All about convolution

Last time: Convolution and cross-correlation

- Cross correlation

$$S[f] = w \otimes f$$
$$S[f](m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m+i, n+j)$$

- Convolution

$$S[f] = w * f$$
$$S[f](m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m-i, n-j)$$

Last time: Convolution and cross-correlation

- Properties
  - Shift-invariant: a sensible thing to require
  - Linearity: convenient
- Can be used for smoothing, sharpening
- Also main component of CNNs

Boundary conditions

$$(w * f)(m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m-i, n-j)$$

- What if m-i <0?
- What if m-i > image size
- Assume f is defined for $[-\infty, \infty]$ in both directions, just 0 everywhere else
- Same for w

$$(w * f)(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} w(i,j) f(m-i, n-j)$$

Boundary conditions

90	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

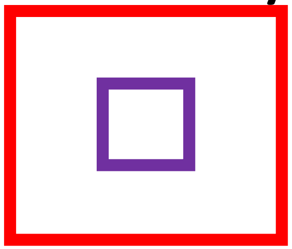
Boundary conditions

90	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

Boundary conditions

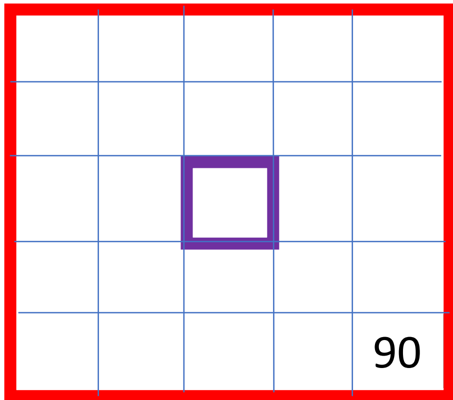
90	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

Boundary conditions



90	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

Boundary conditions



90	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

Boundary conditions in practice

- “Full convolution”: compute if *any* part of kernel intersects with image
  - requires padding
  - Output size = m+k-1
- “Same convolution”: compute if center of kernel is in image
  - requires padding
  - output size = m
- “Valid convolution”: compute only if *all* of kernel is in image
  - no padding
  - output size = m-k+1

More properties of convolution

$$(w * f)(m,n) = \sum_i \sum_j w(i,j) f(m-i, n-j)$$

$$= \sum_i \sum_j w(m-i', n-j') f(i,j)$$

$$= (f * w)(m,n)$$

$$i' = m - i \Rightarrow i = m - i'$$

$$j' = n - j \Rightarrow j = n - j'$$

More properties of convolution

- Convolution is linear
- Convolution is shift-invariant
- Convolution is commutative (w*f = f*w)
- Convolution is *associative* (v*(w*f) = (v*w)*f)
- Every linear shift-invariant operation is a convolution

Optimization: separable filters

- basic alg. is $O(r^2)$: large filters get expensive fast!
- definition: $a_2(x,y)$ is *separable* if it can be written as:
  - this is a useful property for filters because it allows factoring:

$$a_2[i,j] = a_1[i]a_1[j]$$

$$(a_2 \star b)[i,j] = \sum_{i'} \sum_{j'} a_2[i',j']b[i-i',j-j']$$

$$= \sum_{i'} \sum_{j'} a_1[i']a_1[j']b[i-i',j-j']$$

$$= \sum_{i'} a_1[i'] \left(\sum_{j'} a_1[j']b[i-i',j-j'] \right)$$

More convolution filters

- Mean filter

1/25

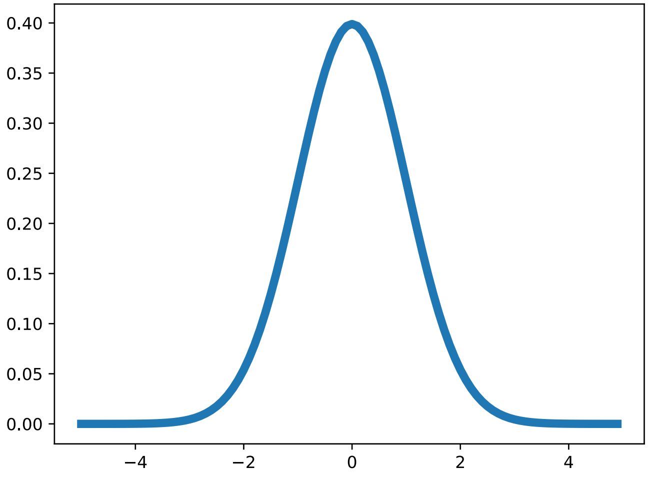
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

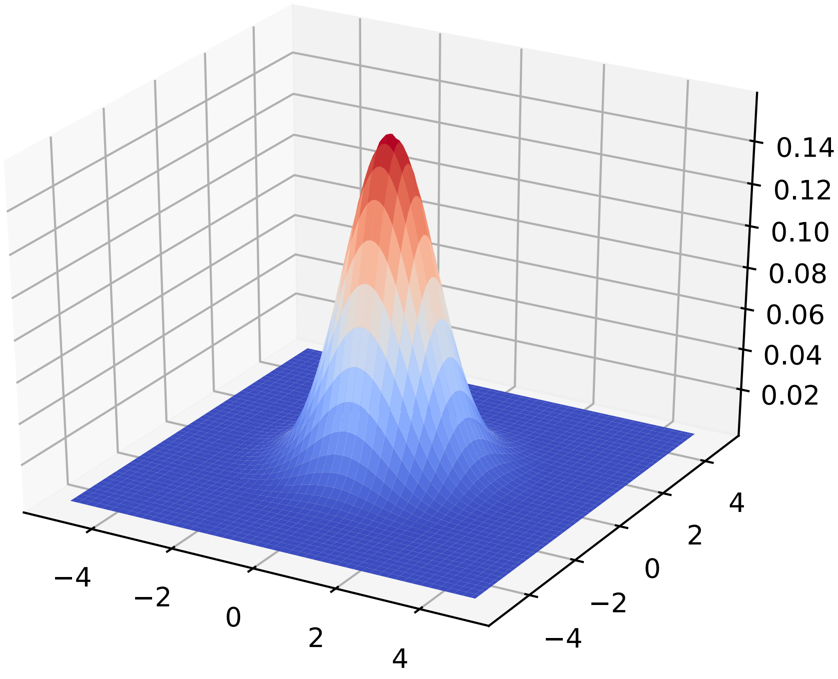
- But nearby pixels are more correlated than far-away pixels
- Weigh nearby pixels more

Gaussian filter

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$





Gaussian filter

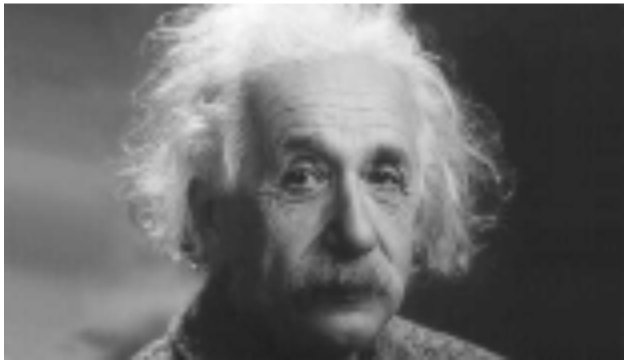
$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

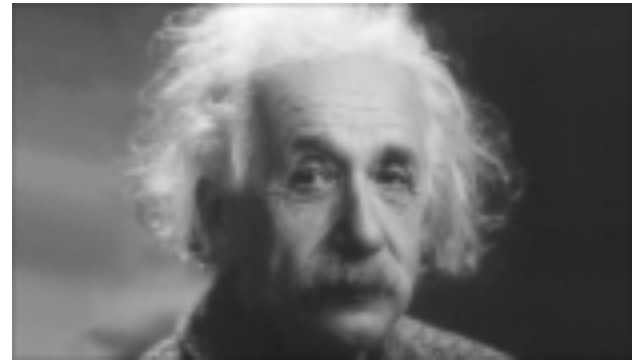
- Ignore factor in front, instead, normalize filter to sum to 1

0.003	0.013	0.022	0.013	0.003
0.013	0.060	0.098	0.060	0.013
0.022	0.098	0.162	0.098	0.022
0.013	0.060	0.098	0.060	0.013
0.003	0.013	0.022	0.013	0.003

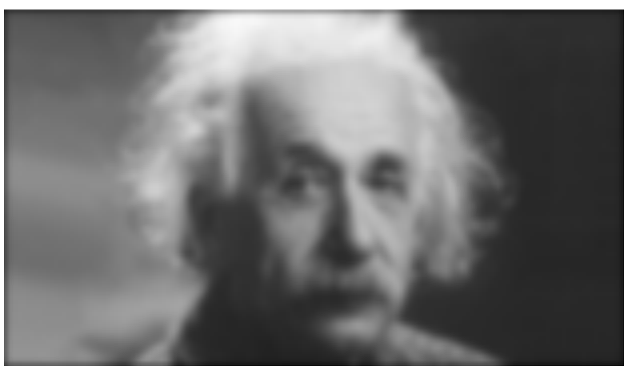
5x5, σ=1

Gaussian filter

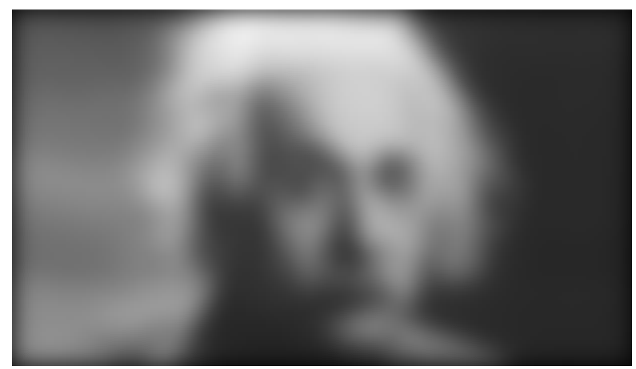




21x21, σ=0.5

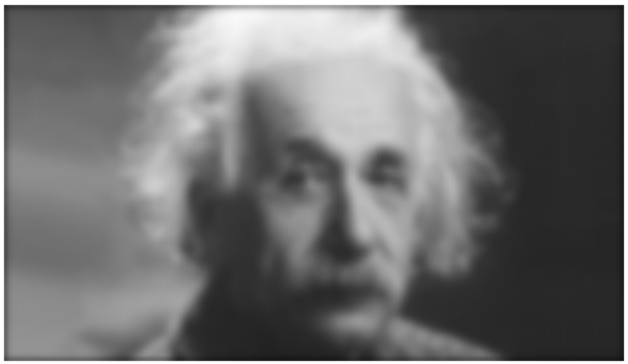


21x21, σ=1

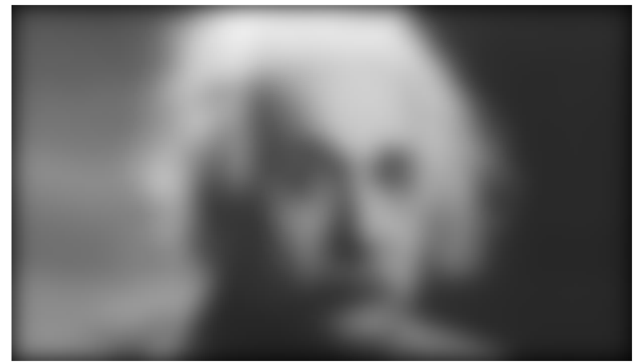


21x21, σ=3

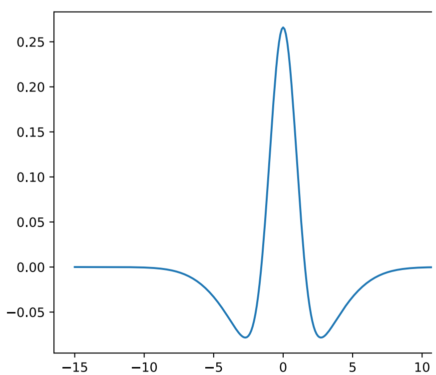
Difference of Gaussians

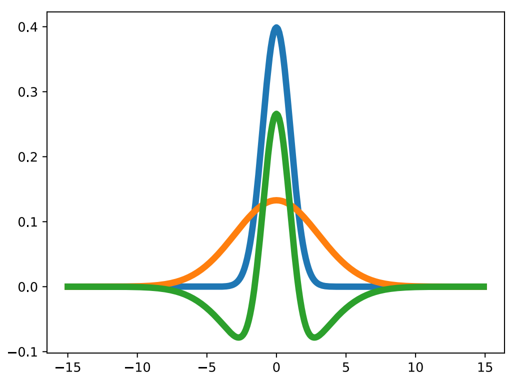


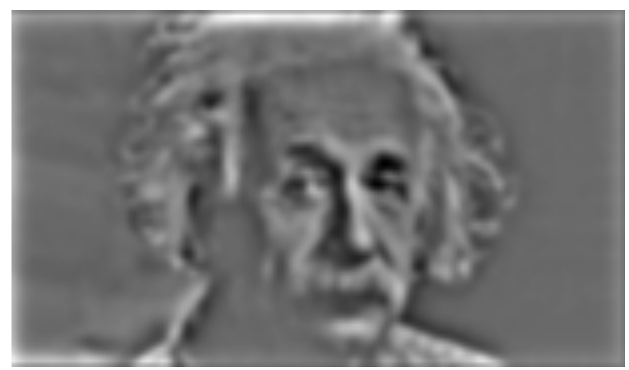
21x21, σ=1



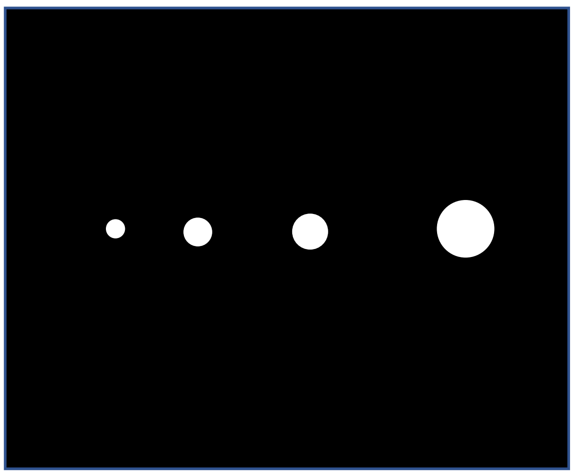
21x21, σ=3



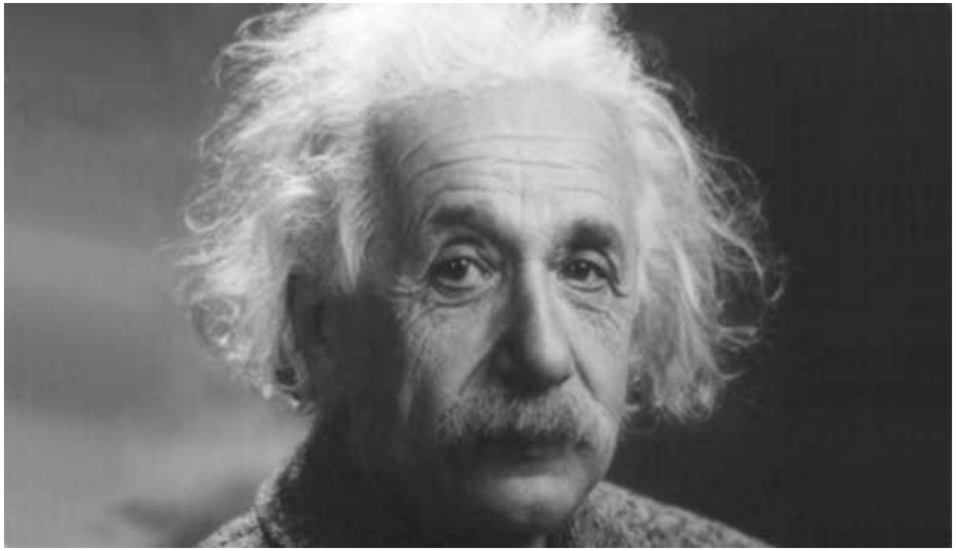


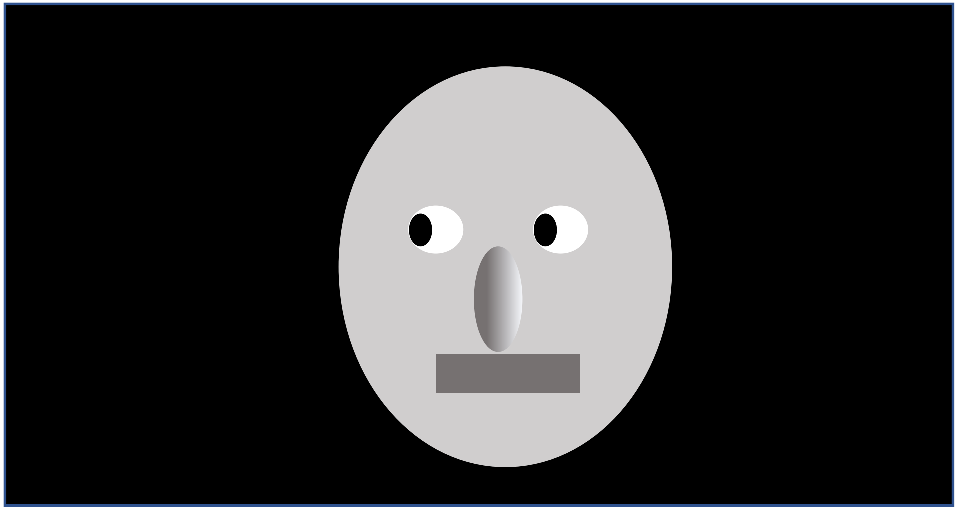


Difference of Gaussians



Images have structure at various scales



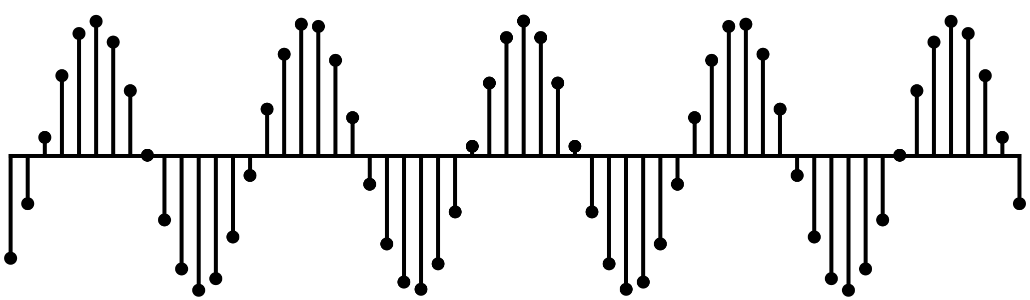


Fourier transform and the frequency domain

Signal processing

- Images are 2D
- For convenience, consider 1D signals
  - Instead of space, time
- f[i] : value of signal at point i (1D analog of f(i,j))

$$(w * f)[n] = \sum_i w[i] f[n-i]$$



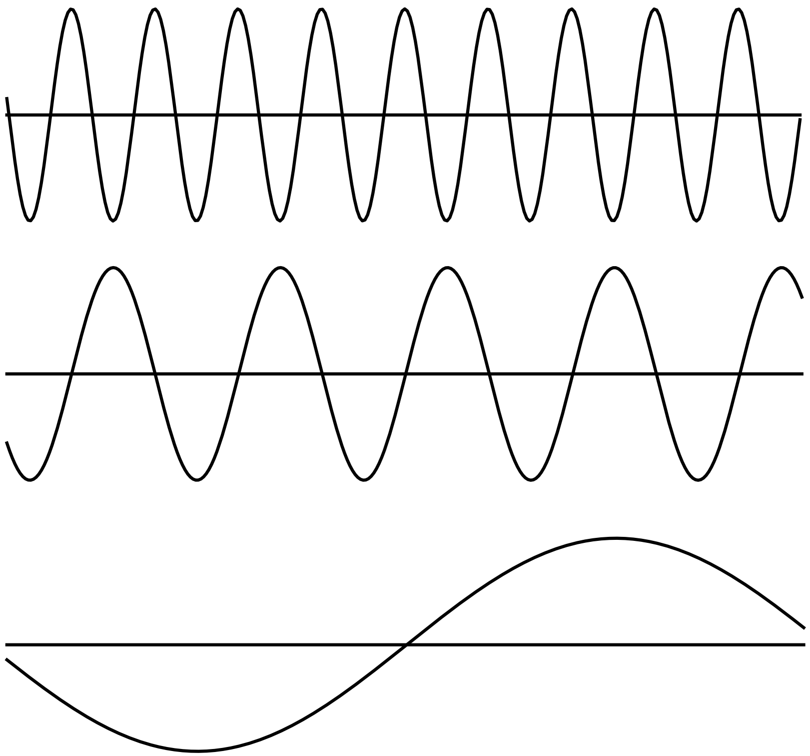
Signal processing

- Instead of discrete signals, we will consider *continuous signals*
- Discrete signals can be considered as samples from continuous signals
- f : **R** → **R**, w : **R** → **R**
- What is convolution for continuous signals?

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

Frequency

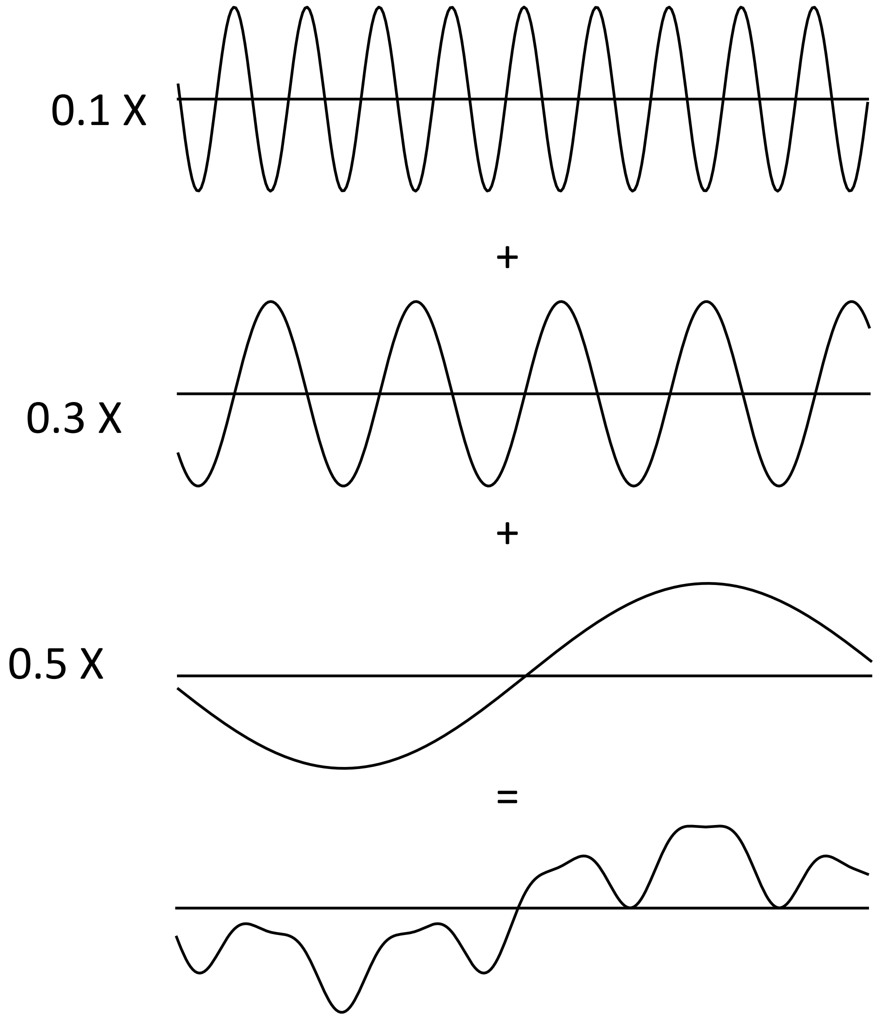
- *Frequency* of a signal is how fast it changes
  - Reflects scale of structure



Frequency

- $x(t) = \cos 2\pi\nu t$
- What is the period?
- What is the frequency?

A combination of frequencies

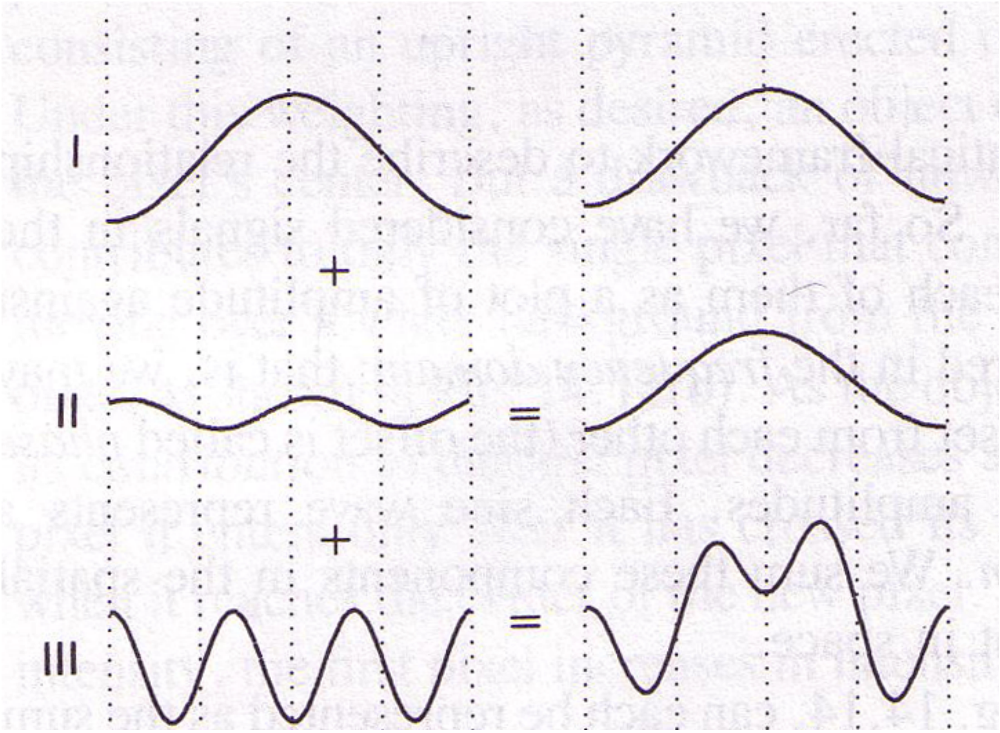


Fourier transform

- 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!*

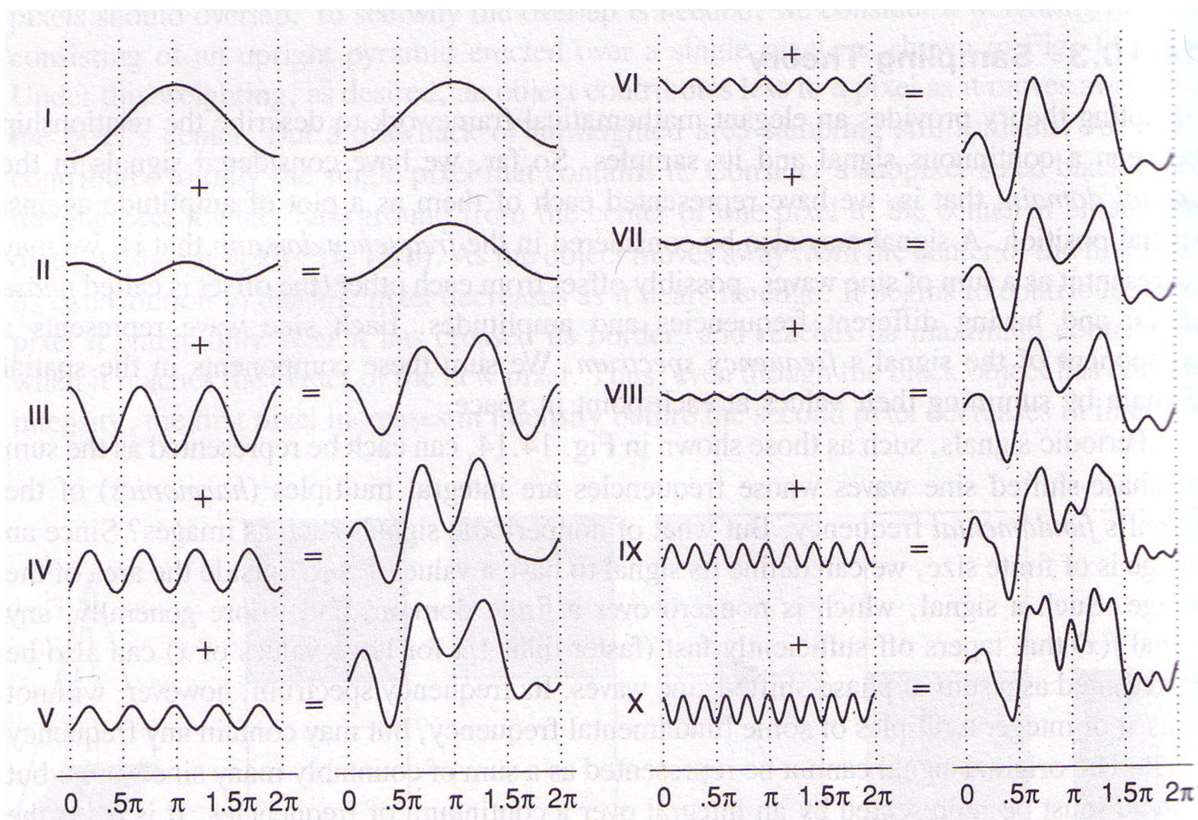
Idea of Fourier Analysis

- Every signal (doesn't matter what it is)
  - Sum of sine/cosine waves



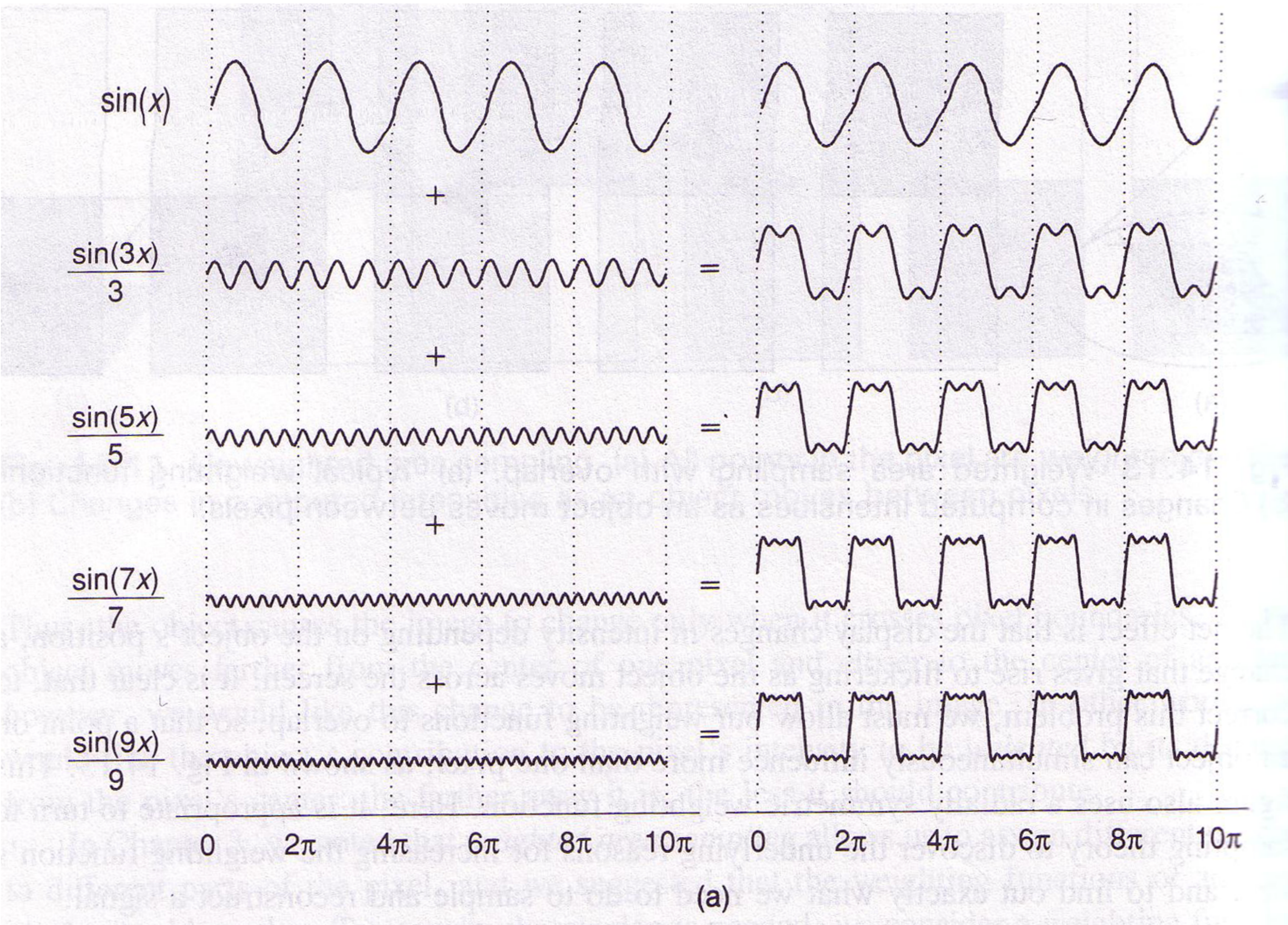
Idea of Fourier Analysis

- Every signal (doesn’t matter what it is)
  - Sum of sine/cosine waves



(b)

A box-like example

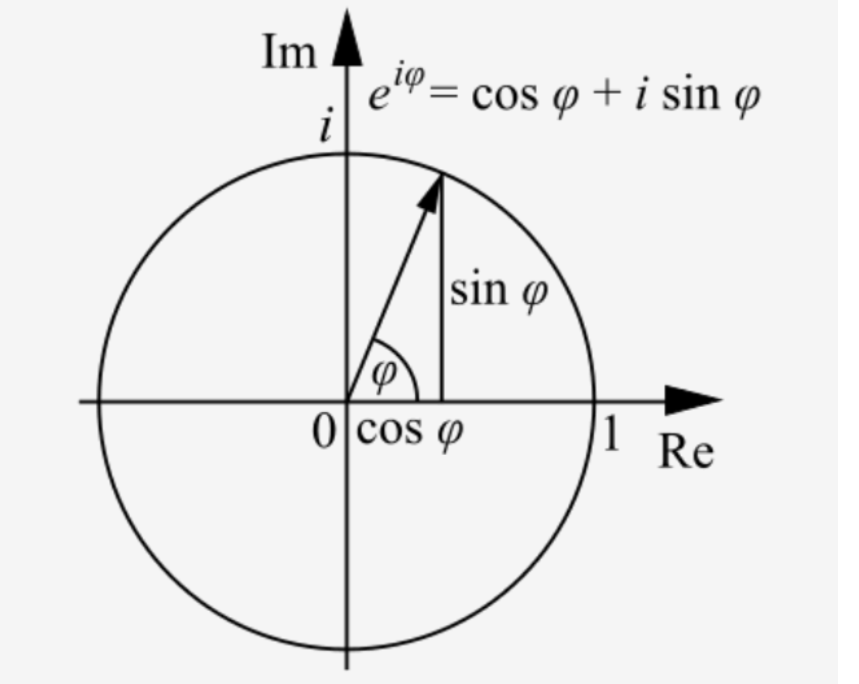


The Fourier bases

- Not exactly sines and cosines, but *complex* variants

$$e^{ix} = \cos(x) + i\sin(x)$$
$$e^{i2\pi\nu t} = \cos(2\pi\nu t) + i\sin(2\pi\nu t)$$

- Euler's formula



$$x(t) = cos(0t) + cos(t) - \frac{1}{3}cos(3t) + \frac{1}{5}cos(5t) - \frac{1}{7}cos(7t)$$

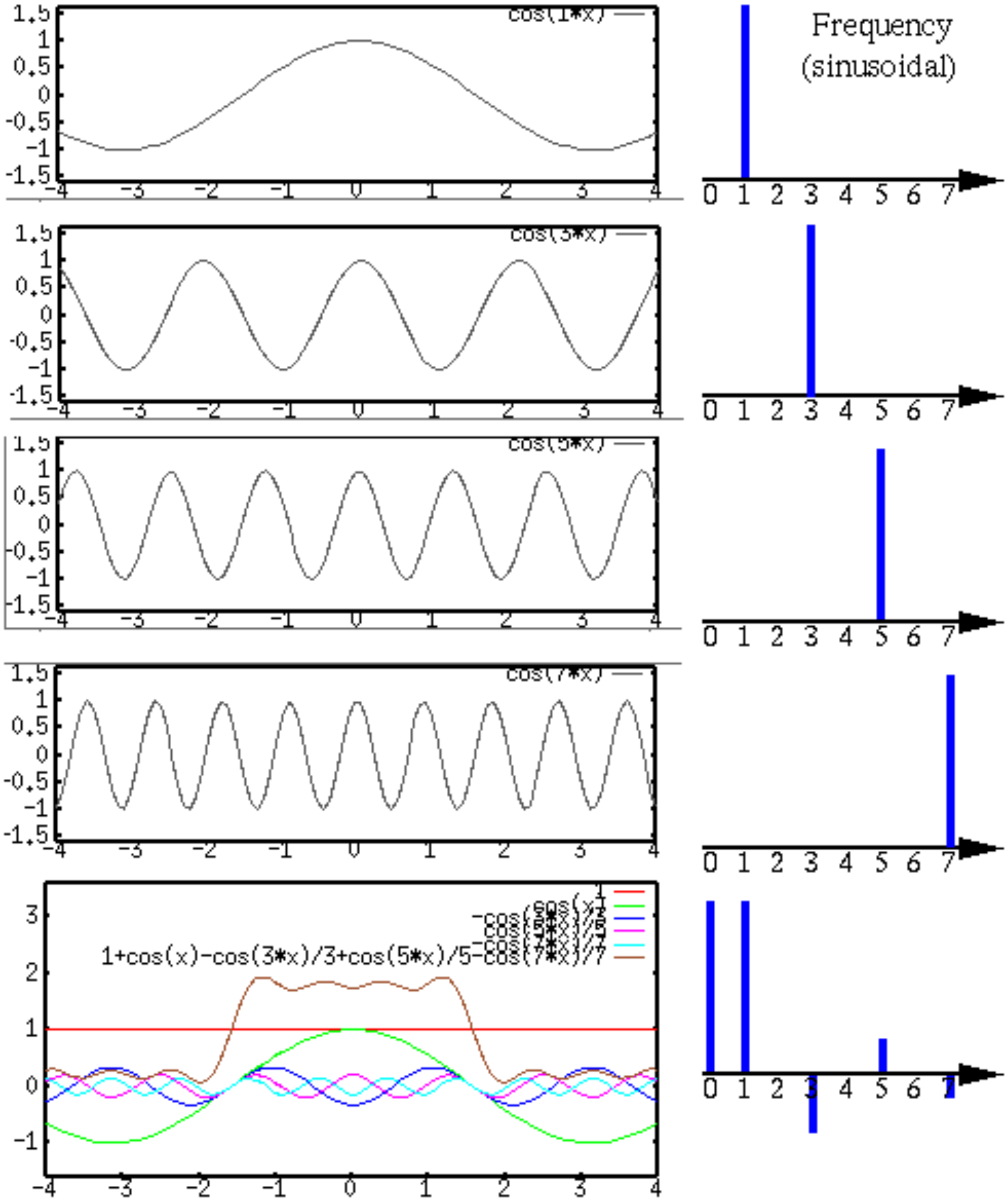
$$\begin{cases} cos(\omega_0 t) = [e^{j\omega_0 t} + e^{-j\omega_0 t}]/2 \\ sin(\omega_0 t) = [e^{j\omega_0 t} - e^{-j\omega_0 t}]/2j \end{cases}$$

i is same as j

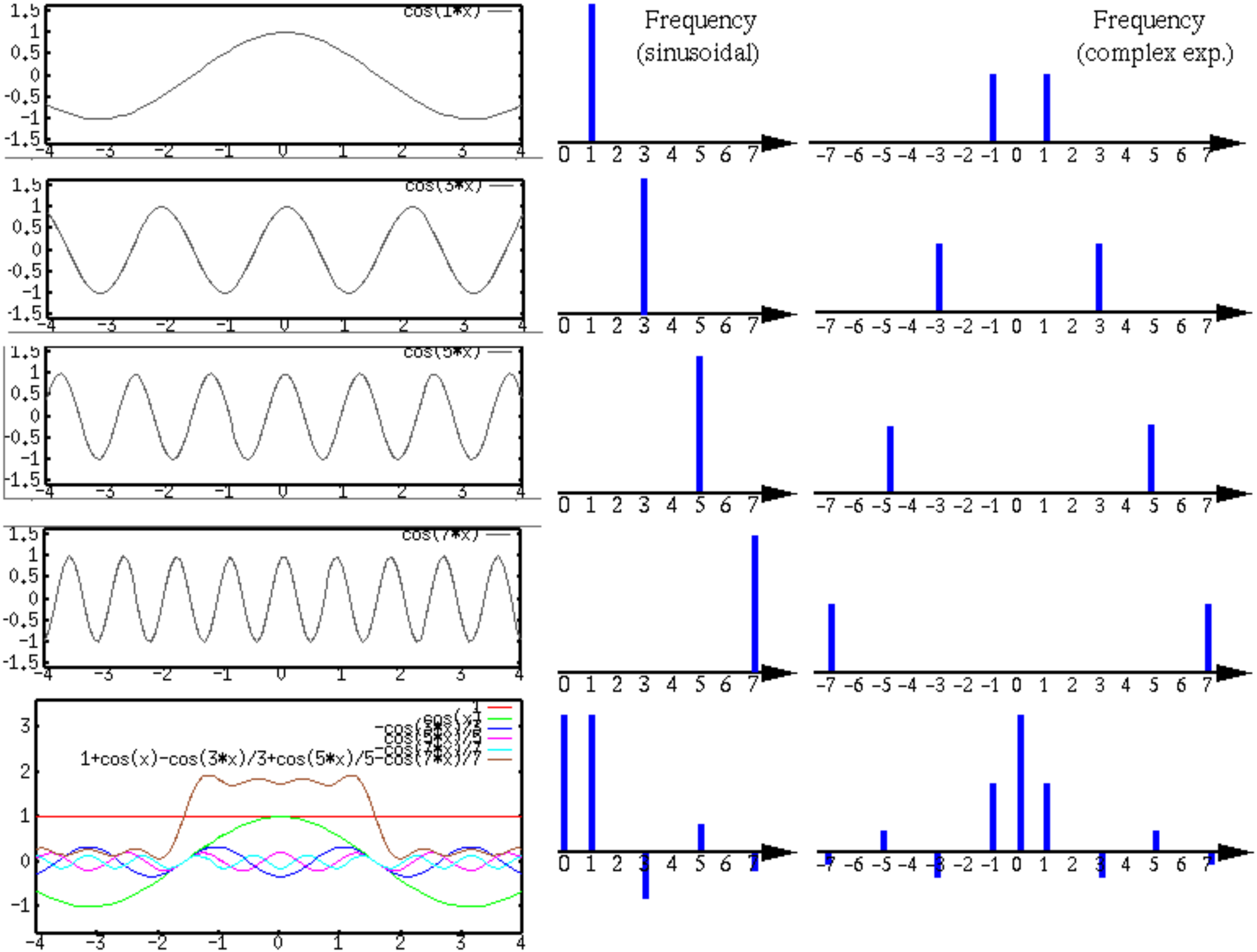
$$x(t) = e^{0t} + \frac{1}{2}[(e^{jt} + e^{-jt}) - \frac{1}{3}(e^{j3t} + e^{-j3t}) + \frac{1}{5}(e^{j5t} + e^{-j5t}) - \frac{1}{7}(e^{j7t} + e^{-j7t})] = \sum_{k=-7}^{7} X[k]e^{jk\omega_0 t}$$

$$X[0] = 0; \;\; X[1] = X[-1] = 1/2, \;\; X[3] = X[-3] = 1/6, \;\; X[5] = X[-5] = 1/10,$$

$$X[7] = X[-7] = 1/14, \;\; X[2] = X[-2] = X[4] = X[-4] = X[6] = X[-6] = 0$$



Frequency
(sinusoidal)
cos(1*x)
cos(3*x)
cos(5*x)
cos(7*x)
1
cos(x)
-cos(3*x)/3
cos(5*x)/5
-cos(7*x)/7
1+cos(x)-cos(3*x)/3+cos(5*x)/5-cos(7*x)/7
0 1 2 3 4 5 6 7


cos(1*x)
cos(3*x)
cos(5*x)
cos(7*x)
1
cos(x)
-cos(3*x)/3
cos(5*x)/5
-cos(7*x)/7
1+cos(x)-cos(3*x)/3+cos(5*x)/5-cos(7*x)/7
Frequency
(sinusoidal)
Frequency
(complex exp.)

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)*

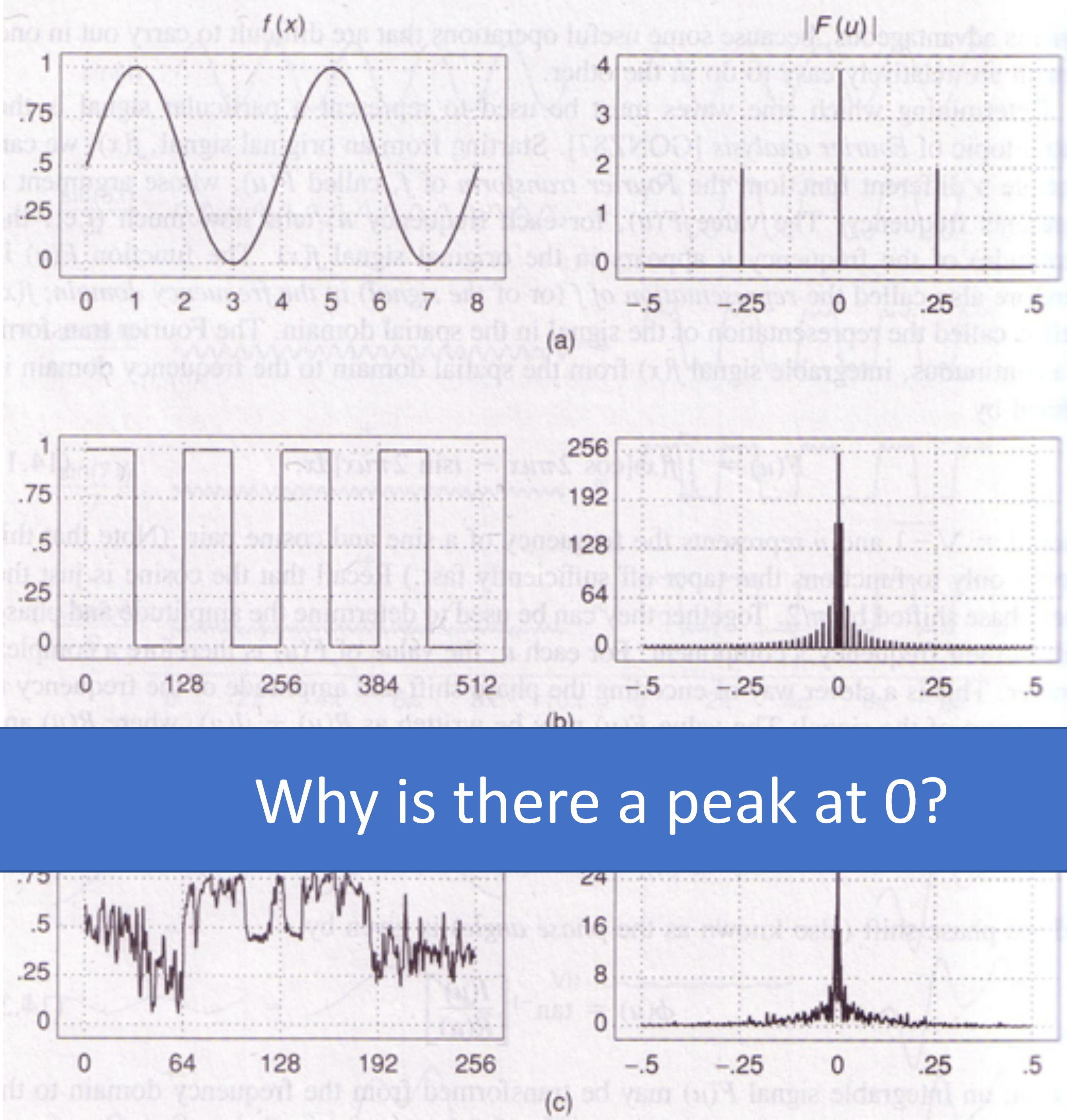
Fourier transform

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

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

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)

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

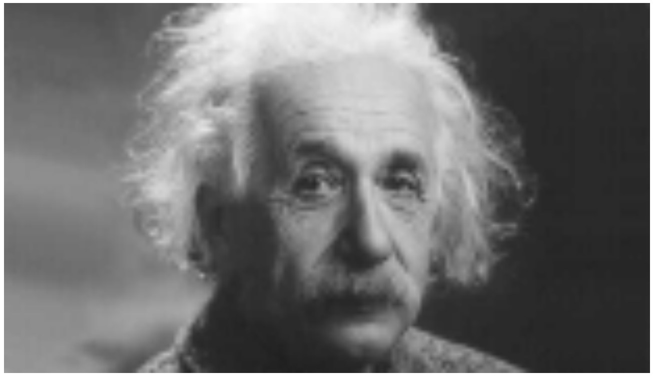
Properties of Fourier transforms

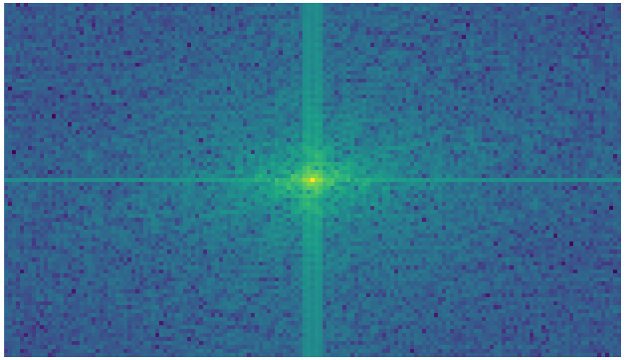
Property	Signal		Transform
superposition	$f_1(x) + f_2(x)$		$F_1(\omega) + F_2(\omega)$
shift	$f(x - x_0)$		$F(\omega)e^{-j\omega x_0}$
reversal	$f(-x)$		$F^*(\omega)$
convolution	$f(x) * h(x)$		$F(\omega)H(\omega)$
correlation	$f(x) \otimes h(x)$		$F(\omega)H^*(\omega)$
multiplication	$f(x)h(x)$		$F(\omega) * H(\omega)$
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

- Sine → Impulse
- Gaussian → Gaussian
- Box → Sinc

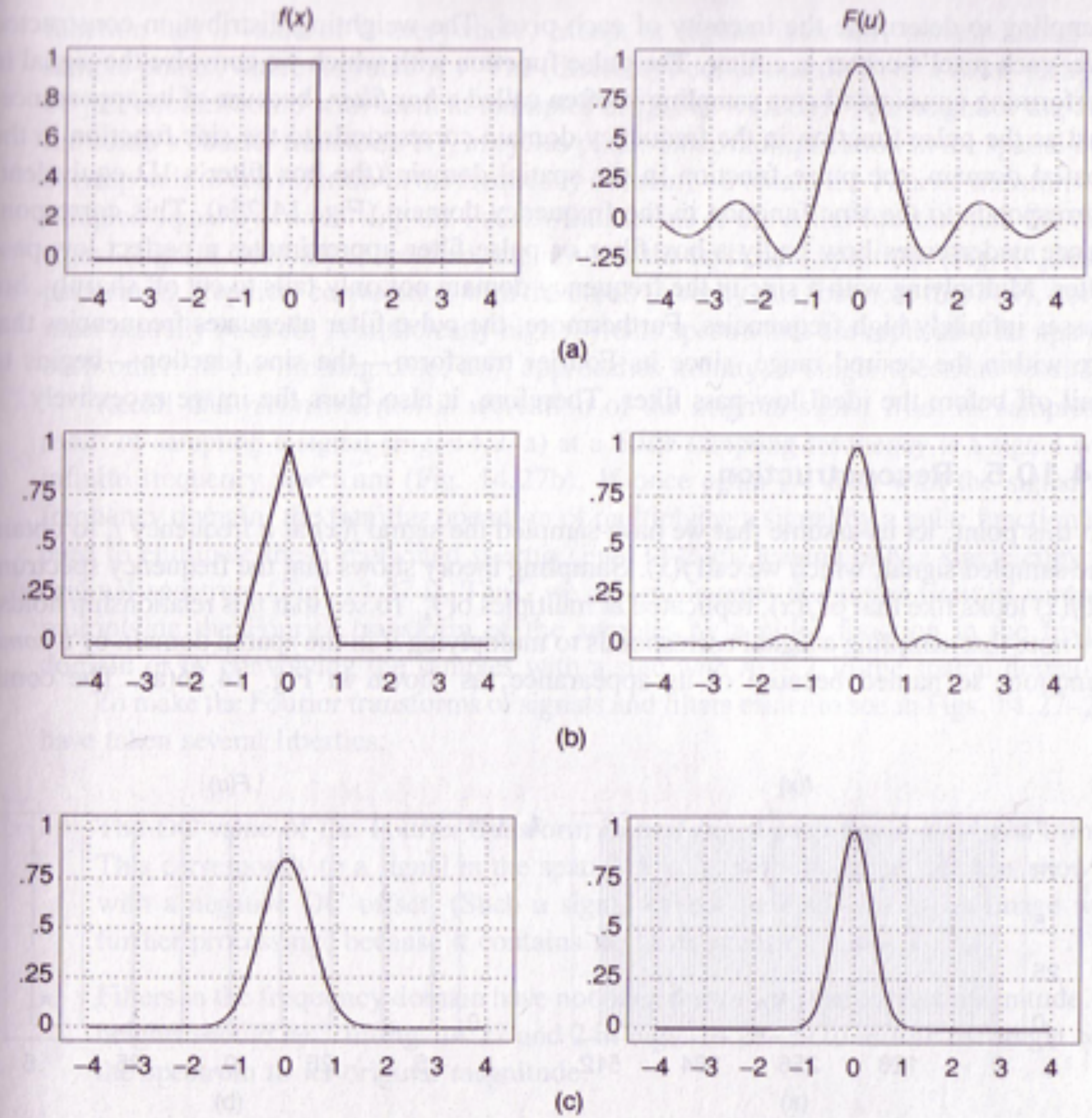
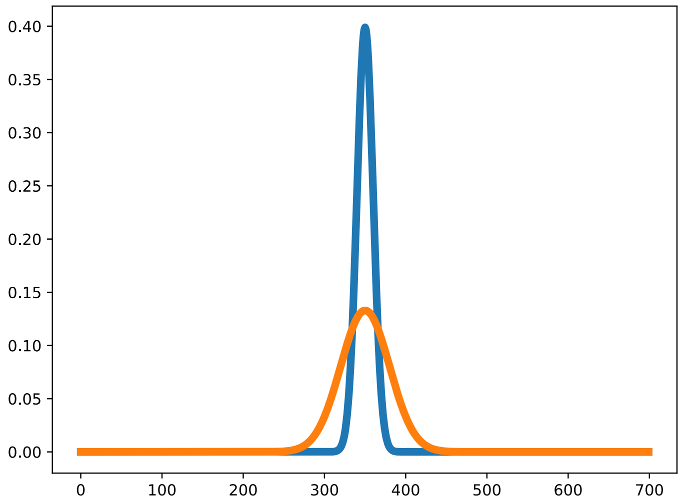
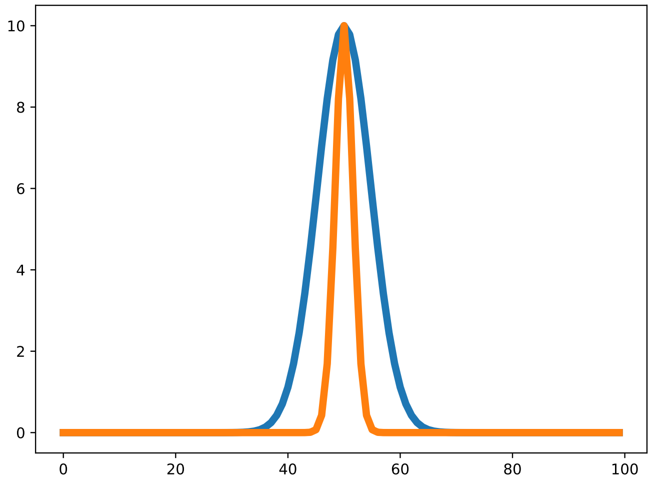


Fig. 14.25 Filters in spatial and frequency domains. (a) Pulse—sinc. (b) Triangle—sinc^2. (c) Gaussian—Gaussian. (Courtesy of George Wolberg, Columbia University.)

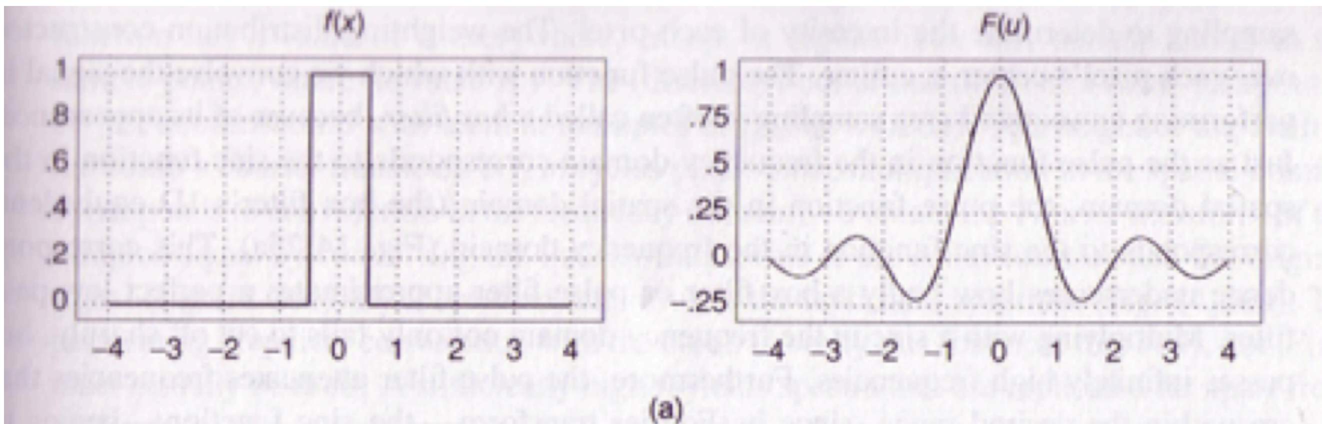
The Gaussian special case

- 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?

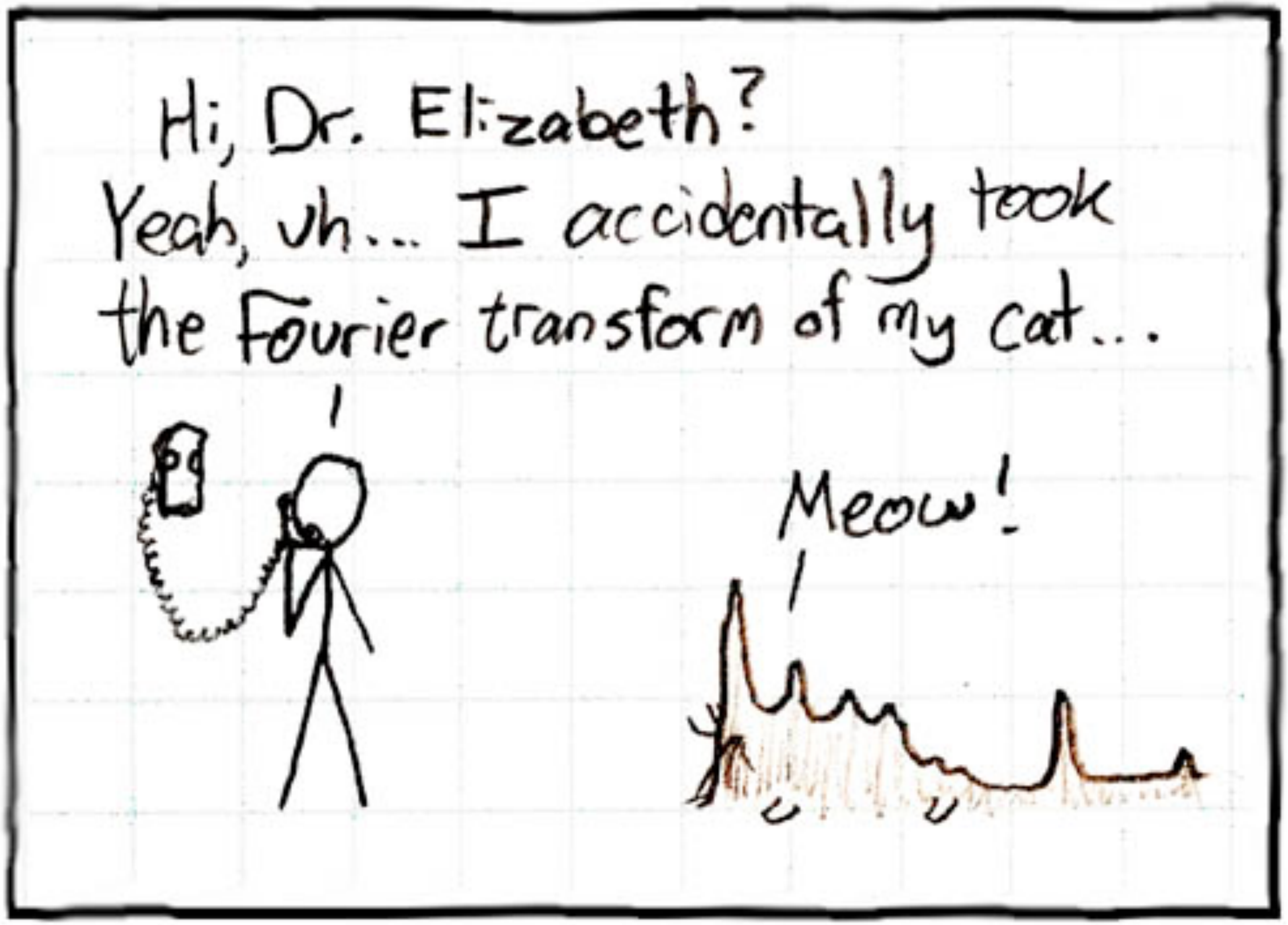
- Convolution is point-wise multiplication in frequency space
  - Analyze which frequency components a particular filter lets through, e.g., *low-pass, high-pass* or *band-pass* filters
  - Leads to fast algorithms for convolution with large filters: Fast FFT

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/