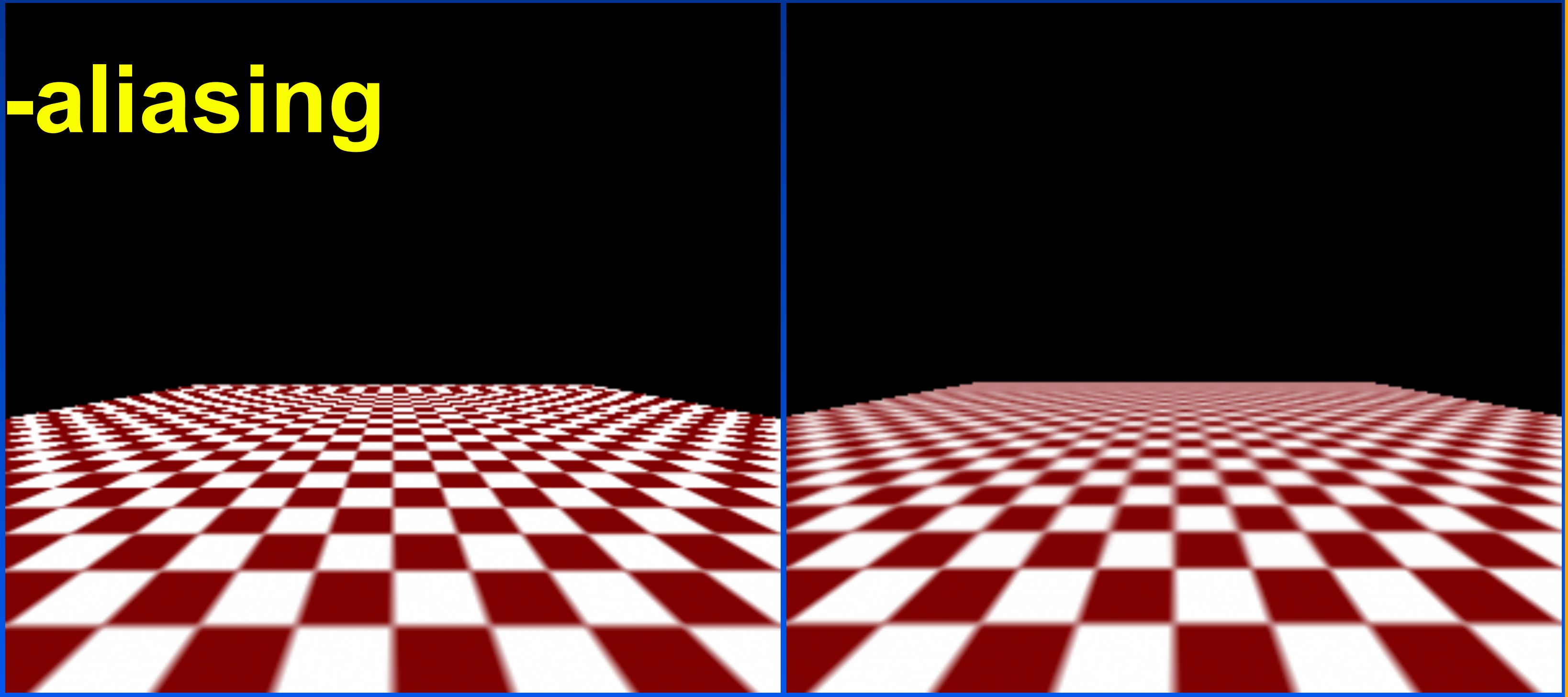
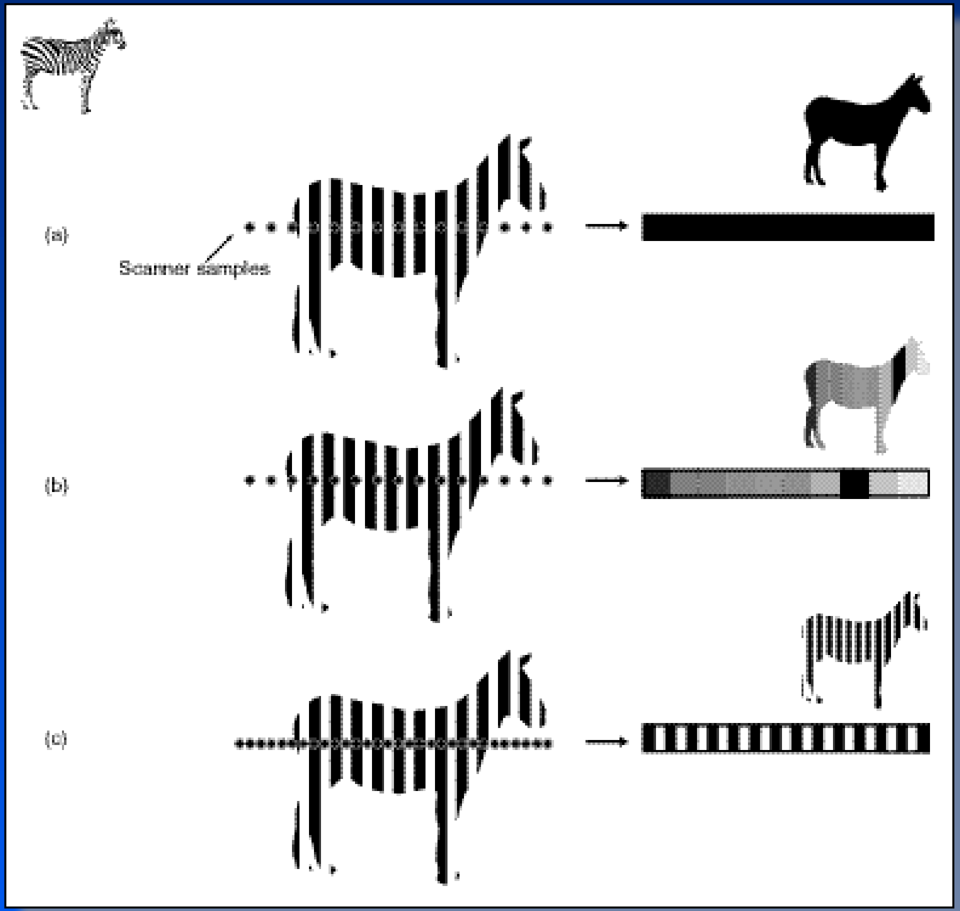
Anti-aliasing



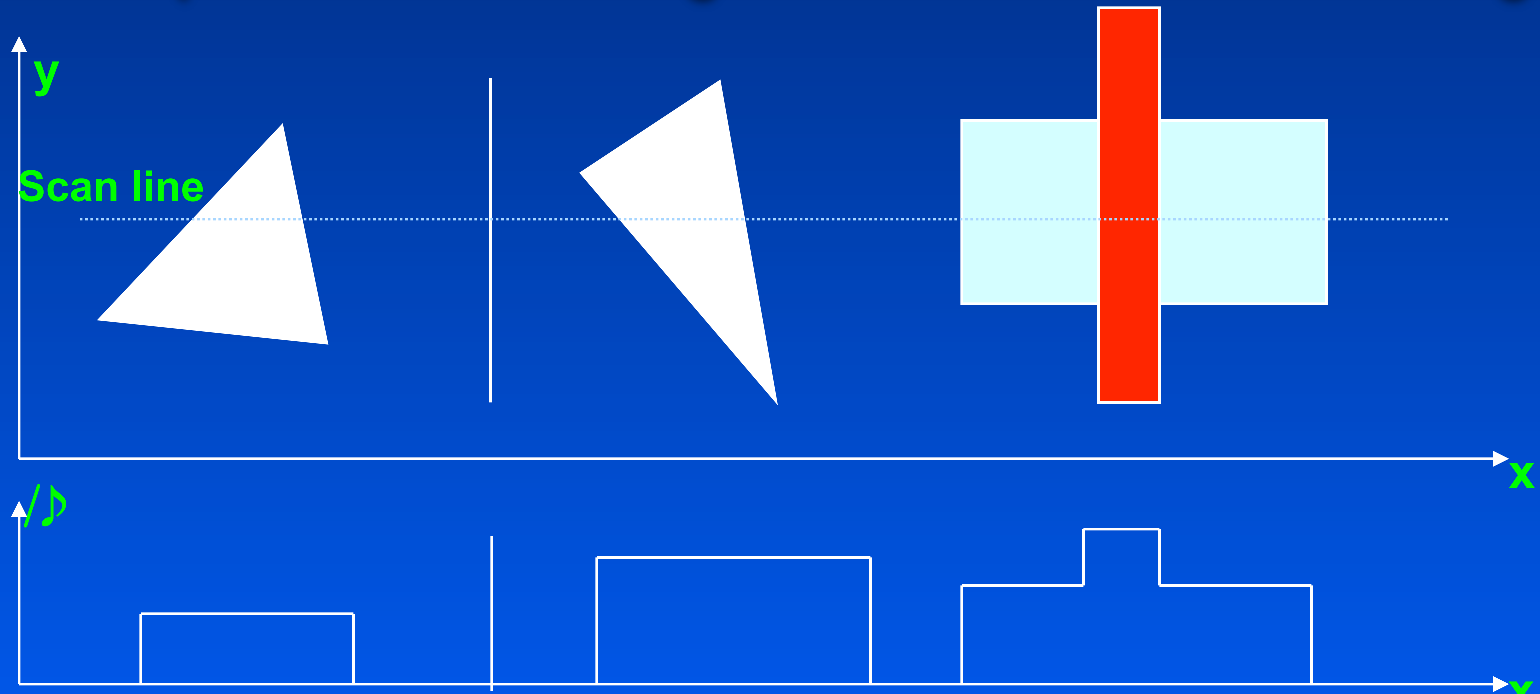

(a)
Scanner samples
(b)
(c)

Basic Problem with Digital Representation

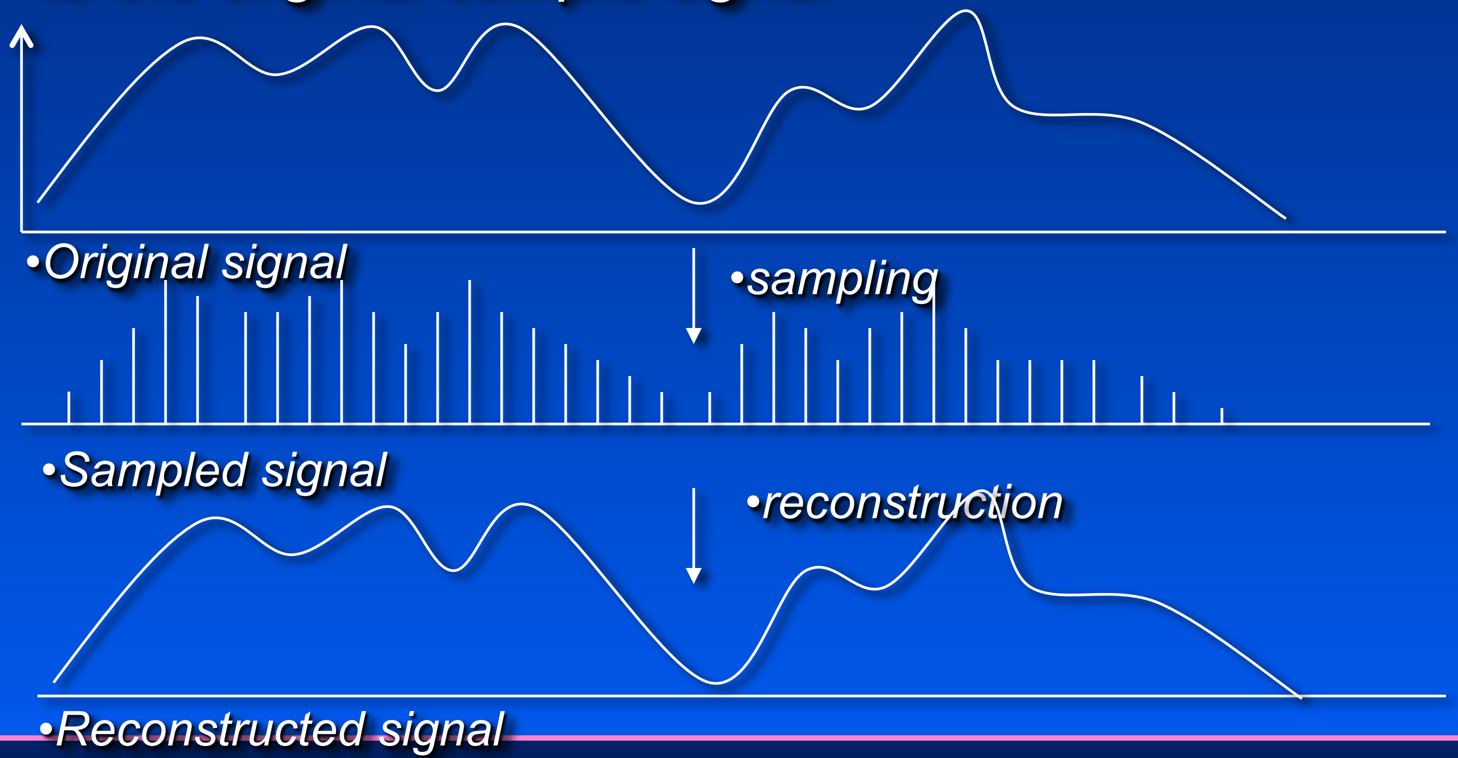
Continuous Signal $\xrightarrow{\text{Sampled}}$ Discrete set of samples $\xrightarrow{\text{Reconstruction}}$ Continuous Signal

- How to sample and reconstruct so that we end up with what we started with?

- Before scan-conversion, projection of 3-D objects onto viewplane can be thought of as a continuous 2-D signal

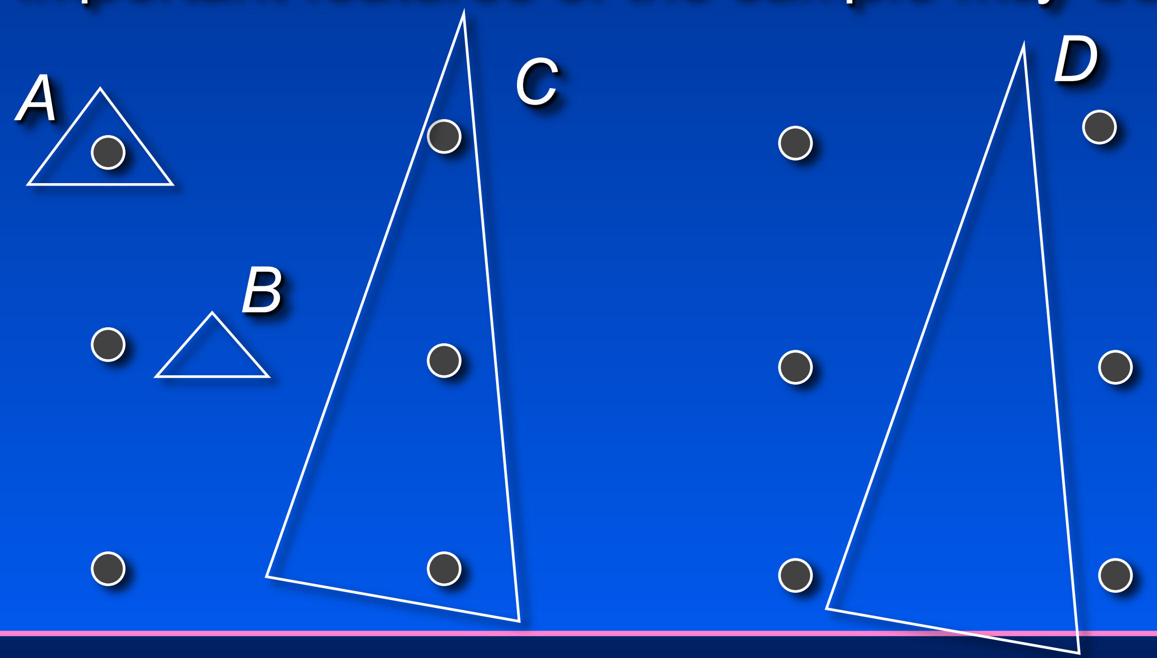


- Goal is to reconstruct a signal which is as close as possible to the original sample signal



Intuitive Approaches: Point Sampling

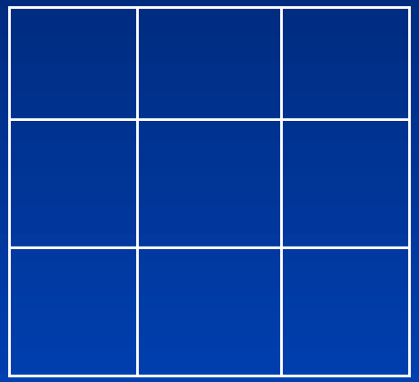
- Select one point for each pixel; evaluate the original signal at this point; assign value to the pixel
- Important features of the sample may be missed:



Supersampling

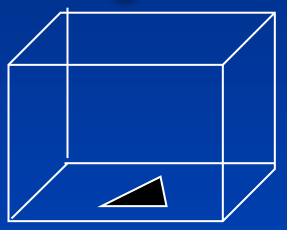
- Increase sampling rate
- Take multiple adjacent samples and average their values to determine a value for a single pixel
- Popular in graphics; easy and often achieves good results despite increase in computation
- However, sometimes minimum sampling rate must be infinitely high!

One pixel

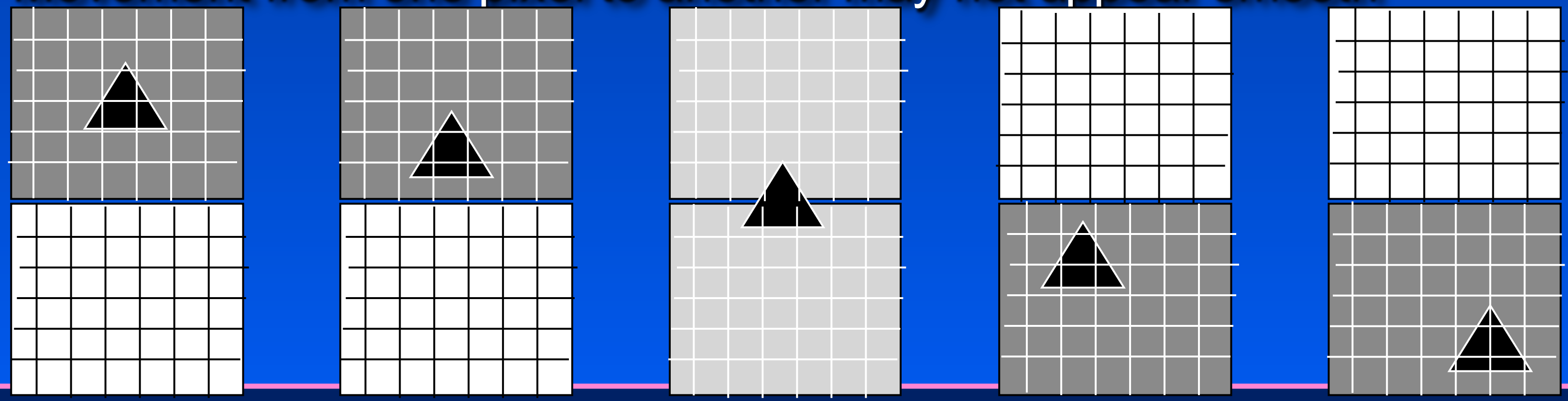


Uniform Weighted Area Sampling

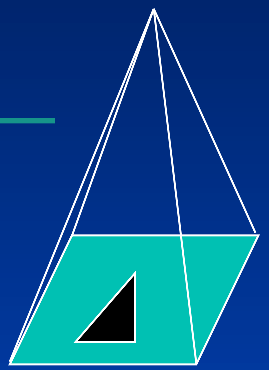
- Instead of point sample, integrate the signal over the area of a square centered about each grid point and selecting the average intensity as that of the pixel



- Often called a box-filter: weight is a box
- No regard to location of object in pixel
- Movement from one pixel to another may not appear smooth

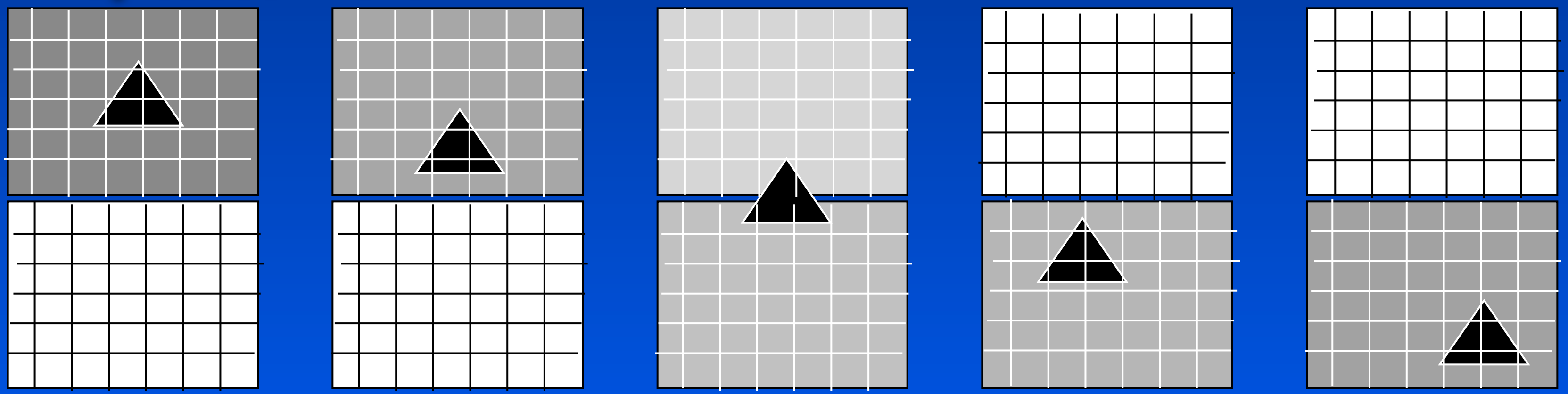


Non-Uniform Weighted Area sampling



- Assign different weights to different parts of the pixel
  - For example using a pyramid or cone-shaped "filter"
- Object still contributes only to the specific pixel containing it

  May cause "flicker"



- Fix this by allowing weighting functions to overlap

Sampling in the Frequency Domain: Fourier Transform

- Any signal *f(x)* spectrum of sine waves

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

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

- Inverse

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{-i2\pi ux} du$$

u – frequency domain

x – space domain

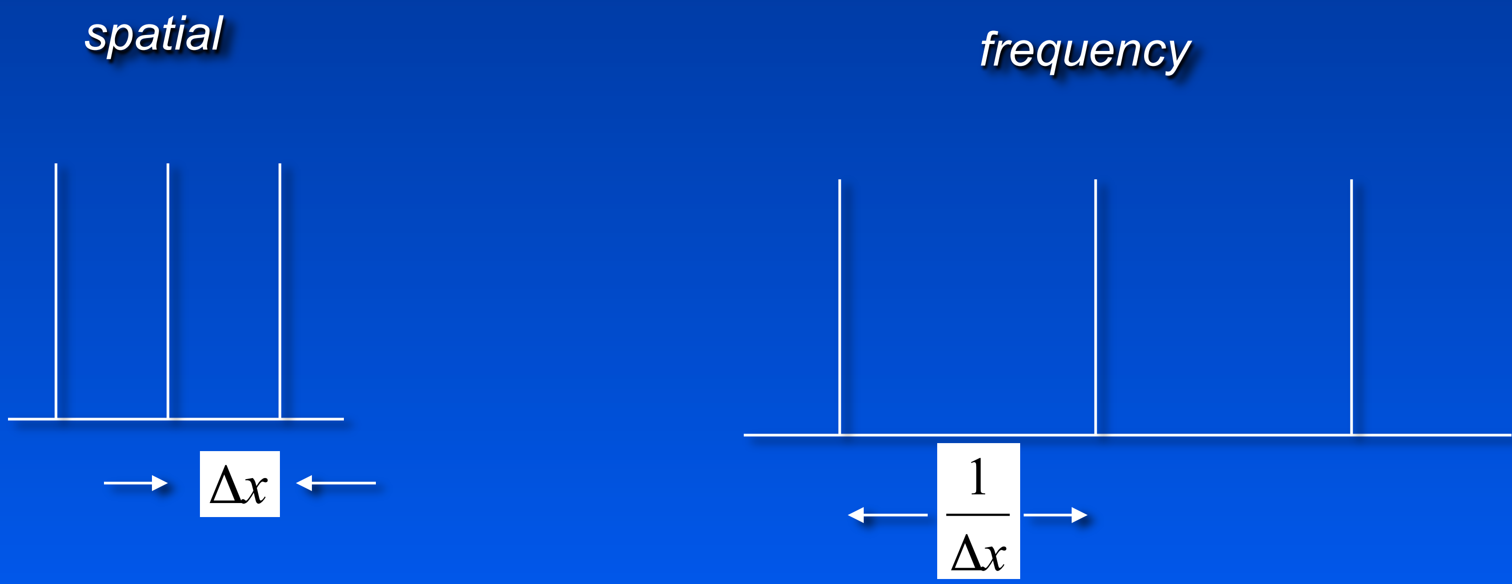
$$F(u) = R(u) + iI(u)$$

$$\text{amplitude}: |F(u)| = \sqrt{R^2(u) + I^2(u)}$$

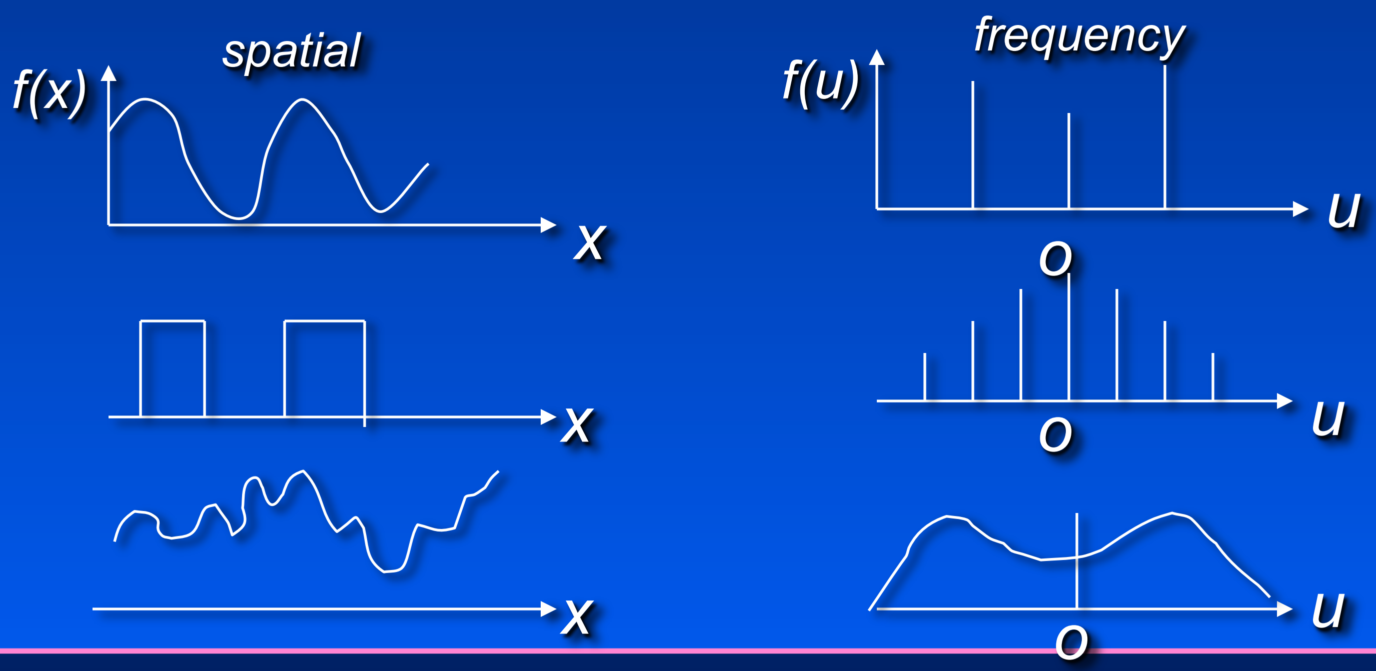
$$\text{phase}: \phi(\text{u}) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right)$$

- Adding a number of sinusoids with amplitude $F(u)$ and phase $\phi(u)$ gives $f(x)$
- Fourier transform of a signal often plotted as magnitude vs. frequency, ignoring phase angle

- Fourier transform of a Dirac comb function is another comb function



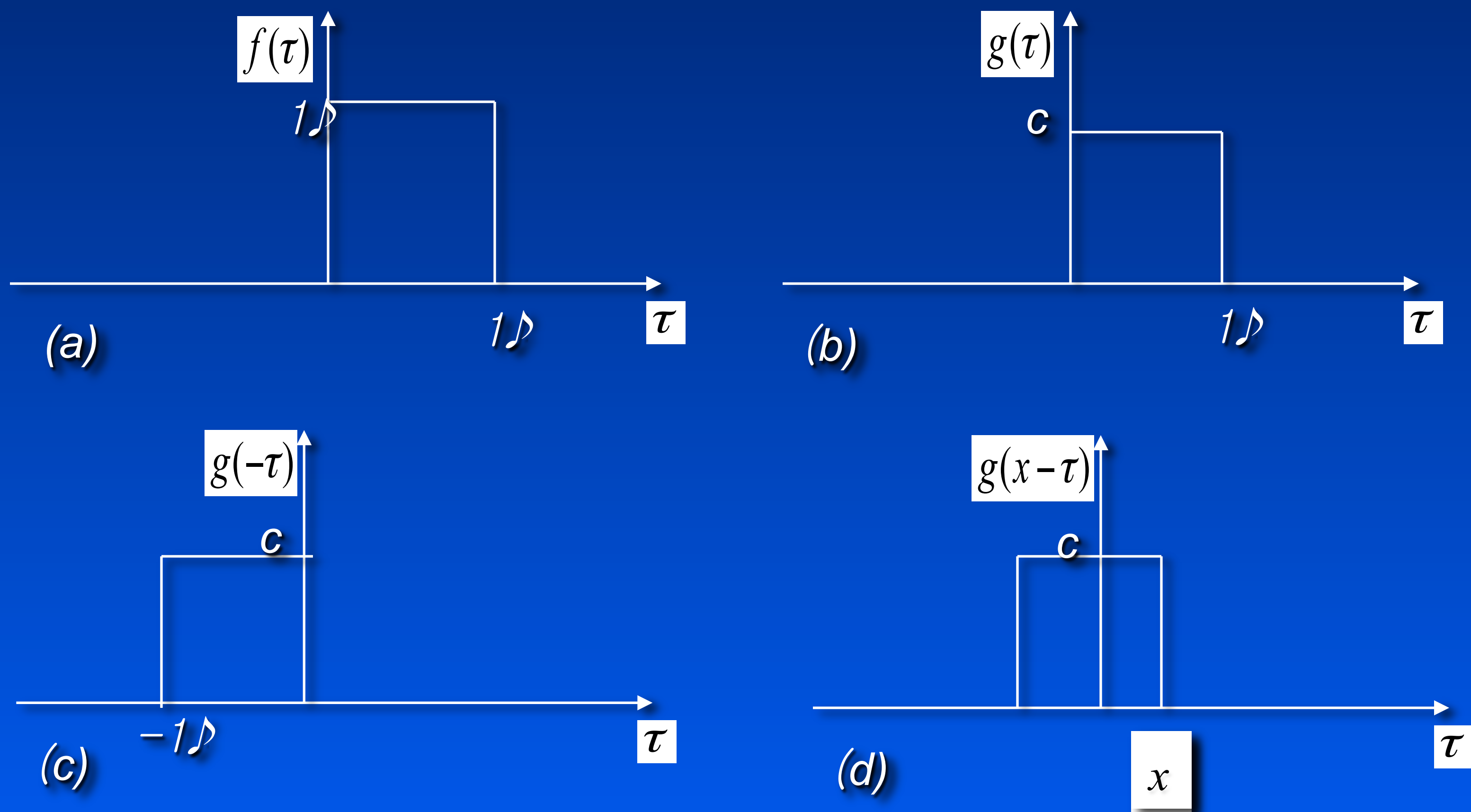
- Period function has a discrete frequency spectrum
- Aperiodic function has continuous frequency spectrum

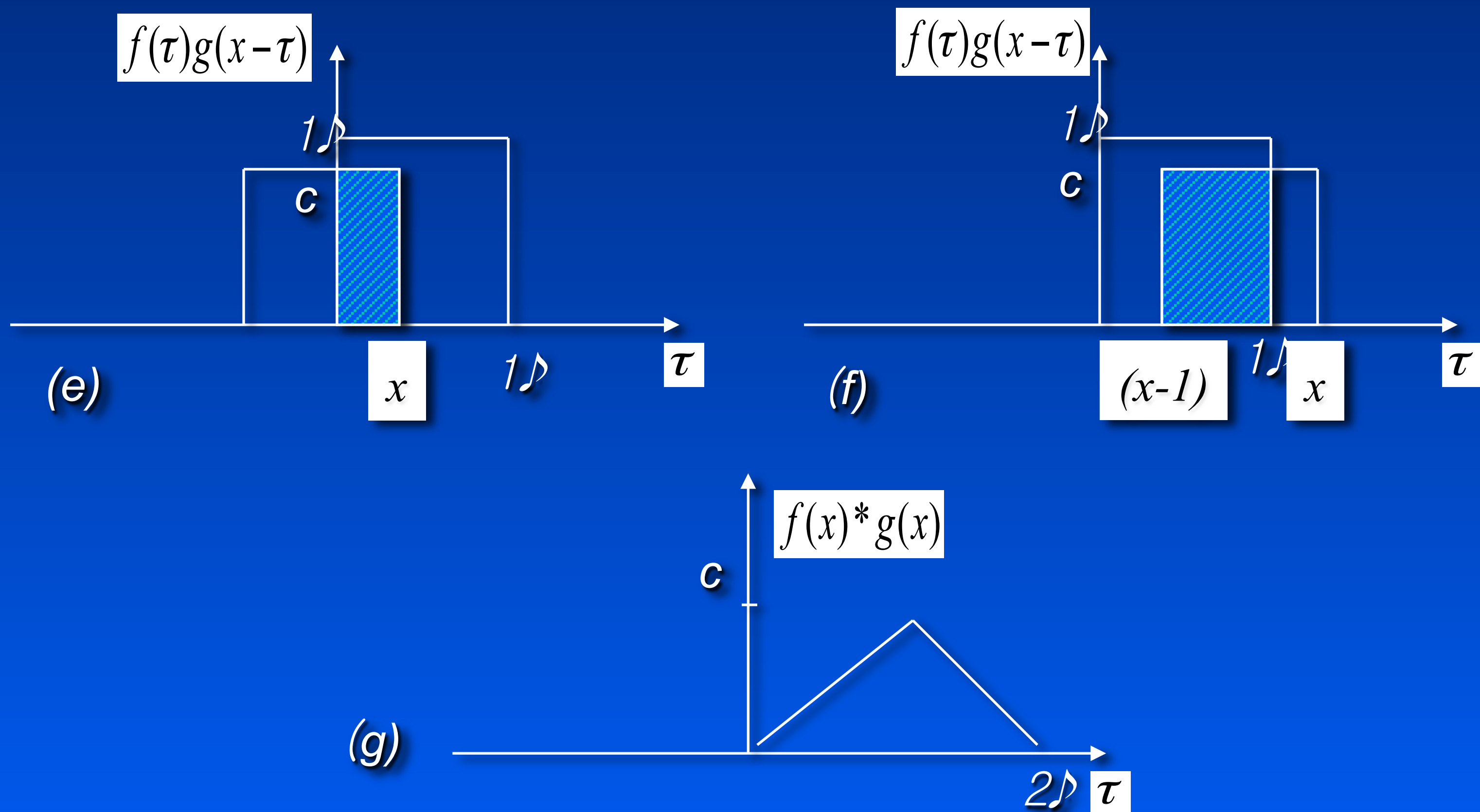


Convolution

- Multiplying in the frequency domain is the same as performing convolution in the spatial domain (and vice versa)
- Convolution of two signals $f(x)$ and $g(x)$ is

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(t)g(x-t)dt$$

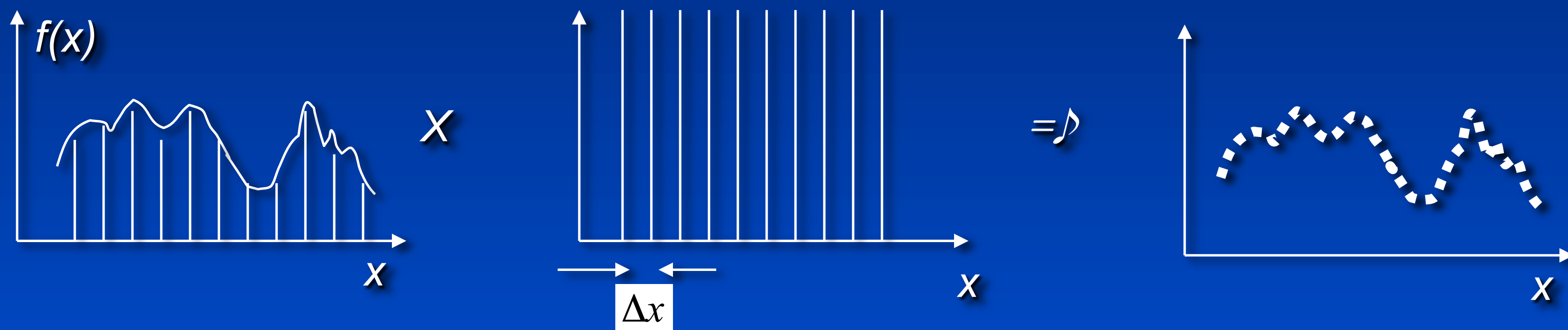

f(τ)
1♪
1♪
τ
(a)
g(τ)
c
1♪
τ
(b)
g(−τ)
c
−1♪
τ
(c)
g(x−τ)
c
x
τ
(d)



$f(\tau)g(x-\tau)$
1♪
c
(e)
x
1♪
τ
$f(\tau)g(x-\tau)$
1♪
c
(f)
(x-1)
1♪
x
τ
$f(x)*g(x)$
c
(g)
2♪
τ

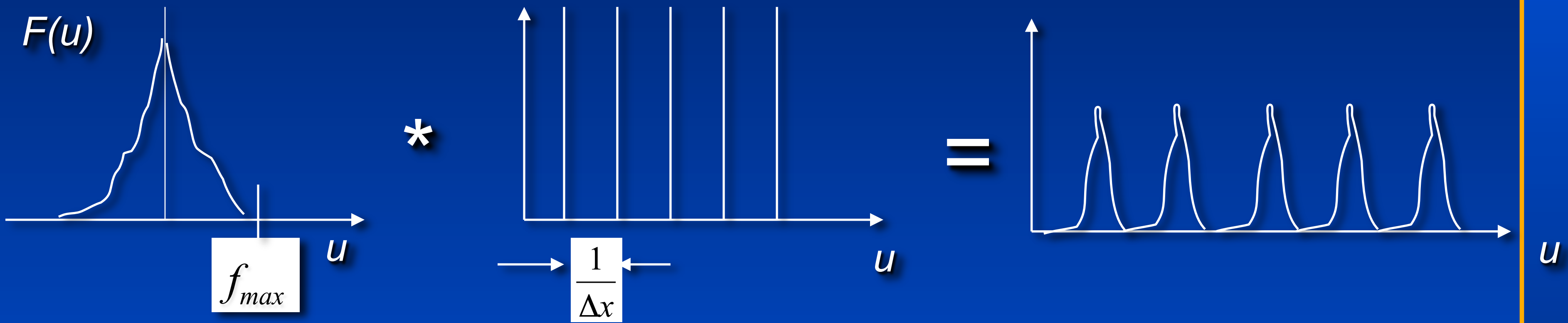
Sampling in Spatial Domain

- Multiply original signal w/ comb function



- This is convolution in frequency domain by Fourier transform of comb
  - which is another comb with spacing $1/\Delta x$

Sampling in Frequency Domain



- No overlap if $2f_{\max} < \frac{1}{\Delta x} \Rightarrow 2f_{\max} < f_s$

  f_s is sampling frequency

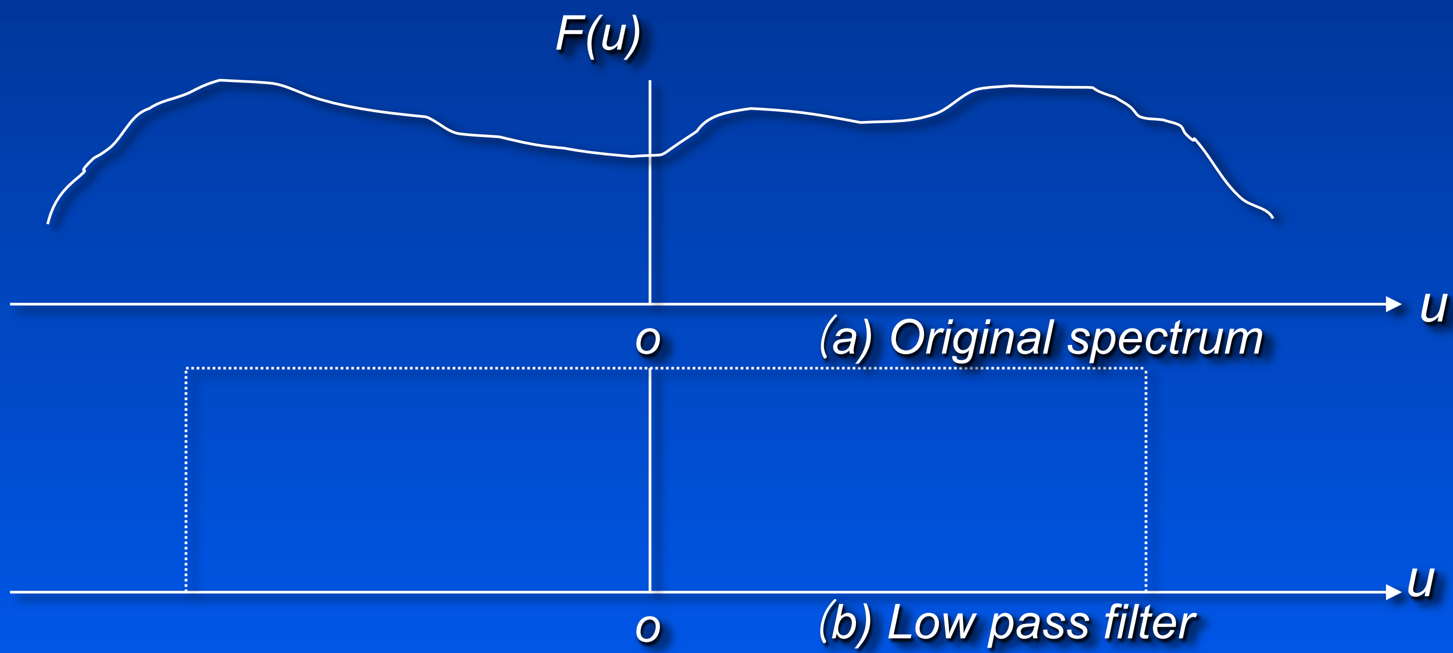
Shannon's theorem

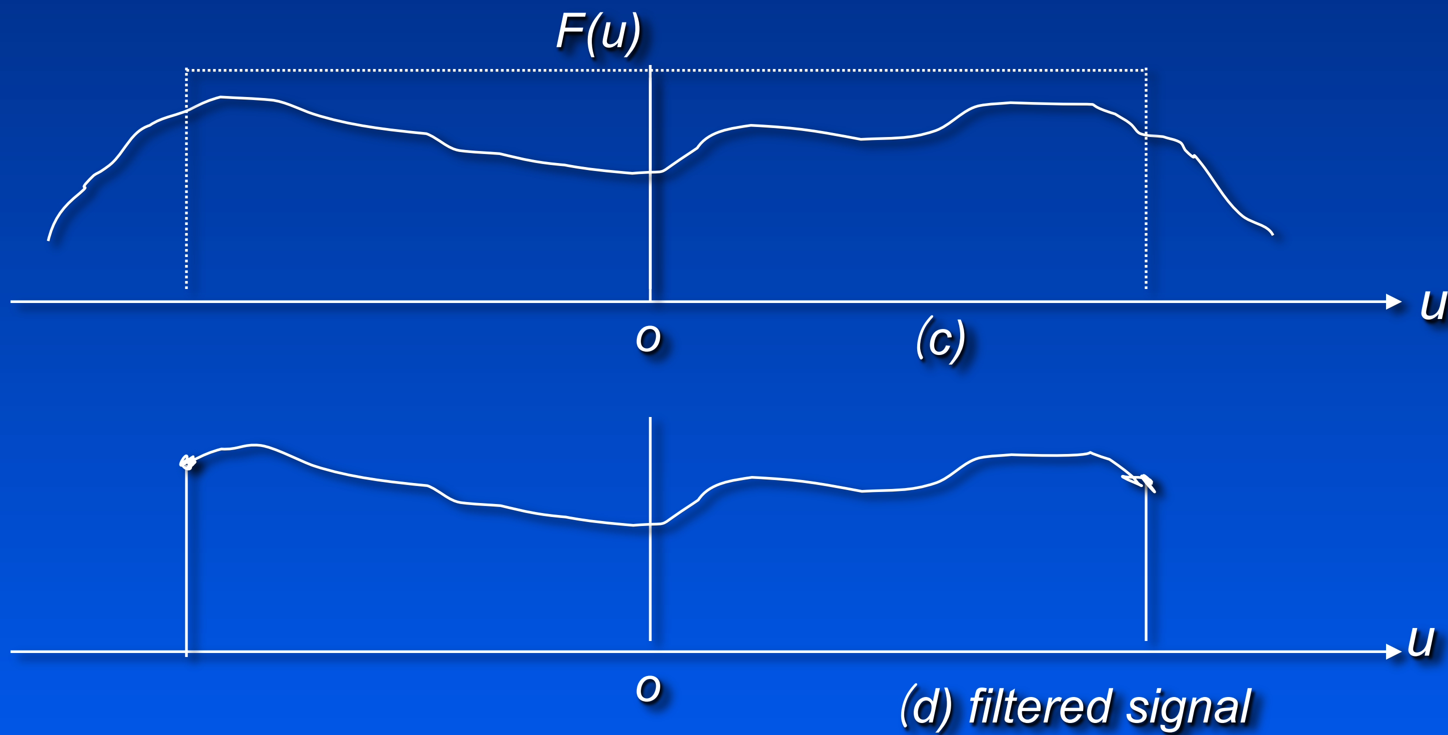
- Signal can be properly reconstructed from its samples if original signal is sampled at frequency that is at least twice the highest frequency component in its spectrum
- Lower bound on sample rate known as the Nyquist frequency
- Sampling below the Nyquist frequency can produce what could have been obtained from sampling a lower frequency signal
- The low frequency and high frequency signals are *aliases*

Filtering

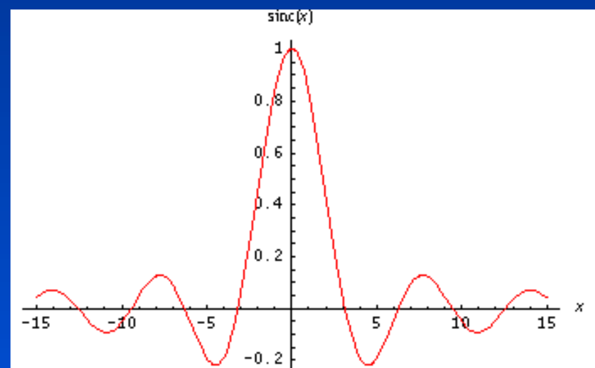
- Previous problems was high-frequency components masquerading as low-frequency components in the reconstructed signal
- Solution : remove the high frequencies from the original signal
  - this is known as band-limiting or low-pass filtering
  - tradeoff aliasing vs blurring

- In the frequency domain, multiply the signal by a pulse function to truncate all frequencies above the band:

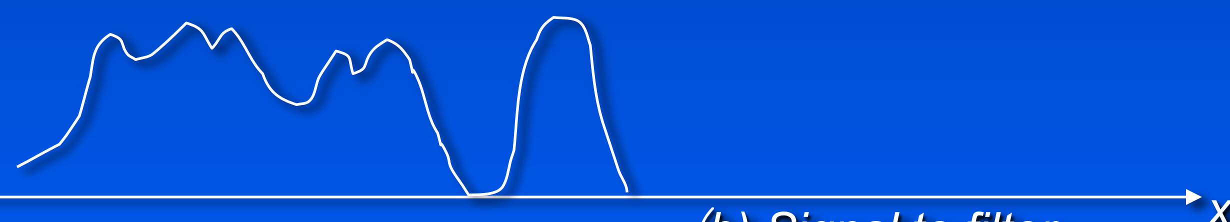



F(u)
u
o
(c)
u
o
(d) filtered signal

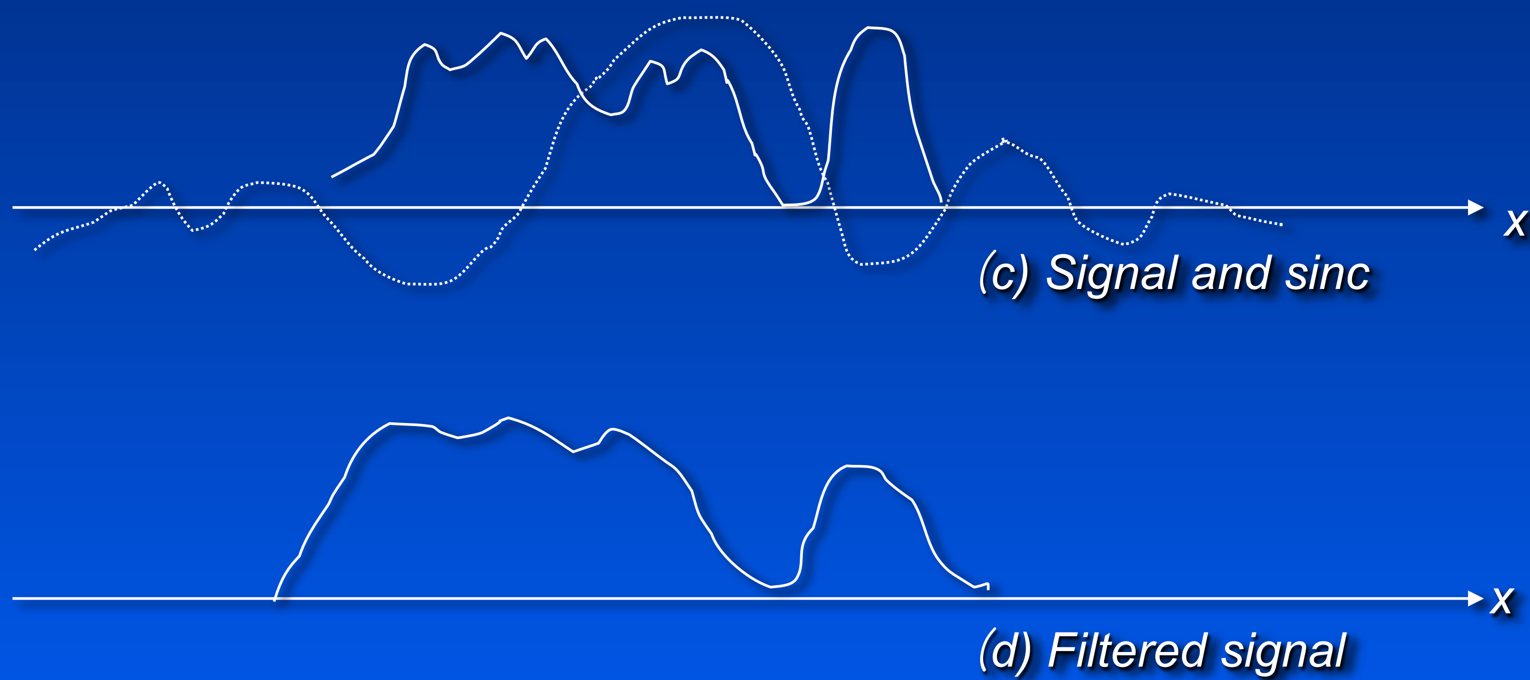
- In the spatial domain, convolve (*) the signal with a sinc function: $\sin(\pi x)/\pi x$



(a) sinc



(b) Signal to filter


x
(c) Signal and sinc
x
(d) Filtered signal

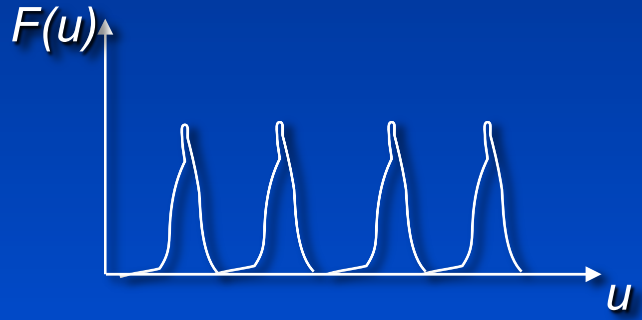
- Problems with the sinc function
- Infinitely wide
- If a truncated version is used, undesirable ringing, or *Gibbs* phenomenon occurs:
  - Has parts that dip below zero, which may result in signal dipping below zero
  - For signals that represent intensity, these must be clamped to zero

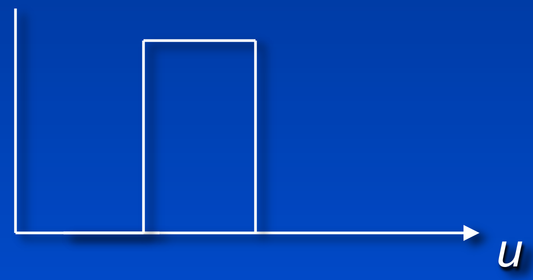
- Other filters are used instead
- Finite impulse response (FIR) filters, in contrast to infinite sinc filter
  - Tent
  - Gaussian
  - Box filter (in frequency domain, it is sync)

- So far: convolve signal with a filter and then sample the filtered signal
- Wasted work sampling the filtered signal anywhere but at the sample points
- Thus, only evaluate convolution integral at each sample point
- This is weighted area sampling!

Reconstruction in Frequency Domain

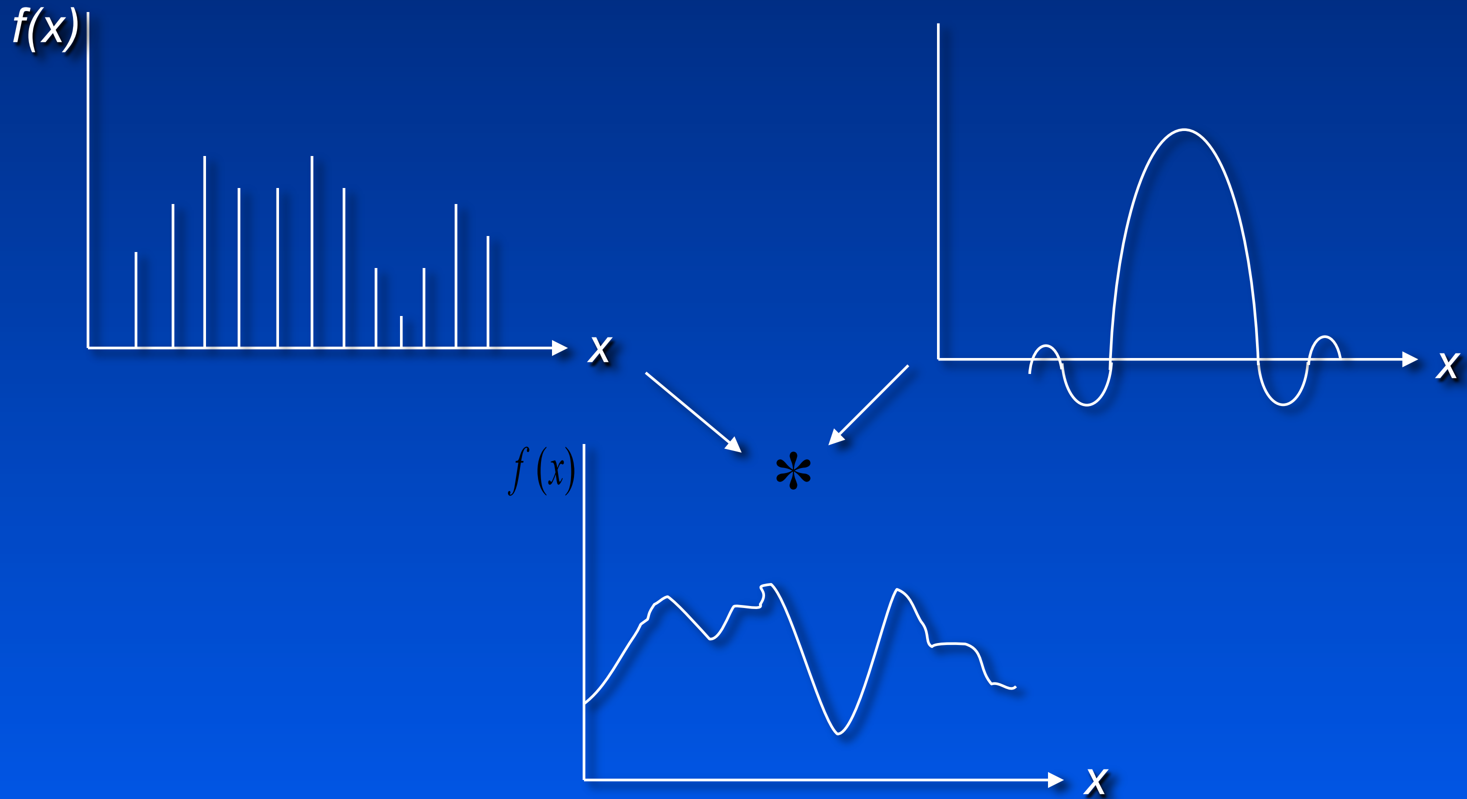
- Multiply signal by pulse function in frequency domain to eliminate replicated spectrum





- Convolve in space domain by sinc

Reconstruction in spatial domain



Reconstruction

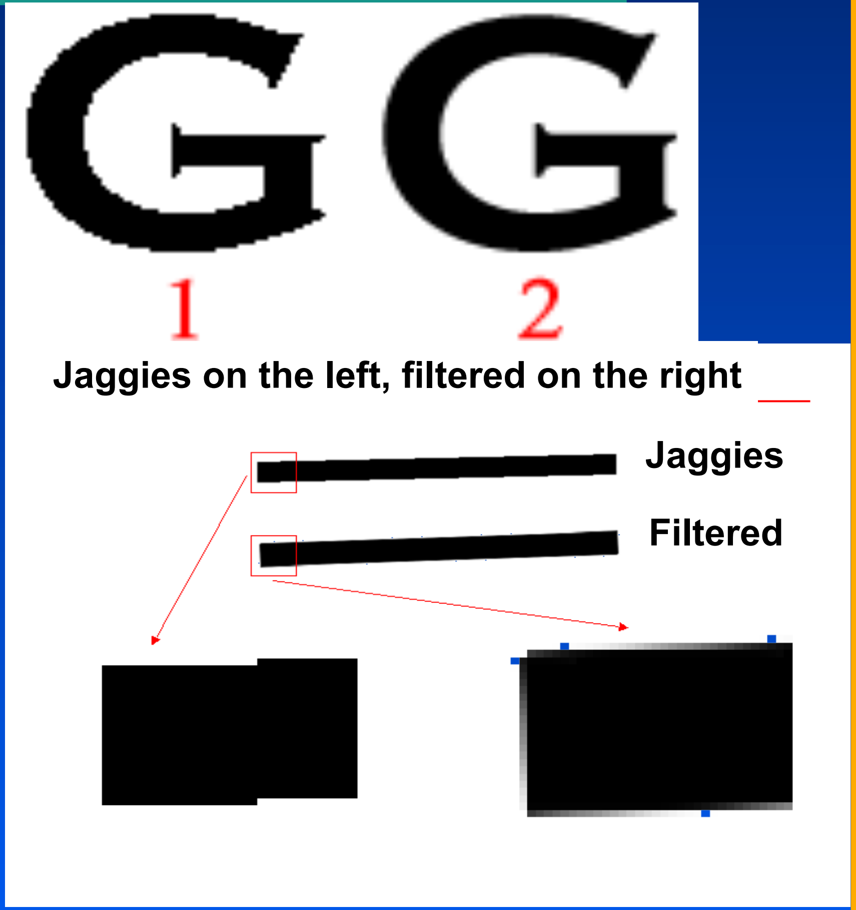
- Recreation of the original signal from its samples
- Samples in frame buffer must be turned into a continuous video signal
  - Sample-and-hold process : signal is reconstructed by holding the value of each successive sample for the duration of a pixel

Sample-and-hold

- Results in sharp transitions between pixels known as rastering (stair steps)
- Effect reduced because CRT/LED/LCD does not produce sharp discontinuous jumps in intensity between pixels
- Effect increased when pixel-replicating zoom is used, or in printing, or digital film recordings
- This stair stepping effect (or the jaggies) is not really aliasing
  - It is not high frequency signals masquerading as low frequency signals

The Infamous Antialiasing and the "Jaggies"

- "Jaggies" are NOT aliases of high-frequency signal masquerading as low frequency signal!
- The high frequencies are still there but in a different direction (in x and y direction as opposed to arbitrary direction)



- The process of filtering reduce high frequency components
- Confusion is that the way to reduce aliasing (filtering) also reduces the “jaggies”
- “Jaggies” are not examples of aliasing artifacts!

Supersampling

- Higher f_s, smaller Δx, greater $1/\Delta x$
  i.e. pushes copies of $F(u)$ further apart
- Raise f_{max}
- Not a complete solution since may need infinite f_s

- Continuous image sampled at n times final resolution
- Low pass filtered
  - apply convolution with filter like Bartelett window

1	2	1
2	4	2
1	2	1

- Re-sampled at final resolution
- Correspond to sampling, reconstruction, and re-sampling

Continuous domain

- Difficult to calculate continuous analytical $I(x,y)$
- Assume I continuous across filter kernel
  e.g. clip polygons against pixel edge
  (filter using box filter)

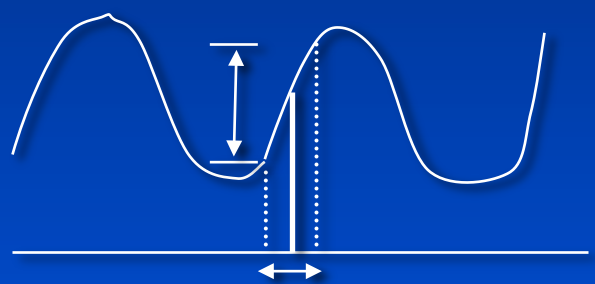
A-buffer

- Sub-pixel mask
- Coverage area weighting by logical ops on masks
- Assume I continuous over mask for each polygon
- Hybrid of area filtering and supersampling

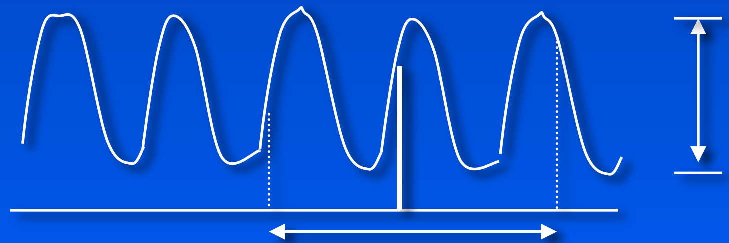
Stochastic Supersampling

- Distribute samples stochastically in various dimensions
- Trade aliasing artifacts with noise
- In our eyes, Poisson distribution (minimum-distance) no pair of samples closer than some minimum distance

- Jittered sampling: each sample jittered by random noise
- Amount of jittering a function of sampling wavelength

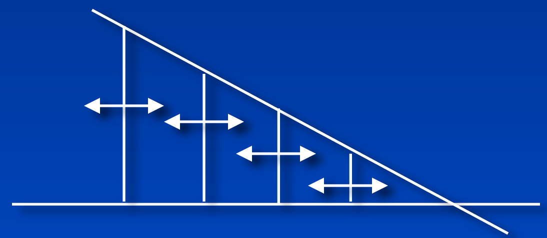


Jittering sampling frequency above Nyquist Limit

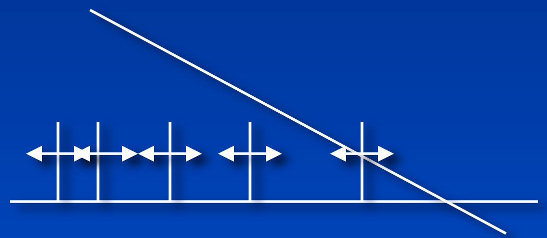


Below Nyquist Limit

- Supersampling must get weighted sum of samples alternative : importance sampling



Weighted sum



Importance sampling

- Divide weighting filter into region of equal area – assign same number of equally weighted sample in each area jitter proportional to sampling period

In general: Monte Carlo Integration

- Can sample in other dimensions
  - time - temporal aliasing
  - depth of field – distribute over area of camera lens
  - rough surface – distribute over BRDF
  - soft shadows – distribute over solid angle of light as seen from surface

Pre-filtering

- In texture mapping, can pre-filter for efficiency
  - MIP maps
  - Summed Area Tables
- Space variant filter

Next: Texture mapping