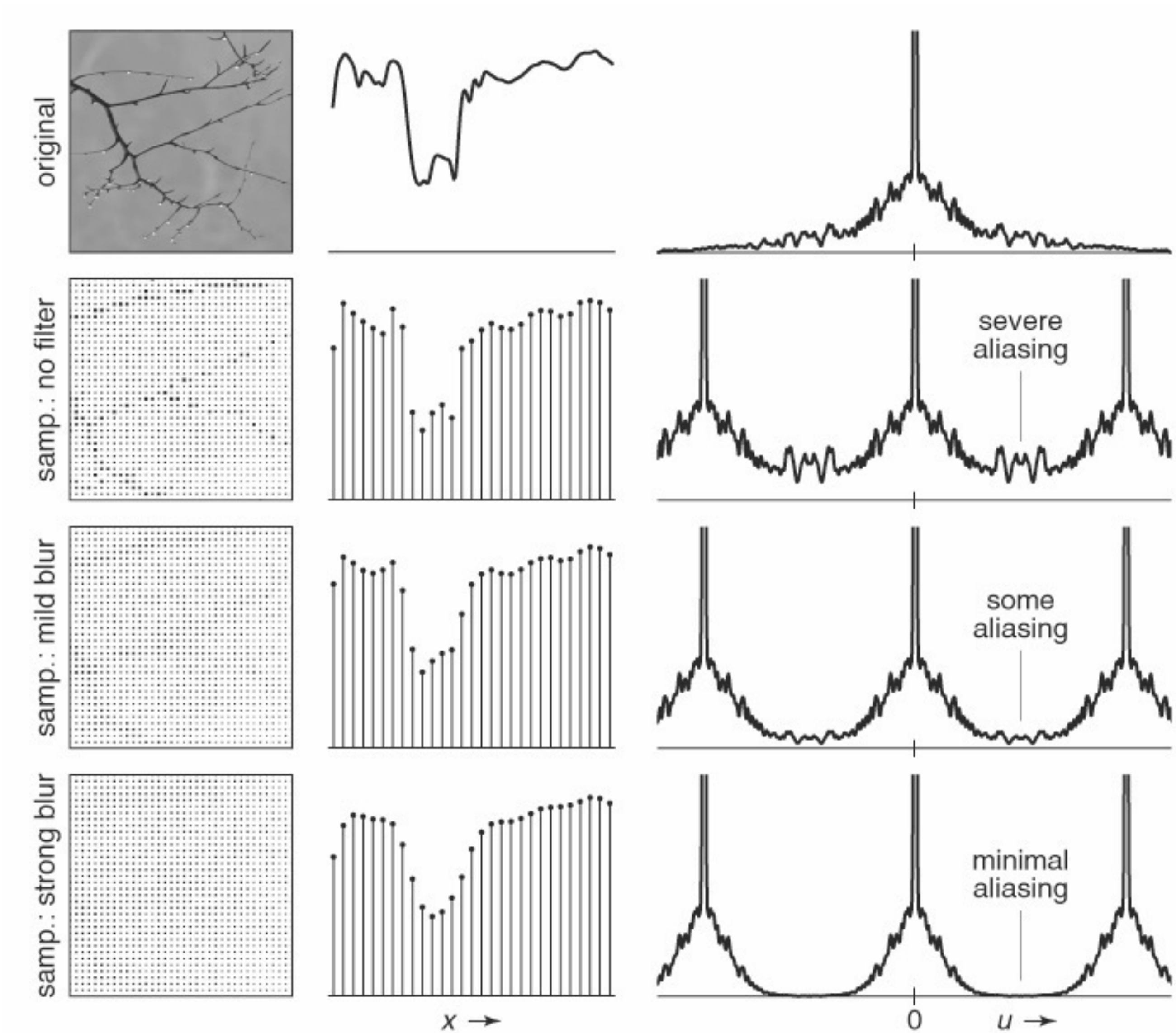
Effect of sample rate on aliasing



Smoothing (lowpass filtering)

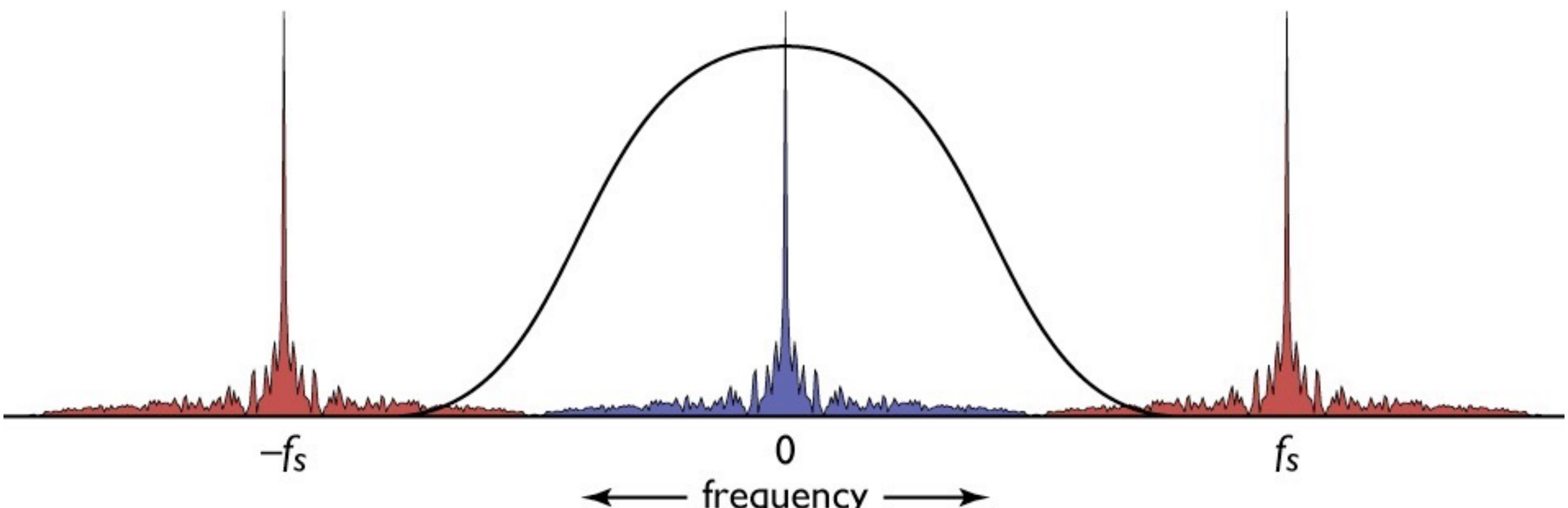


Effect of smoothing on aliasing

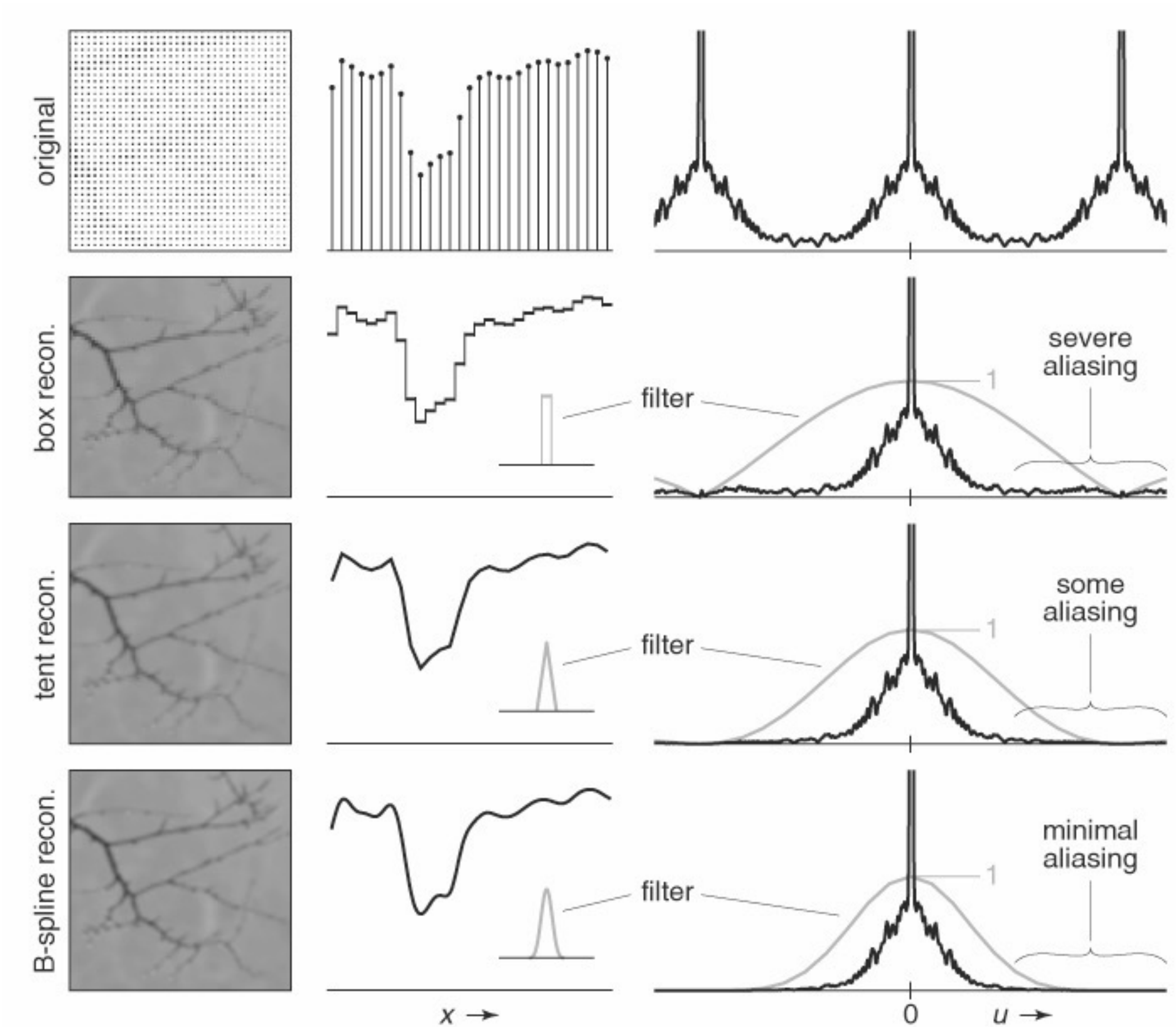


Aliasing in reconstruction

- If reconstruction filter is inadequate, will catch alias spectra
- Result: high frequency alias components
- Can happen even if sampling is ideal

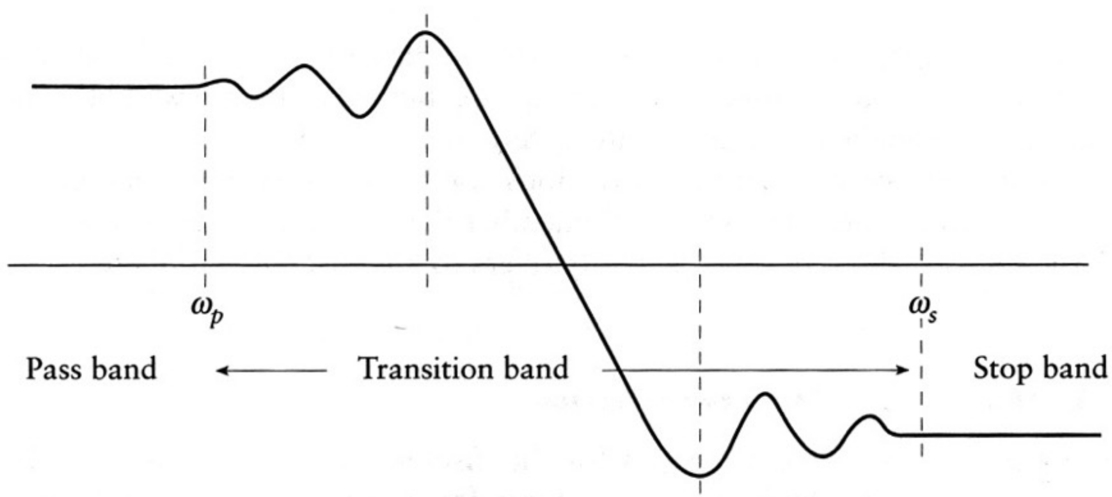


Reconstruction filters



Sampling filters

- "Ideal" is box filter in frequency
  - which is sinc function in space
- Finite support is desirable
  - compromises are necessary
- Filter design: passband, stopband, and in between



[Glassner fig.5-27]

Useful sampling filters

- Sampling theory gives criteria for choosing
- Box filter
  - sampling: unweighted area average
  - reconstruction: e.g. LCD
- Gaussian filter
  - sampling: gaussian-weighted area average
  - reconstruction: e.g. CRT
- Piecewise cubic
  - good small-support reconstruction filter
  - popular choice for high-quality resampling

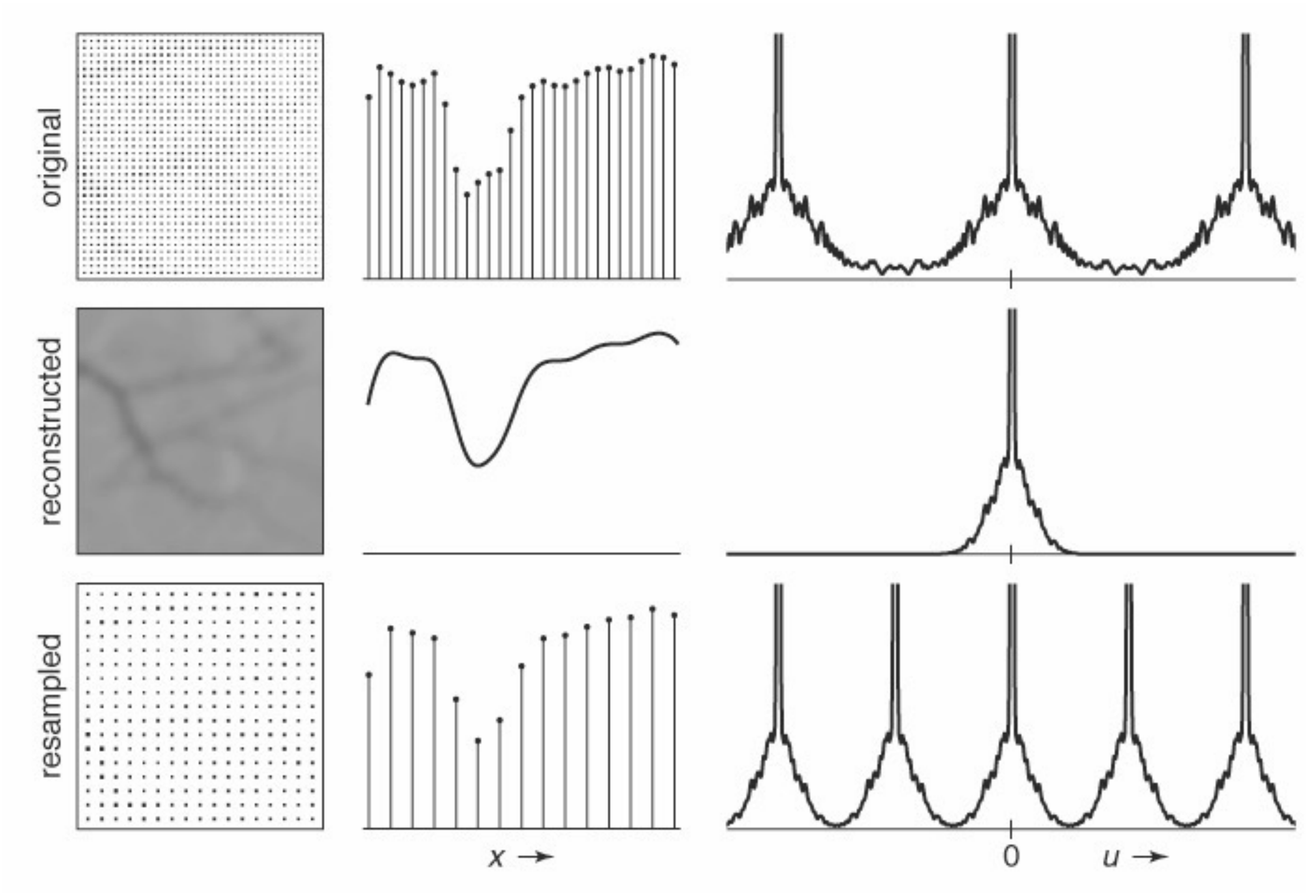
Resampling filters

- Resampling, logically, is two steps
  - first: reconstruct continuous signal
  - second: sample signal at the new sample rate
- Each step requires filtering
  - reconstruction filter
  - sampling filter
- This amounts to two successive convolutions
  - so regroup into one operation:

$$f_{\text{samp}} \star f_{\text{recon}} \star g = (f_{\text{samp}} \star f_{\text{recon}}) \star g$$

  - single filter both reconstructs and antialiases

Resampling in frequency space



Sizing reconstruction filters

- Has to perform as a reconstruction filter
  - has to be at least big enough relative to input grid
- Has to perform as a sampling filter
  - has to be at least big enough relative to output grid
- Result: filter size is max of two grid spacings
  - upsampling (enlargement): determined by input
  - downsampling (reduction): determined by output
  - for intuition think of extreme case (10x larger or smaller)

Summary

- Want to explain aliasing and answer questions about how to avoid it
- Formalized sampling and reconstruction using impulse grids and convolution
- Fourier transform gives insight into what happens when we sample
- Nyquist criterion tells us what kind of filters to use

Supersampling

- When we can't have a bandlimited signal we can improve matters by taking several samples per pixel
  - think of this as an estimate of the convolution integral
- Regular sampling is a simple quadrature rule:

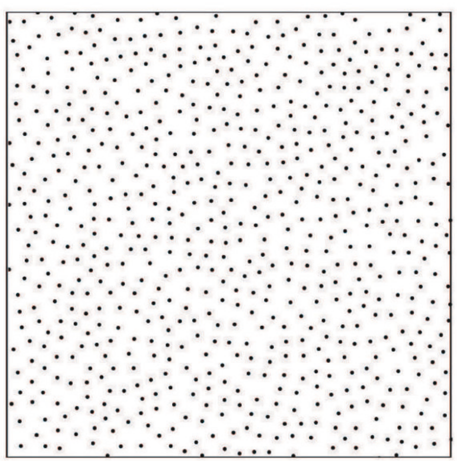
$$I[i,j] = \sum_k I(x_k, y_k) A_k \approx \int I(x,y)\,dA$$

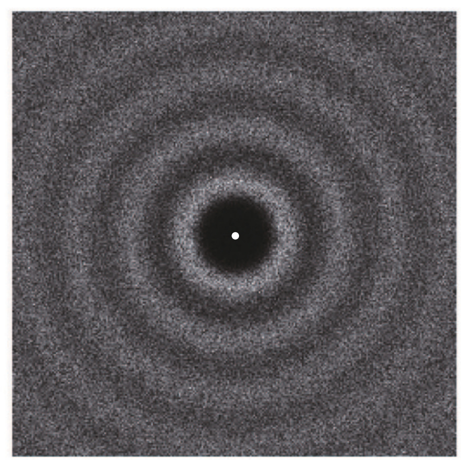
- Irregular sampling can be seen as a Monte Carlo estimate:

$$I[i,j] = \sum_k I(x_k, y_k) \approx \int I(x,y) p(x,y)\,dx\,dy$$

Regular supersampling in FT

- Really, we are first sampling at a higher rate, then convolving with the sampling filter
  - regular supersampling pushes the alias spectra farther away from the main spectrum
  - the signal we are sampling still contains regular spikes, though
- Irregular sampling patterns have a different kind of FT





Dong-Ming Yan et al. *J Comp. Sci. Tech.* 2015