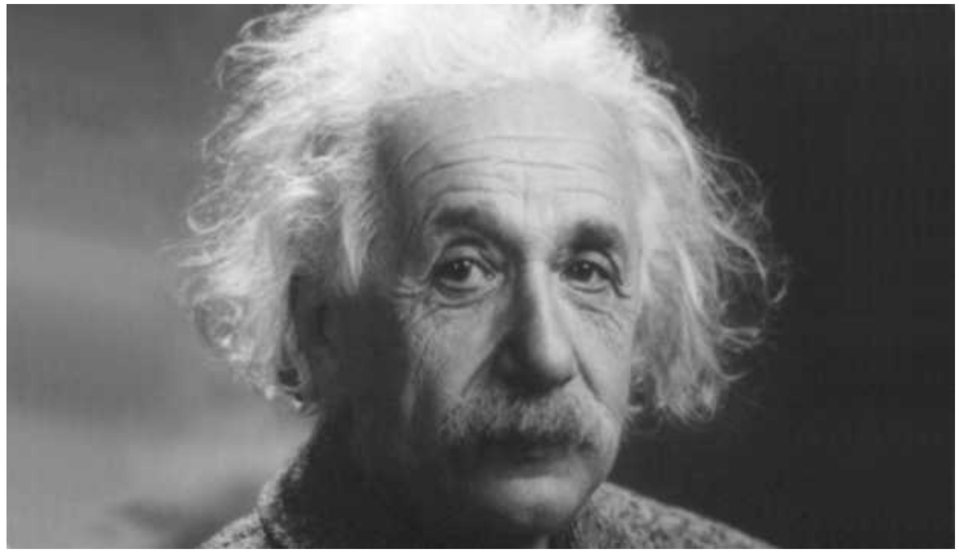
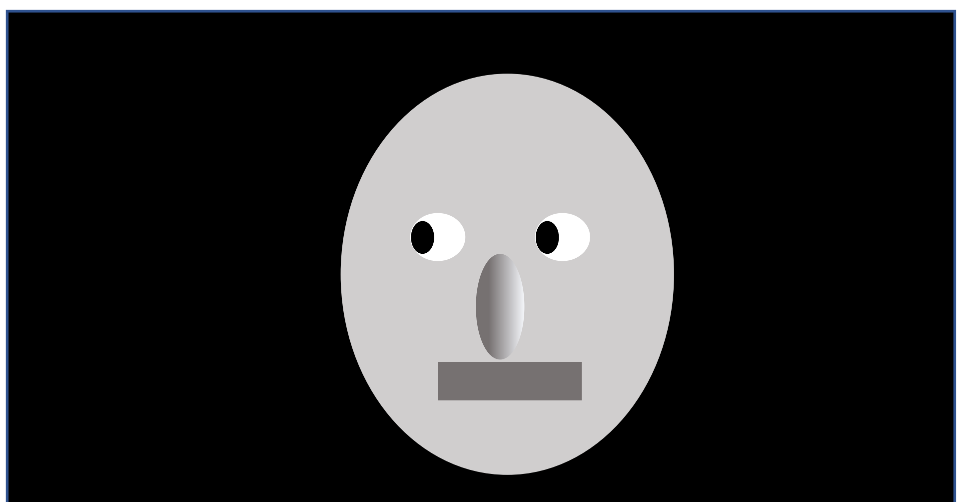
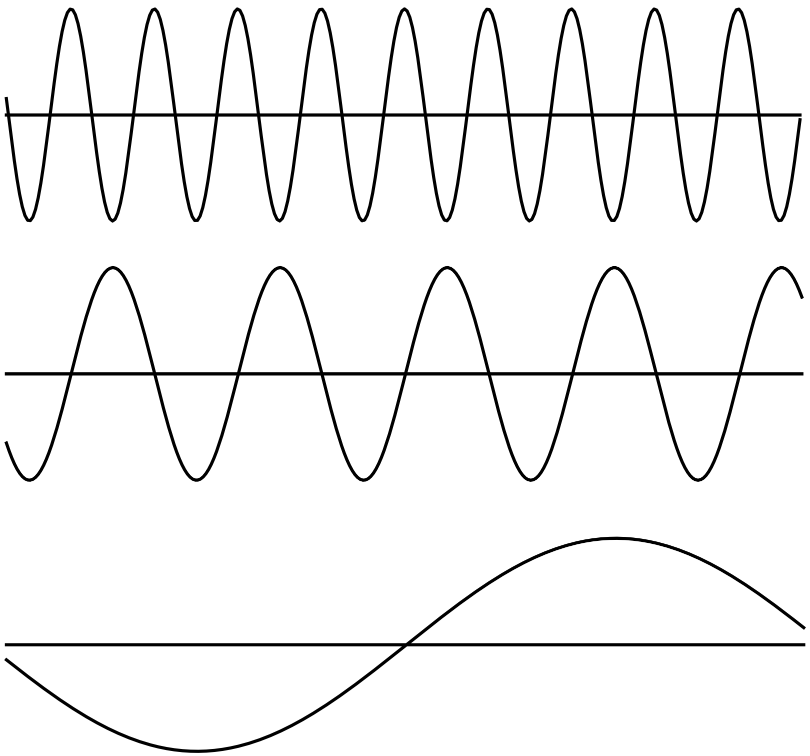
Images have structure at various scales



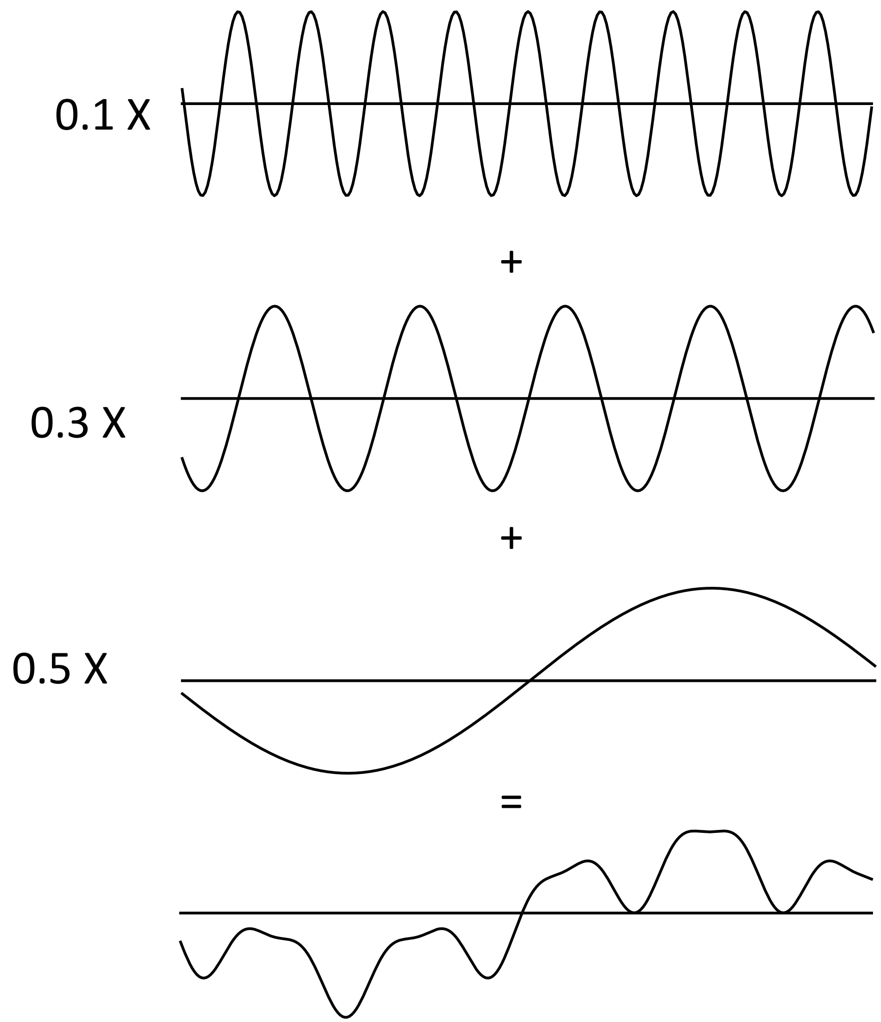


Frequency

- *Frequency* of a signal is how fast it changes
  - Reflects scale of structure



A combination of frequencies



Fourier transform

- Can we figure out the canonical single-frequency signals that make up a complex signal?
  - *Yes!*
- Can *any* signal be decomposed in this way?
  - *Yes!*

Fourier transform for periodic signals

- Suppose x is periodic with period T
- All components must be periodic with period T/k for some integer k
  - Only frequencies are of the form k/T
- Thus:

$$x(t) = \sum_{-\infty}^{\infty} a_k e^{i \frac{2\pi k t}{T}}$$

”Pure” signal

- Given a signal x(t), Fourier transform gives us the coefficients a_k (we will denote these as X[k])

Fourier transform for aperiodic signals

- What if signal is not periodic?
- Can *still* decompose into sines and cosines!
- But no restriction on frequency
- Now need a *continuous space* of frequencies

$$x(t) = \int_{-\infty}^{\infty} X(\nu) e^{i2\pi\nu t} d\nu$$

"Pure" signal

- Fourier transform gives us the *function X(v)*

Fourier transform

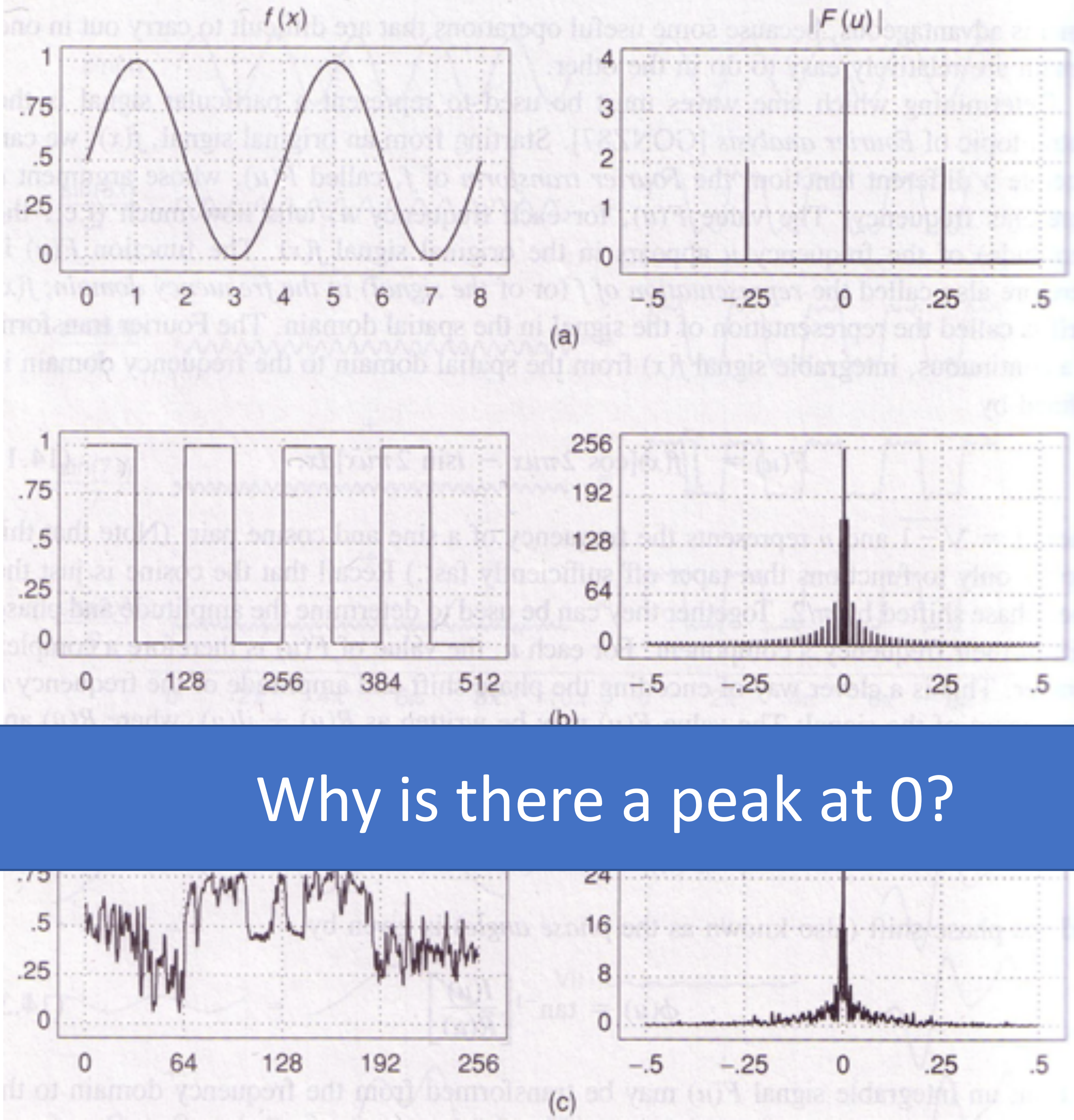
$$x(t) = \int_{-\infty}^{\infty} X(\nu) e^{i2\pi\nu t} d\nu$$

$$X(\nu) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\nu t} dt$$

Note: X can in principle be complex: we often look at the magnitude |X(v)|

Time

Frequency



Why is there a peak at 0?

Fourier transform

$$x(t) = \int_{-\infty}^{\infty} X(\nu) e^{i2\pi\nu t} d\nu$$

$$X(\nu) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\nu t} dt$$

Dual domains

- Signal: time domain (or spatial domain)
- Fourier Transform: Frequency domain
  - Amplitudes are called spectrum
- For any transformations we do in time domain, there are corresponding transformations we can do in the frequency domain
- *And vice-versa*

Dual domains

- *Convolution* in time domain = *Point-wise multiplication* in frequency domain

$$h = f * g$$

$$H = FG$$

$$H(\nu) = F(\nu)G(\nu)$$

- *Convolution* in frequency domain = *Point-wise multiplication* in time domain

Proof (if curious)

$$
\begin{aligned}
H(\nu) &= \int_{-\infty}^{\infty} h(t)e^{-i2\pi\nu t}dt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)g(t-x)dxe^{-i2\pi\nu t}dt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)e^{-i2\pi\nu x}g(t-x)e^{-i2\pi\nu(t-x)}dxdt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)e^{-i2\pi\nu x}g(u)e^{-i2\pi\nu u}dxdu \\
&= \int_{-\infty}^{\infty} f(x)e^{-i2\pi\nu x}dx\int_{-\infty}^{\infty} g(u)e^{-i2\pi\nu u}du \\
&= F(\nu)G(\nu)
\end{aligned}
$$

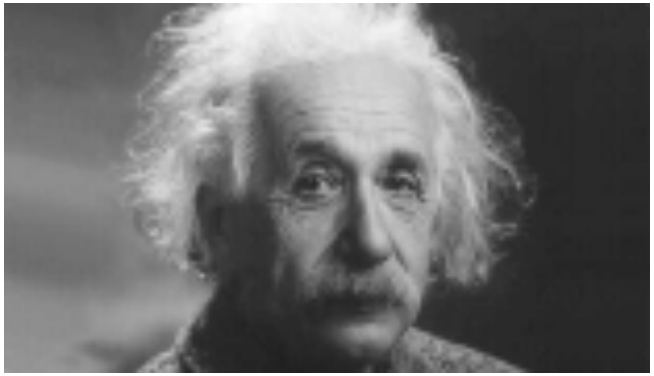
Properties of Fourier transforms

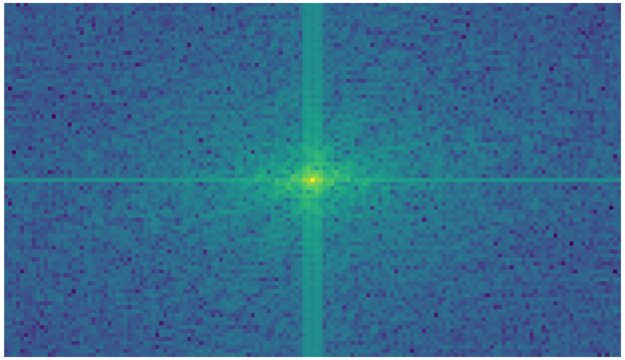
Property	Signal		Transform
superposition	$f_1(x) + f_2(x)$		$F_1(\omega) + F_2(\omega)$
shift	$f(x - x_0)$		$F(\omega)e^{-j\omega x_0}$
reversal	$f(-x)$		$F^*(\omega)$
convolution	$f(x) * h(x)$		$F(\omega)H(\omega)$
correlation	$f(x) \otimes h(x)$		$F(\omega)H^*(\omega)$
multiplication	$f(x)h(x)$		$F(\omega) * H(\omega)$
differentiation	$f'(x)$		$j\omega F(\omega)$
domain scaling	$f(ax)$		$1/aF(\omega/a)$
real images	$f(x) = f^*(x)$	$\Leftrightarrow$	$F(\omega) = F(-\omega)$
Parseval's Theorem	$\sum_x [f(x)]^2$	$=$	$\sum_\omega [F(\omega)]^2$

Back to 2D images

- Images are 2D signals
- Discrete, but consider as samples from continuous function
- Signal: f(x,y)
- Fourier transform $F(v_x, v_y)$: contribution of a "pure" signal with frequency v_x in x and v_y in y

Back to 2D images





Signals and their Fourier transform

Spatial ⟺ Frequency

Spatial	Frequency
Sine	Impulse
Gaussian	Gaussian
Box	Sinc

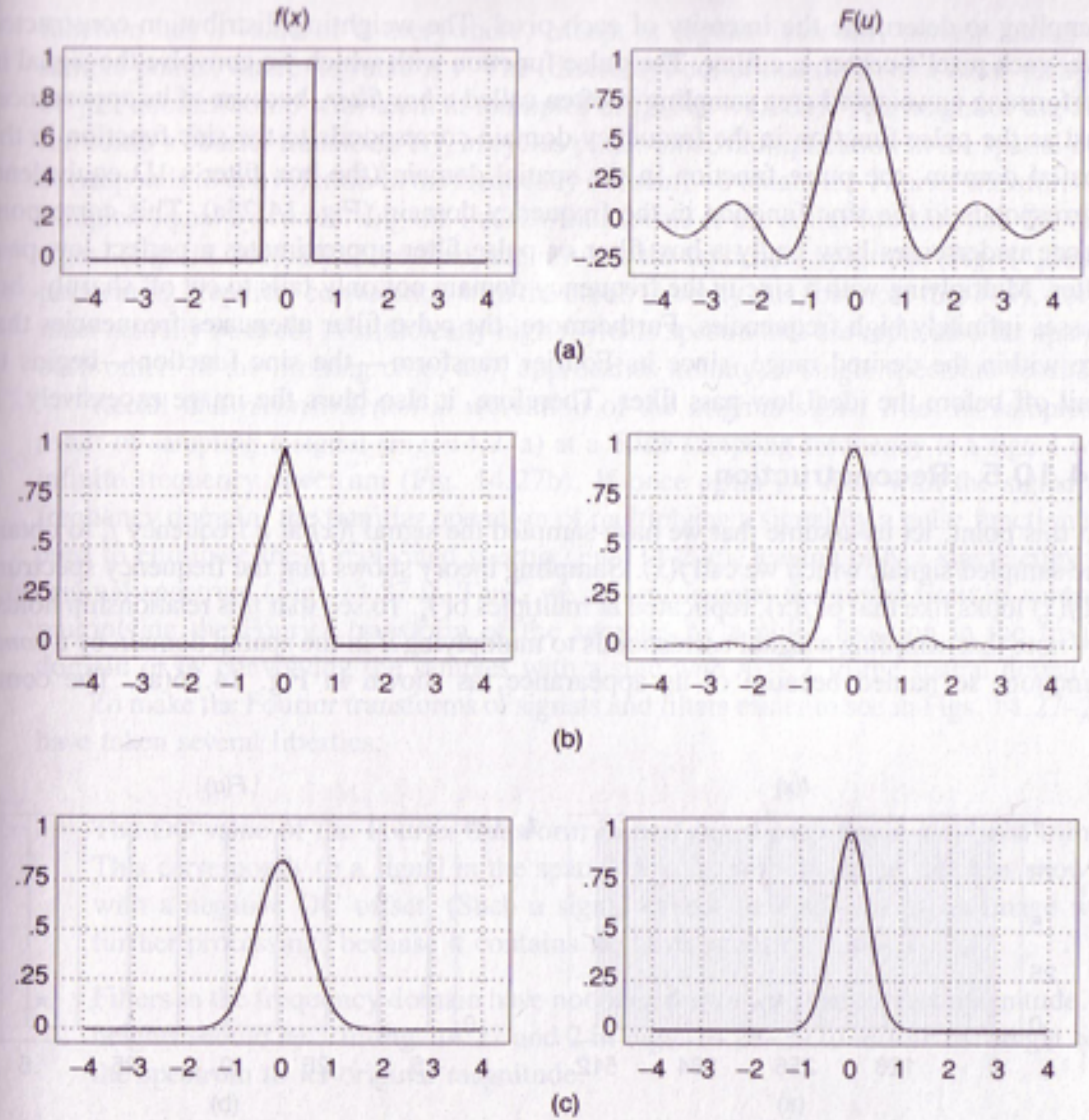
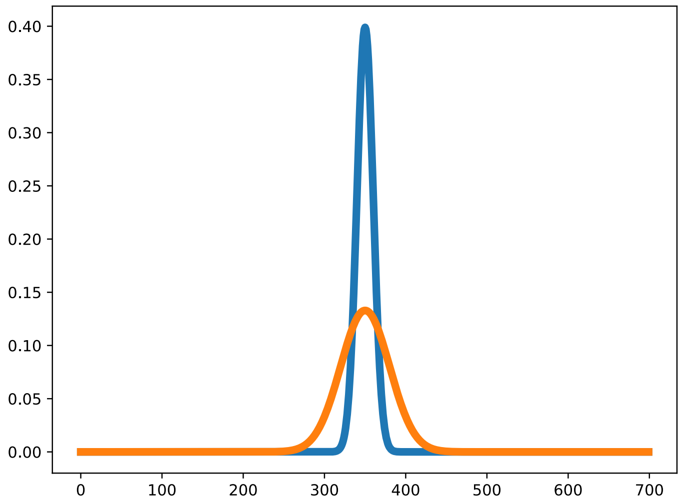
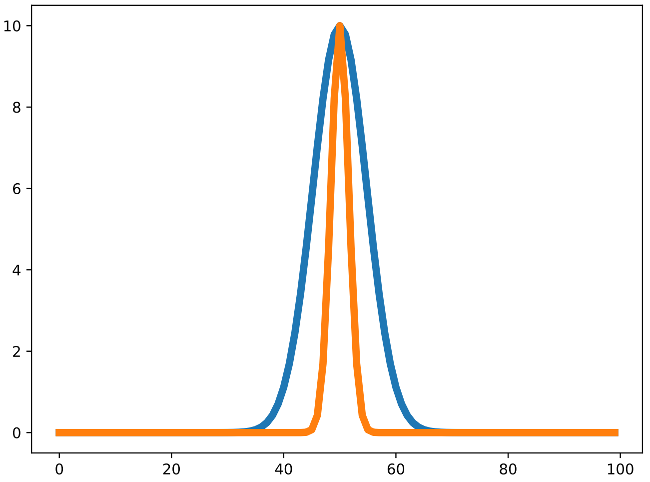


Fig. 14.25 Filters in spatial and frequency domains. (a) Pulse—sinc. (b) Triangle—sinc². (c) Gaussian—Gaussian. (Courtesy of George Wolberg, Columbia University.)

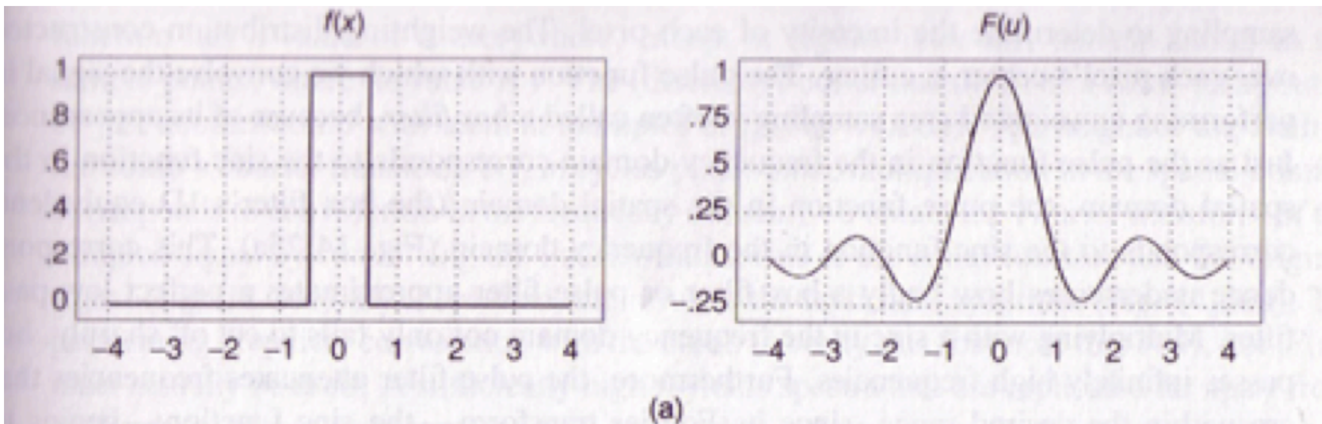
The Gaussian special case

- Fourier transform of a Gaussian is a gaussian





Sharp discontinuities require very high frequencies



$$\mathrm{sinc}(x) = \frac{\sin(x)}{x}$$

Duality

$$x(t) = \int_{-\infty}^{\infty} X(\nu) e^{i2\pi\nu t} d\nu$$

$$X(\nu) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\nu t} dt$$

- Since Fourier and inverse Fourier look so much alike:
  - Fourier transform of sinc is box
  - Fourier transform of impulse is sine

Why talk about Fourier transforms?

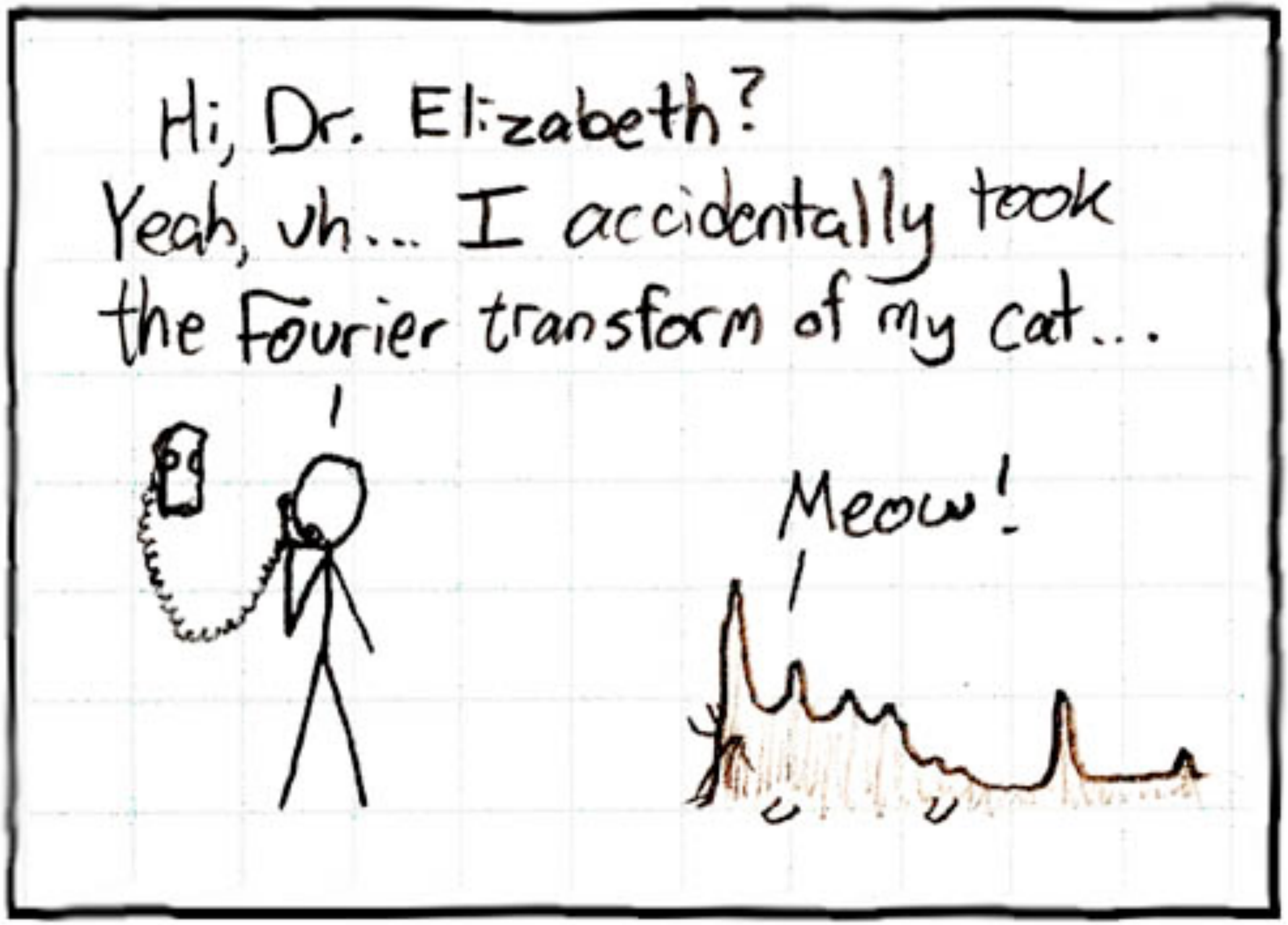
- Convolution is point-wise multiplication in frequency space
  - Analyze which frequency components a particular filter lets through, e.g., *low-pass, high-pass* or *band-pass* filters
  - Leads to fast algorithms for convolution with large filters: Fast FFT

Why talk about Fourier transforms

- Frequency space reveals structure at various scales
  - Noise is high-frequency
  - ”Average brightness” is low-frequency
- Useful to understand how we resize/resample images
  - Sampling causes information loss
  - What is lost exactly?
  - What can we recover?

Fourier transforms from far away

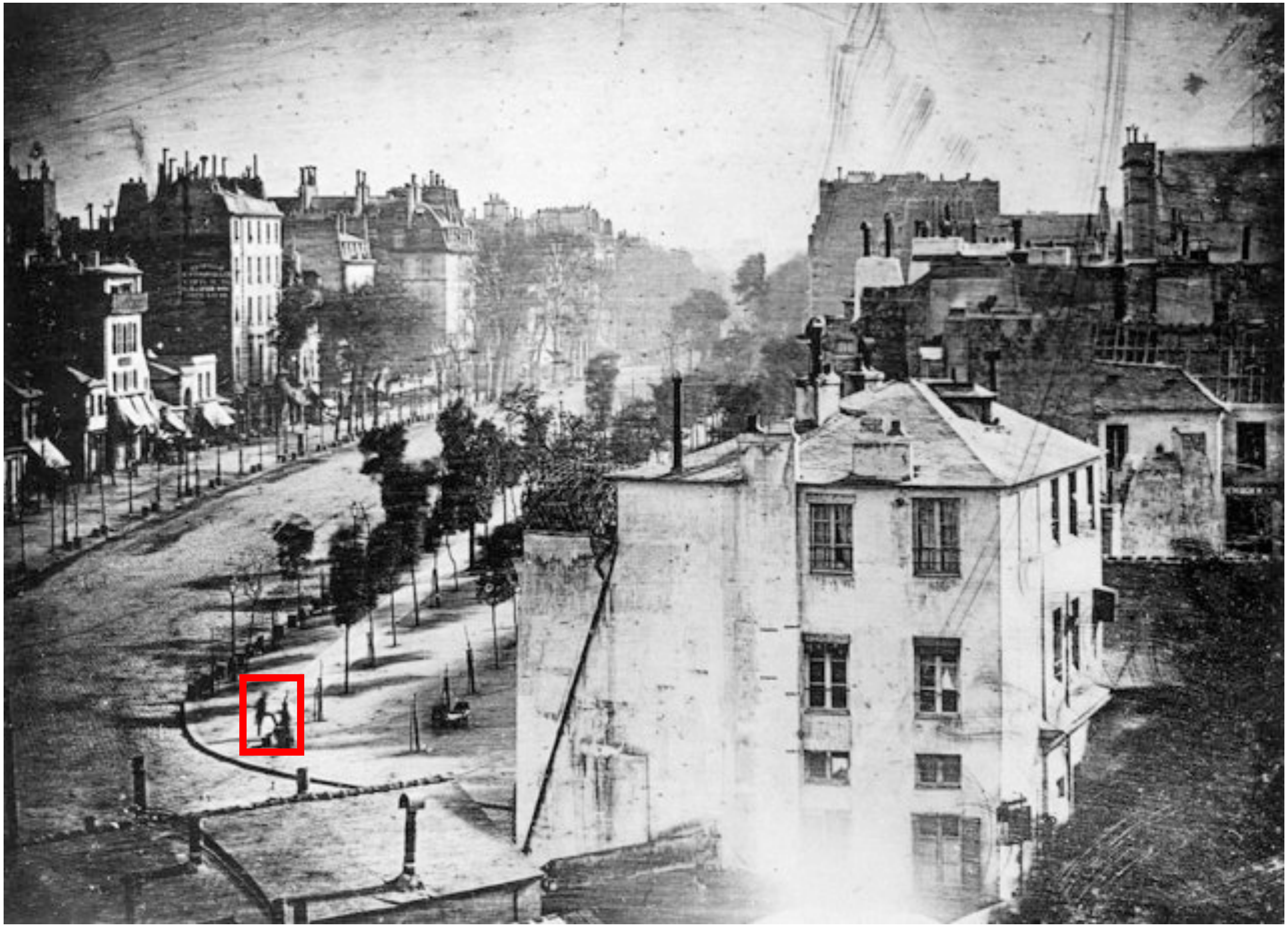
- Fourier transforms are basically a “change of basis”
- Instead of representing image as “the value of each pixel”,
- Represent image as “how much of each frequency component”
- “Frequency components” are intuitive: slowly-changing or fast-changing images



”The cat has some serious periodic components.”
https://xkcd.com/26/

Resizing and resampling

Let's enhance!



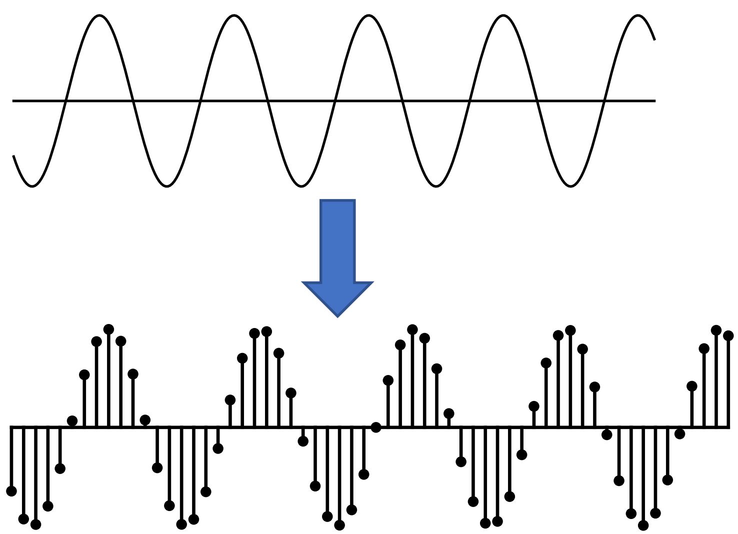
Louis Daguerre, 1838

Let’s enhance!

- When is enhancement possible?
- How can we model what happens when we upsample or downsample an image?
- Resizing up or down very common operation
  - Searching across scales
  - applications have different memory/quality tradeoffs

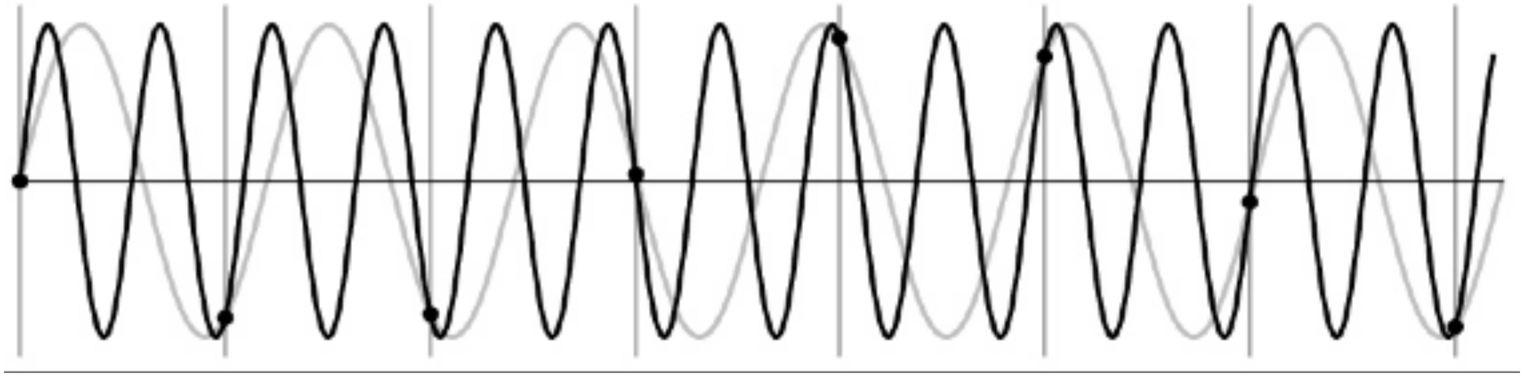
What is a (digital) image?

- True image is a function from R^2 to R
- Digital image is a sample from it
- 1D example:



- To enhance, we need to recover the original signal and sample again

Undersampling

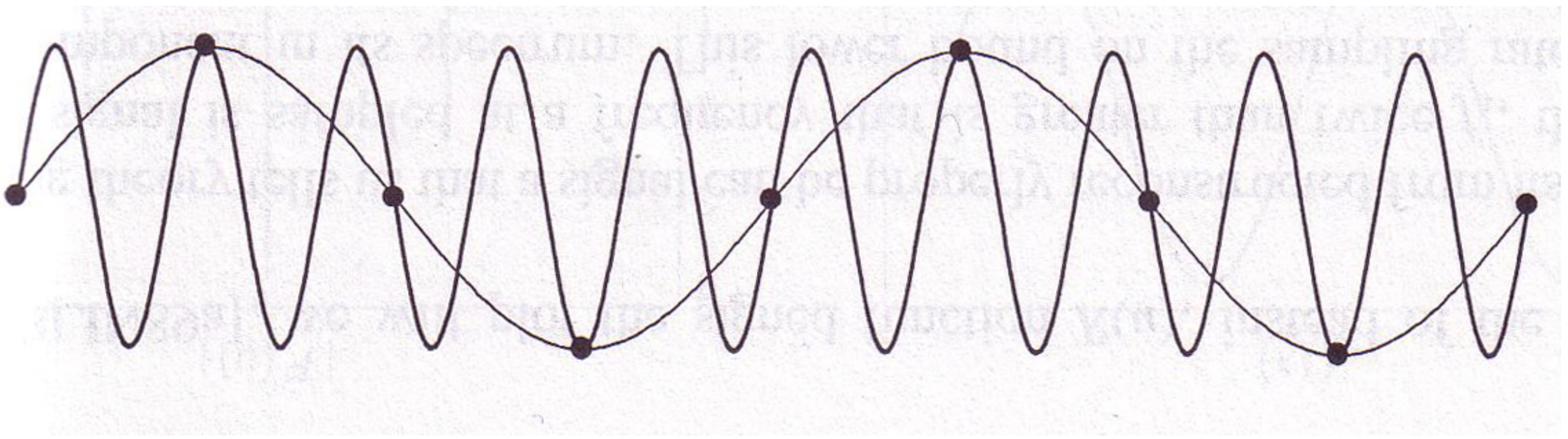


Undersampling

- What if we "missed" things between the samples?
- Simple example: undersampling a sine wave
  - unsurprising result: information is lost
  - surprising result: indistinguishable from lower frequency
  - also was always indistinguishable from higher frequencies
  - *aliasing*: signals "traveling in disguise" as other frequencies

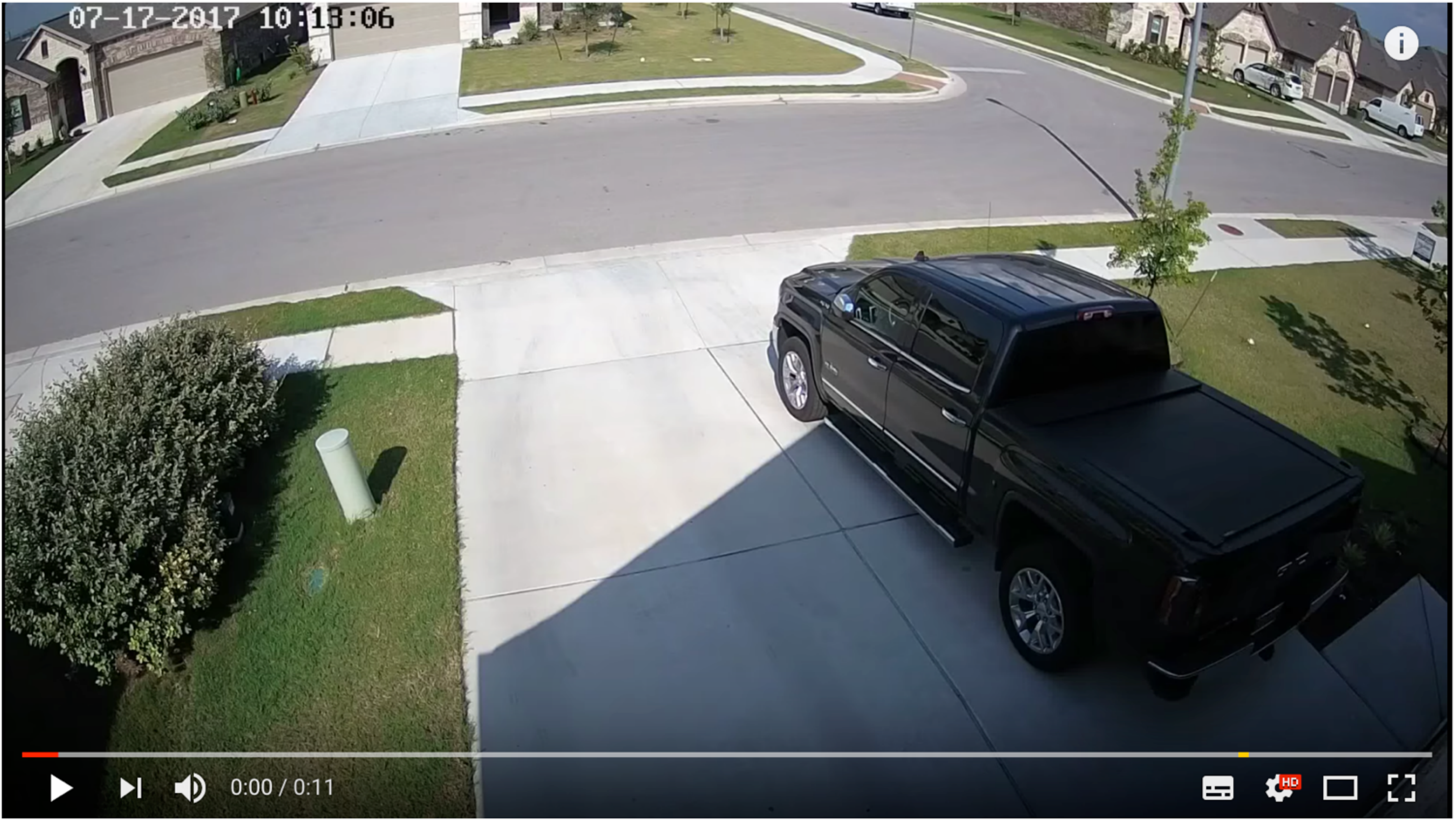
Aliasing

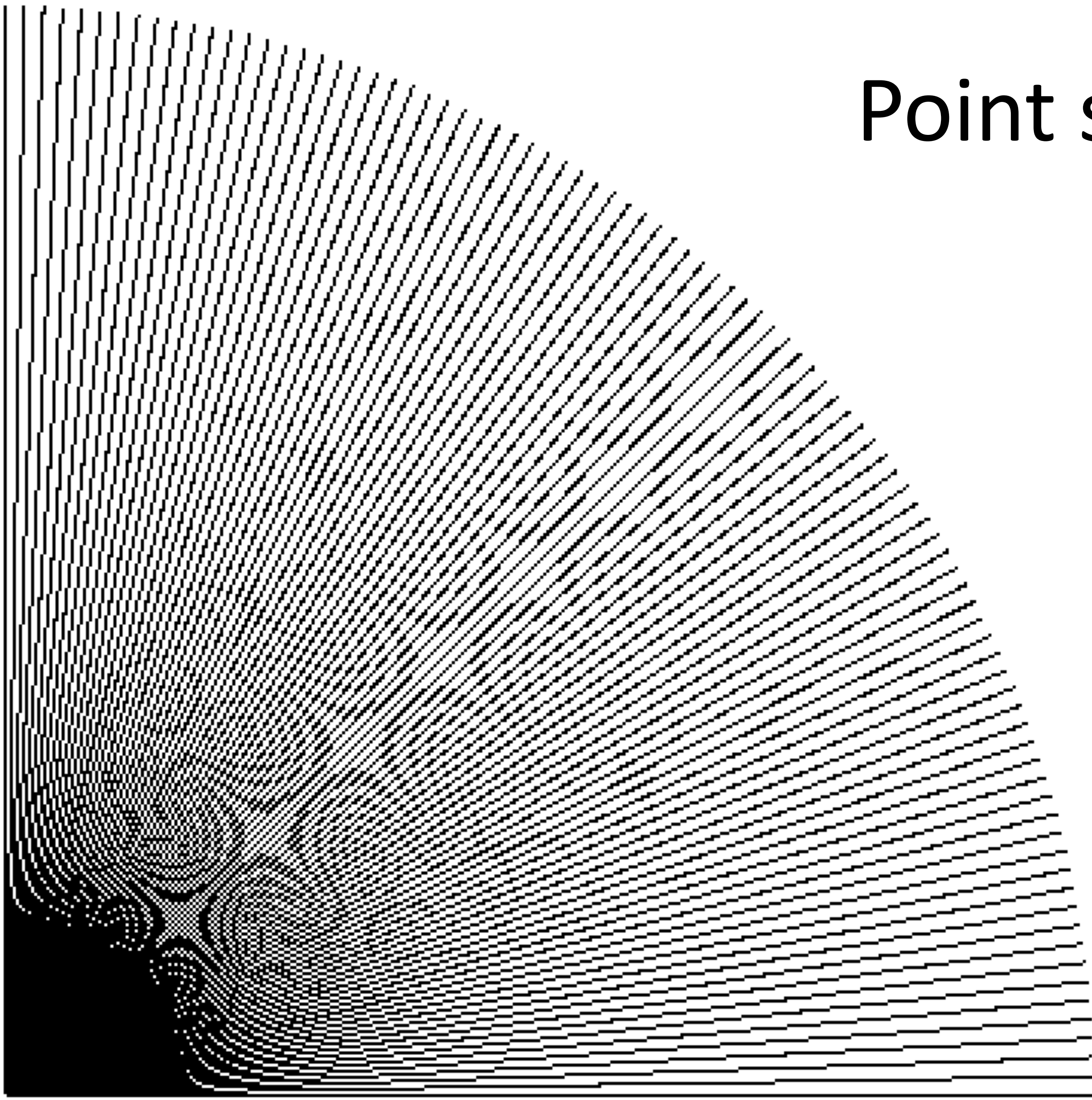
- When sampling is not adequate, impossible to distinguish between low and high frequency signal



Aliasing in time

Aliasing in time

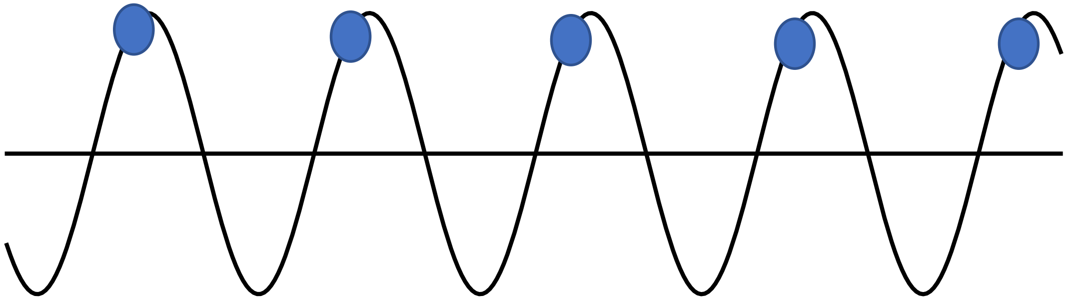




Point sampling in action

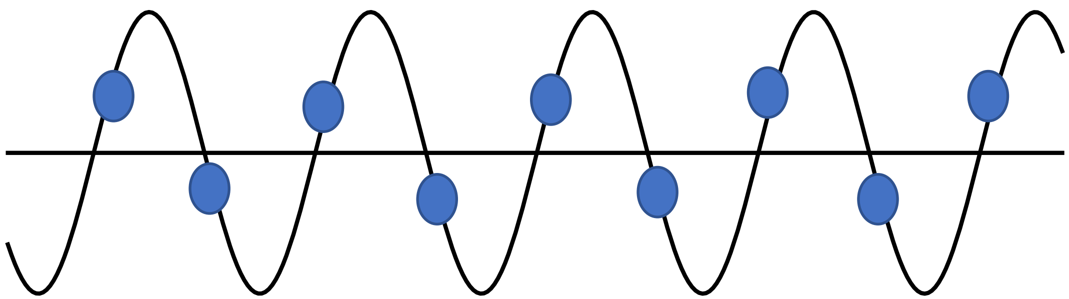
How many samples do we need?

- 1 sample per time period is too less:



How many samples do we need?

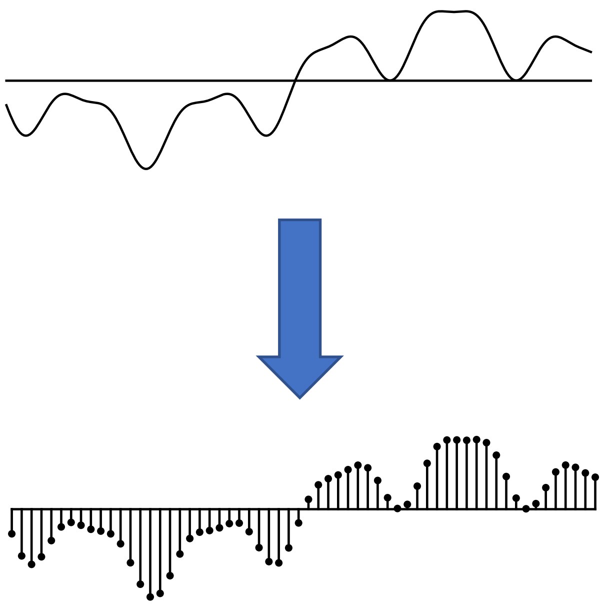
- 2 samples per time-period is enough



- Nyquist sampling theorem: Need to sample at least 2 times the frequency
- General signals? Need to sample at least 2 times the maximum frequency

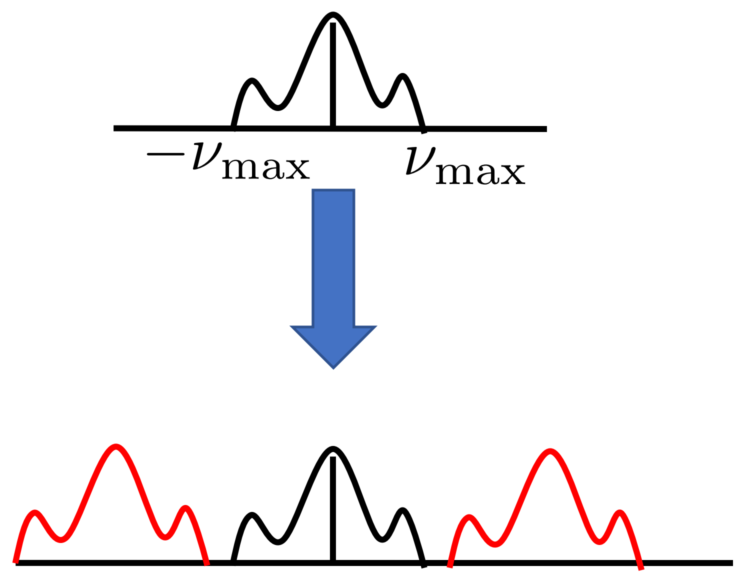
Nyquist sampling: why?

Spatial domain



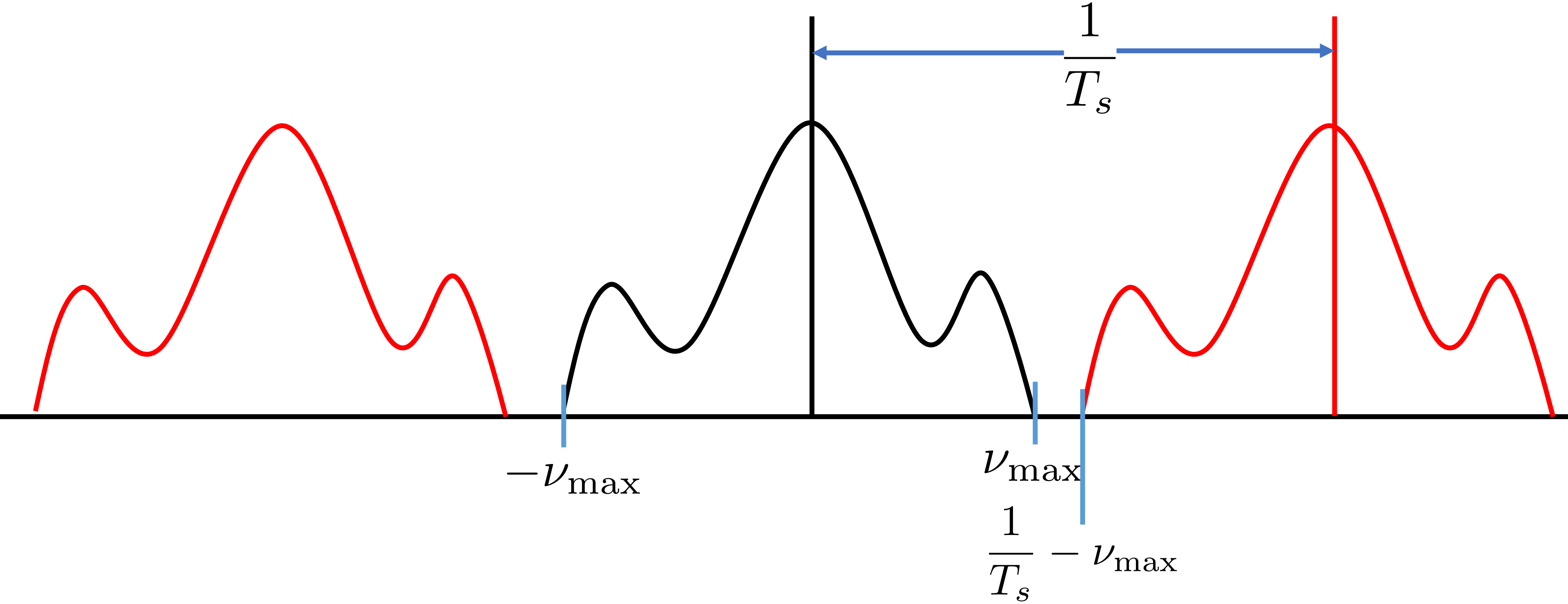
Sampling = Keep values at $t = kT_s$, make everything else 0

Frequency domain



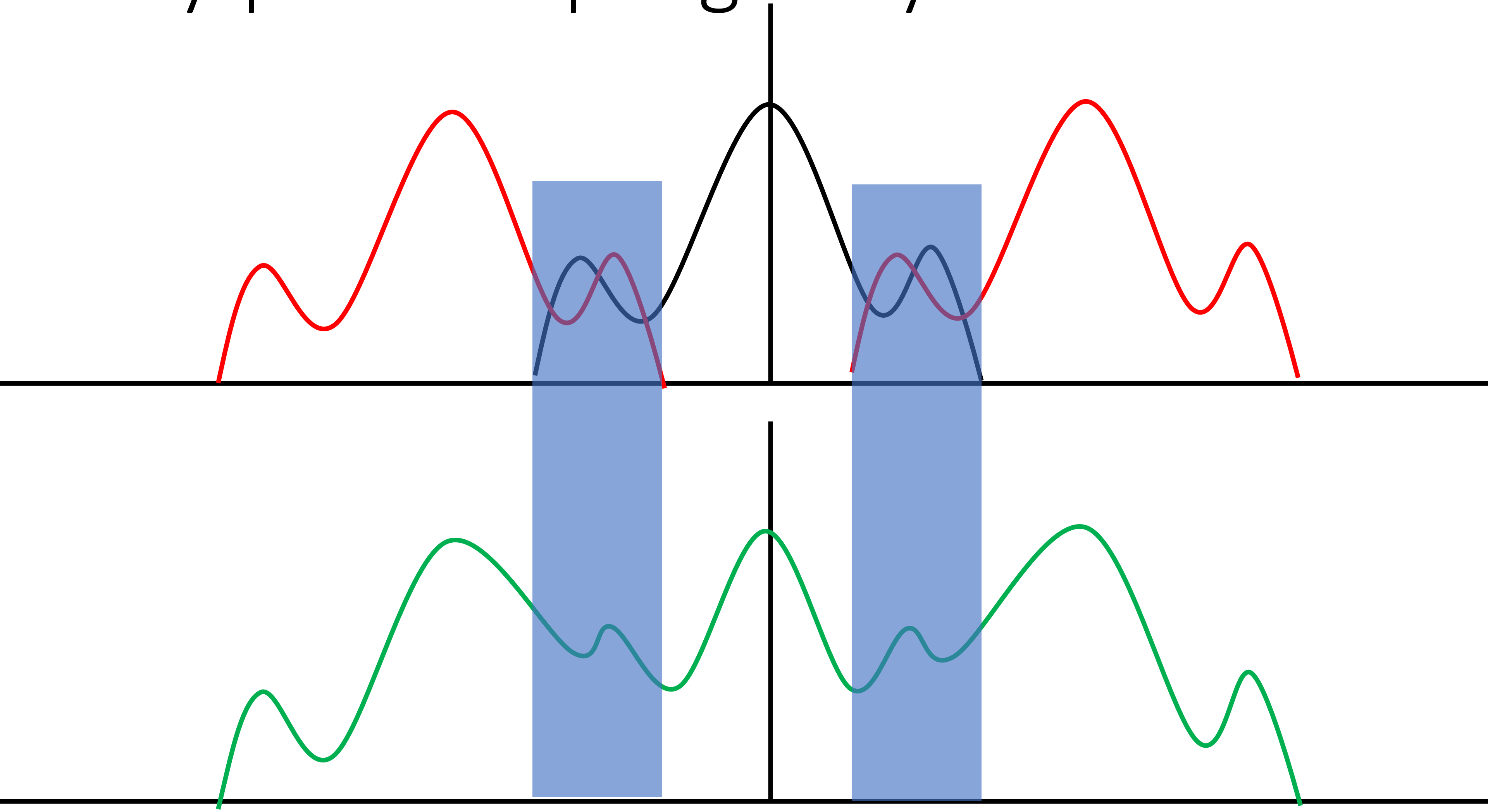
Sampling = Make frequency domain periodic with period $\nu = 1/T_s$ by making copies

Nyquist sampling: why?



$$\frac{1}{T_s} - \nu_{\max} > \nu_{\max} \Rightarrow \frac{1}{T_s} > 2\nu_{\max}$$

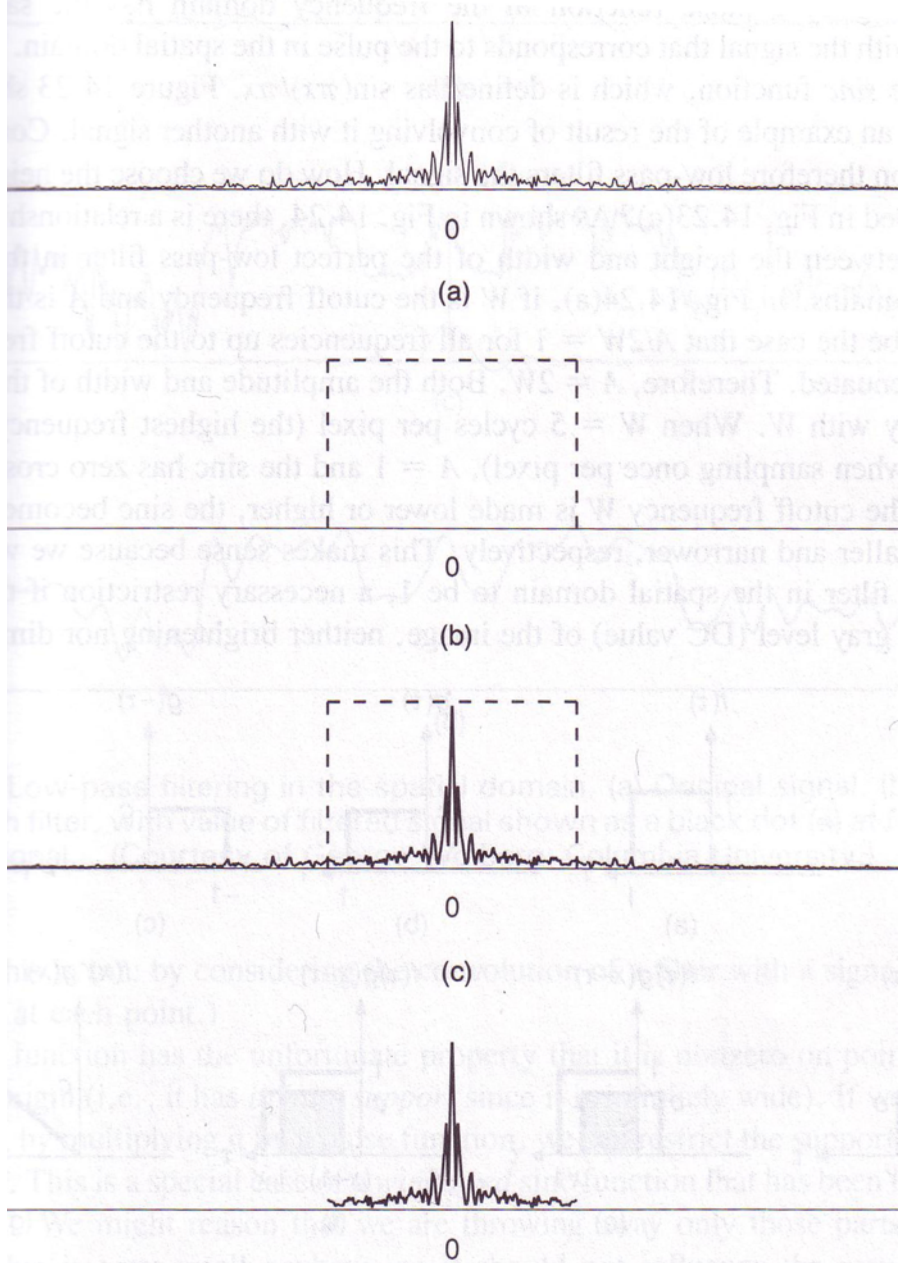
Nyquist sampling: why?



How to subsample correctly

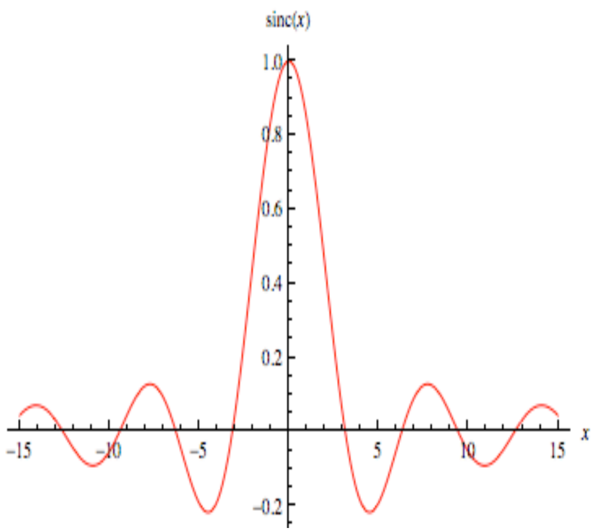
- Nyquist says must sample at at least twice maximum frequency
- What if signal has high frequencies?
- Eliminate them before sampling!
  - Convert to frequency space
  - Multiply with band-pass filter

Eliminating High Frequencies



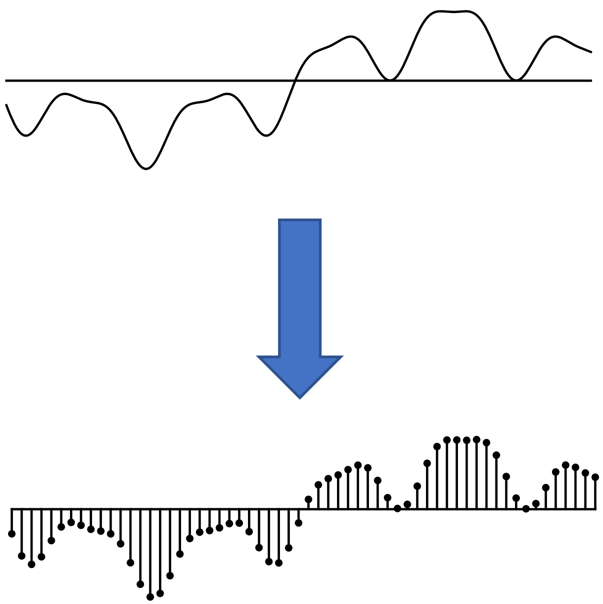
Process

- Can we do this in spatial domain?
  - Yes!
- Multiplication in frequency domain = convolution in spatial domain
- Box filter in frequency domain = sinc in spatial domain
- Multiplication with box filter in frequency domain = convolution with sinc filter in spatial domain



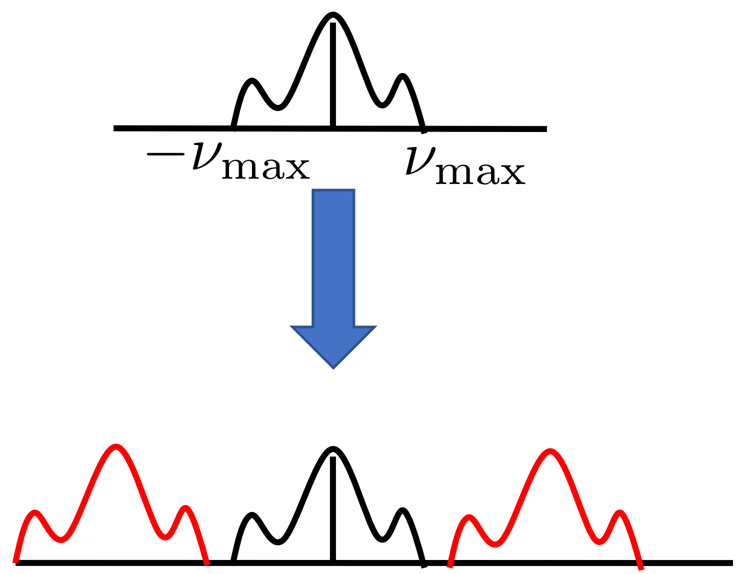
Reconstruction from samples

Spatial domain



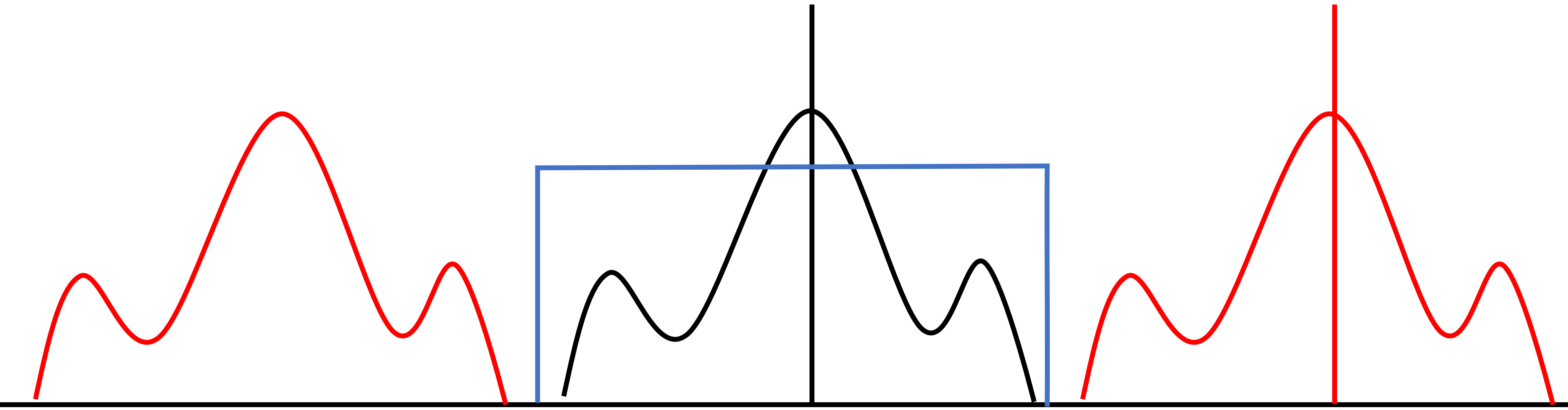
Sampling = Keep values at $t = kT_s$, make everything else 0

Frequency domain

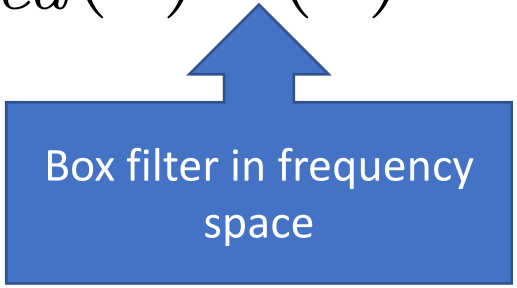


Sampling = Make frequency domain periodic with period $\nu = 1/T_s$ by making copies

Reconstruction from samples



$$F_{recons}(\nu) = F_{sampled}(\nu)B(\nu)$$



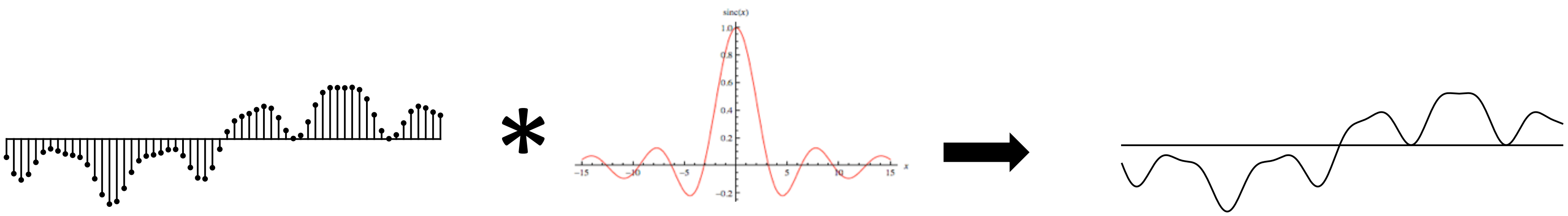
Reconstruction from samples

$$F_{recons}(\nu) = F_{sampled}(\nu)B(\nu)$$

- Multiplication in frequency domain = convolution in spatial domain
- Box filter in frequency domain = sinc filter in spatial domain
- Convolve sampled signal with sinc filter to reconstruct

Reconstruction from samples

- ”Sampled signal” is non-zero at sample points and 0 everywhere else
  - i.e., has holes



Recap: subsampling and reconstruction

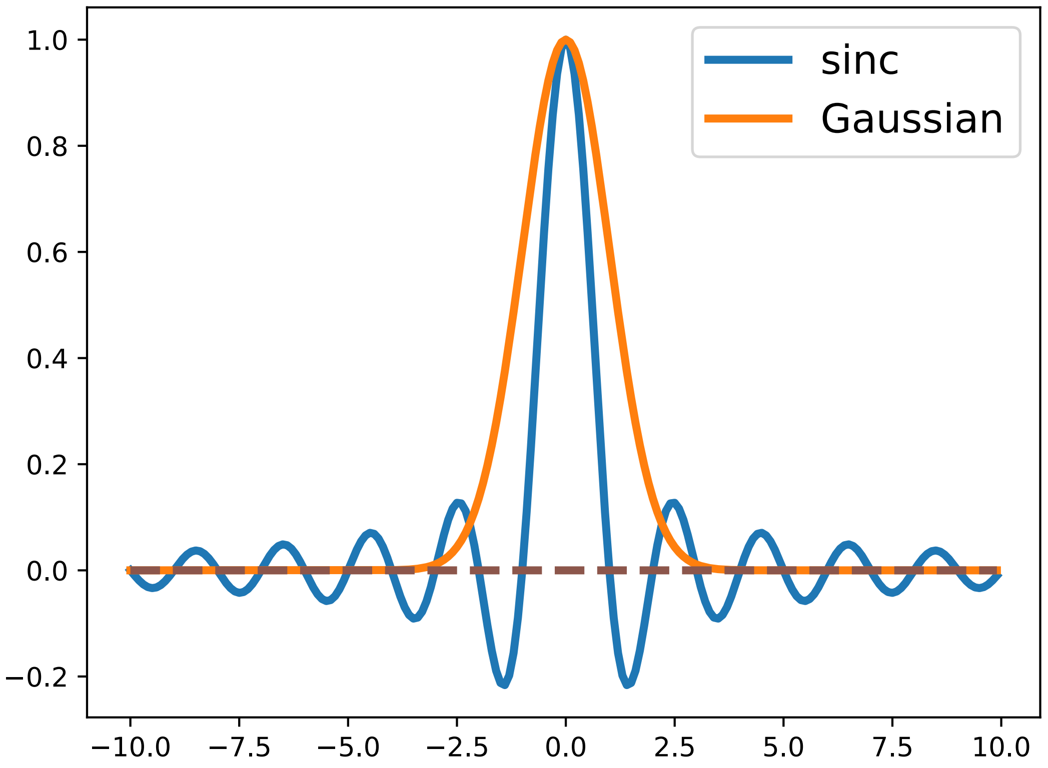
Subsampling

1. Convolve with sinc filter to eliminate high frequencies
2. Sample by picking only values at sample points

Reconstruction

1. Start with sampled signal (0 at non-sample points)
2. Convolve with sinc to reconstruct

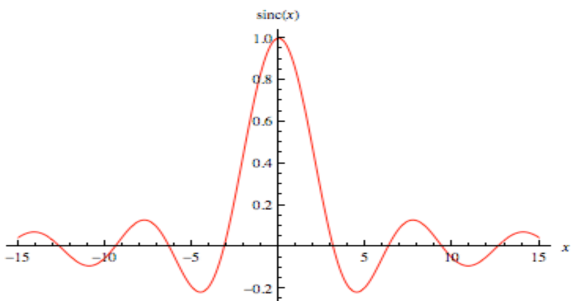
Sinc is annoying

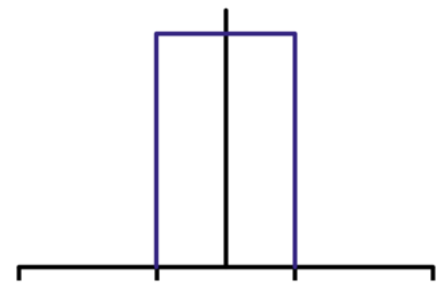


Sinc and Gaussian

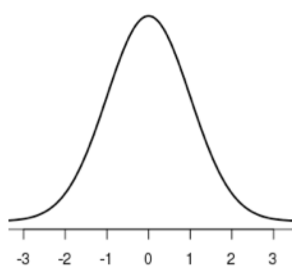
- Sinc is annoying: infinite spatial extent
- Use Gaussian instead!

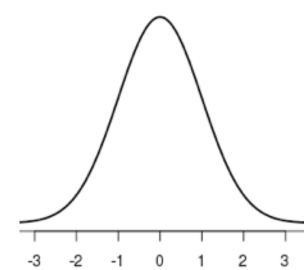
Sinc/box





Gaussian





Spatial domain

Frequency domain

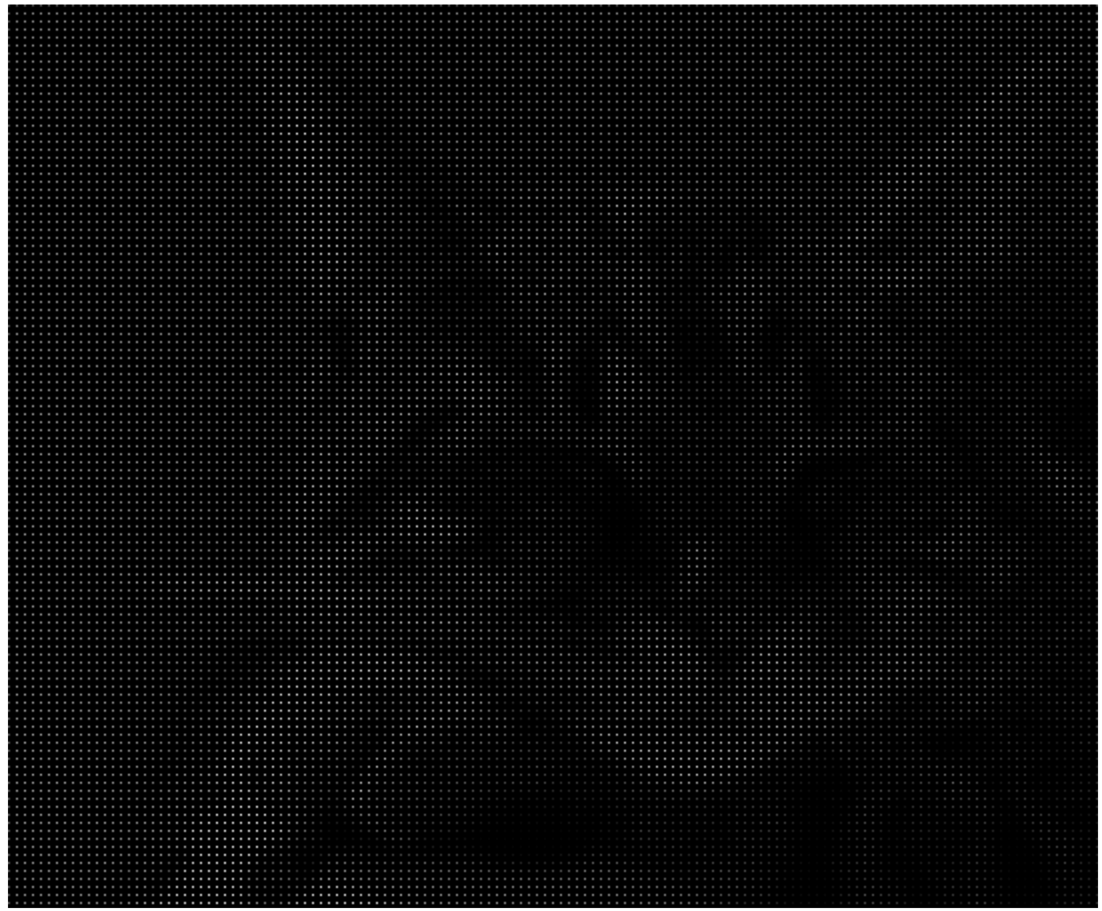
Subsampling images

- Step 1: Convolve with Gaussian to eliminate high frequencies
- Step 2: Drop unneeded pixels

Subsampling without removing high frequencies

Subsampling after removing high frequencies

Upsampling images



Step 1: blow up to original size with 0's in between

Upsampling images

Step 2: Convolve with Gaussian

Take-away

- Subsampling causes aliasing
  - High frequencies masquerading as low frequencies
- Remove low frequencies by blurring!
  - Ideal: sinc
  - Common: Gaussian
- When upsampling, reconstruct missing values by convolution
  - Ideal: sinc
  - Common: Gaussian

So… can we enhance?

- Nyquist theorem limits frequencies we can reconstruct from subsampled image
- Can only reconstruct max sampling frequency/2
- Sorry CSI!