COSC428 Computer Vision

Filters

What is image filtering?

- Modify the pixels in an image based on some function of a local neighborhood of the pixels.

10	5	3
4	5	1
1	1	7

Local image data

Some function $\longrightarrow$

	7	

Modified image data

Linear functions

- Simplest: linear filtering.
  - Replace each pixel by a linear combination of its neighbors.
- The prescription for the linear combination is called the “convolution kernel”.

10	5	3
4	5	1
1	1	7

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

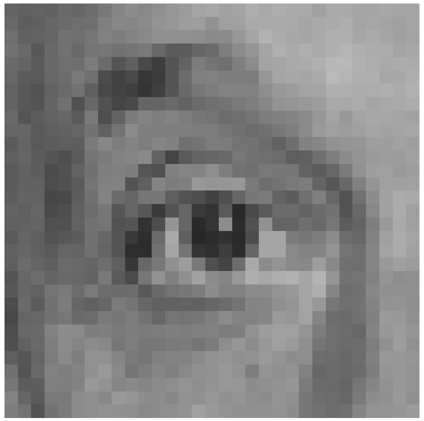
	7	

Modified image data

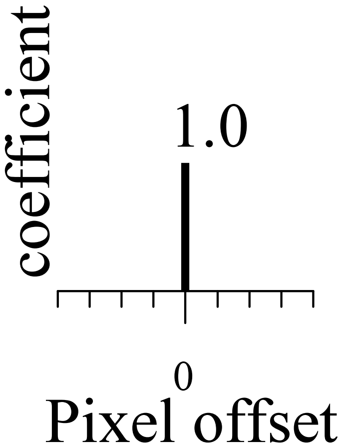
Convolution

$$f[m,n] = I \otimes g = \sum_{k,l} I[m-k, n-l]\, g[k,l]$$

Linear filtering (warm-up slide)

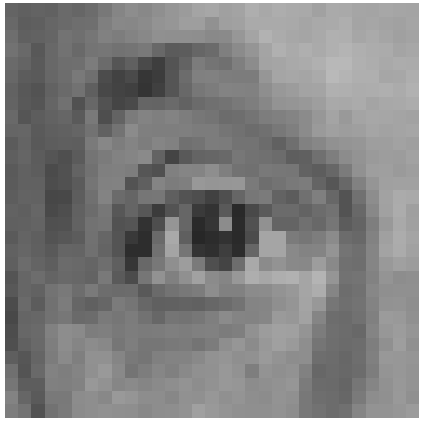


original

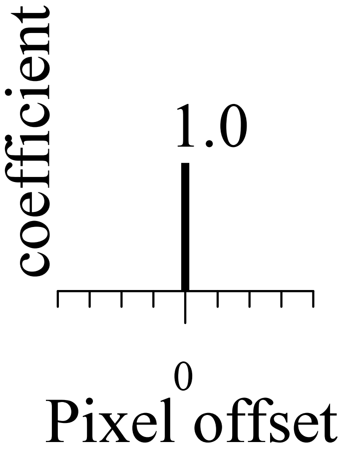


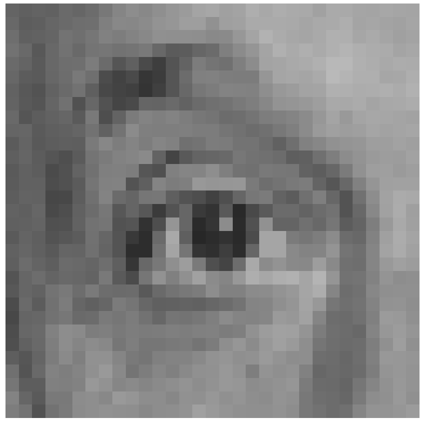
?

Linear filtering (warm-up slide)



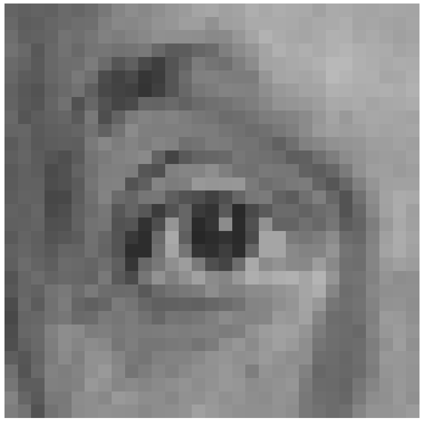
original



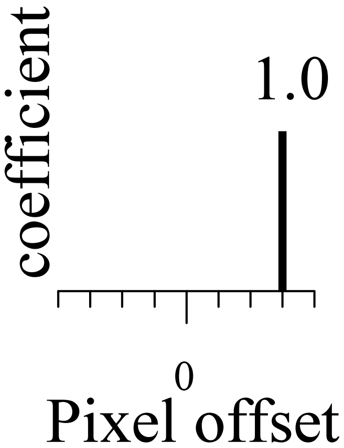


Filtered
(no change)

Linear filtering

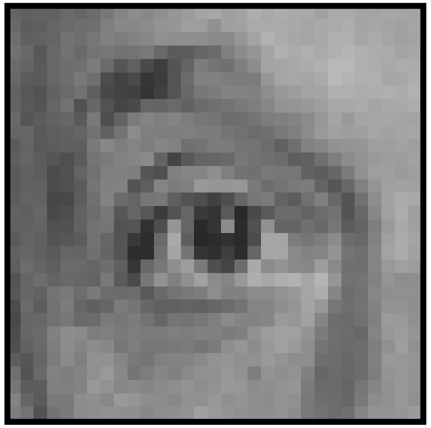


original

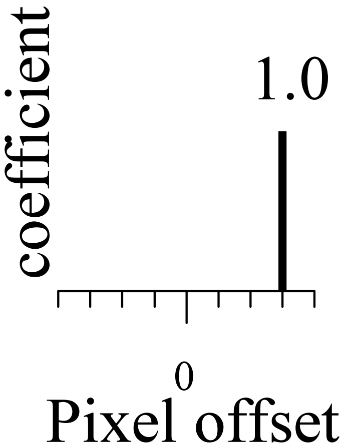


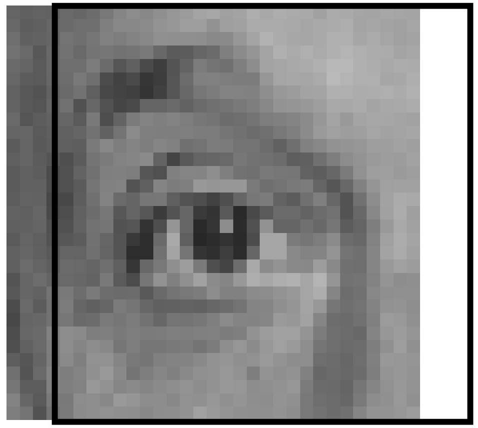
?

shift



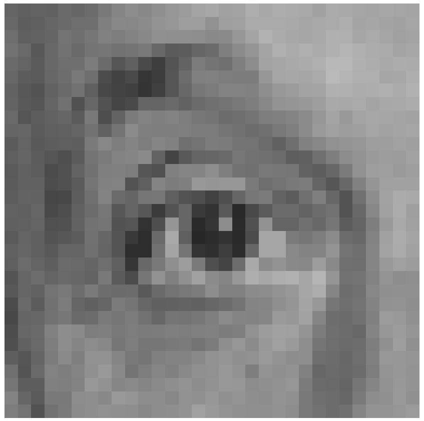
original





shifted

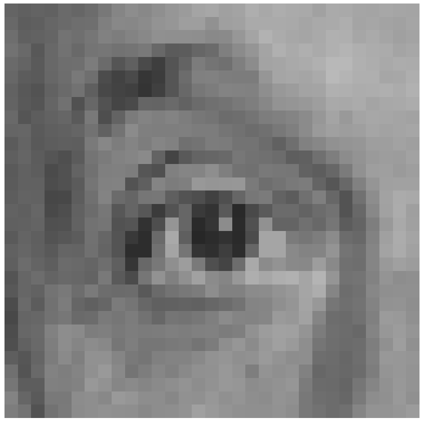
Linear filtering



original

?

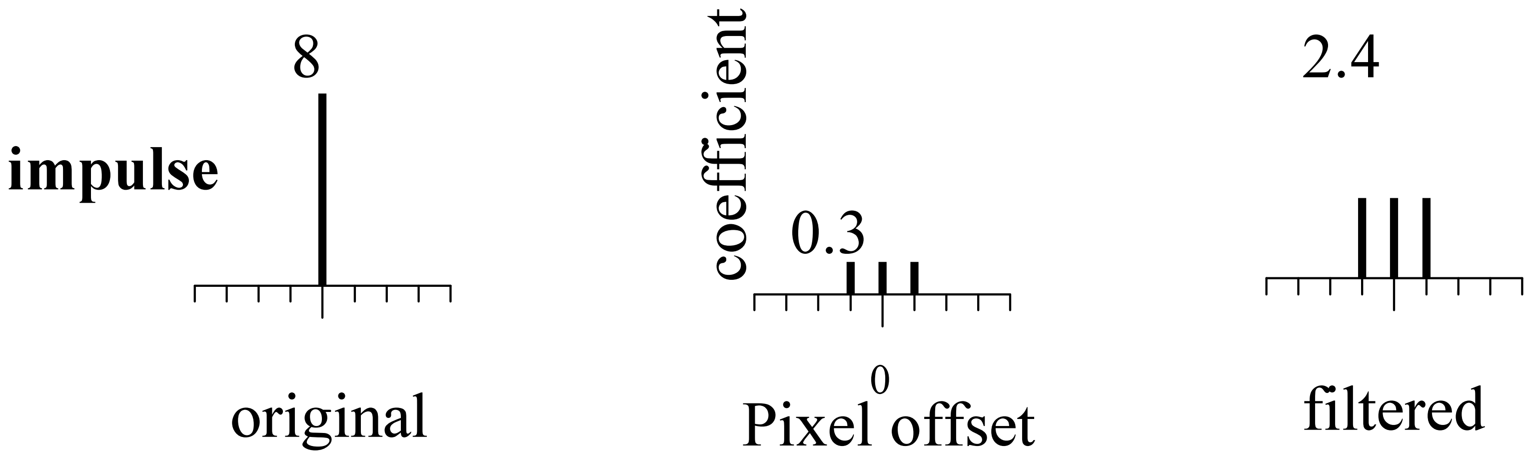
Blurring



original

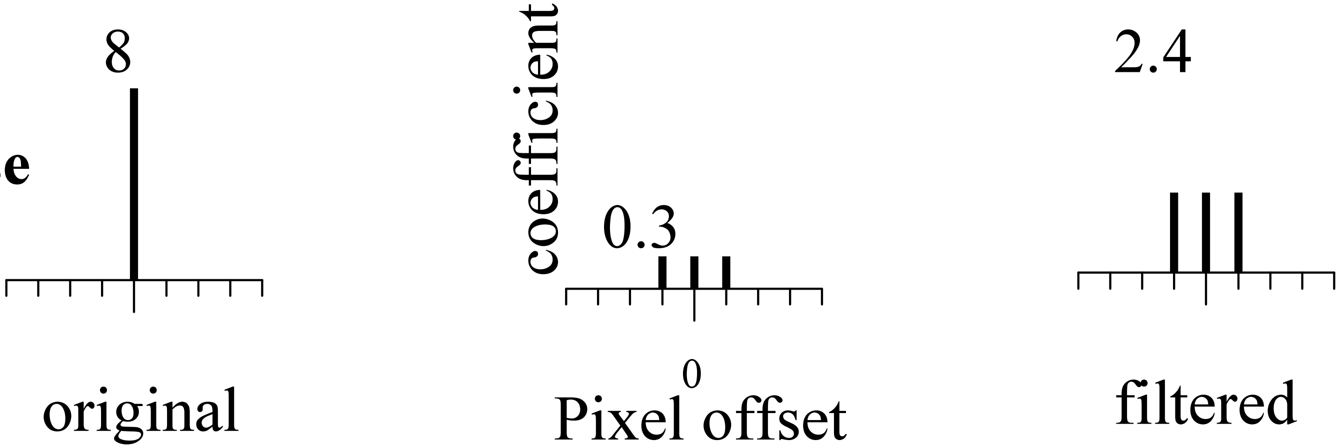
Blurred (filter applied in both dimensions).

Blur examples

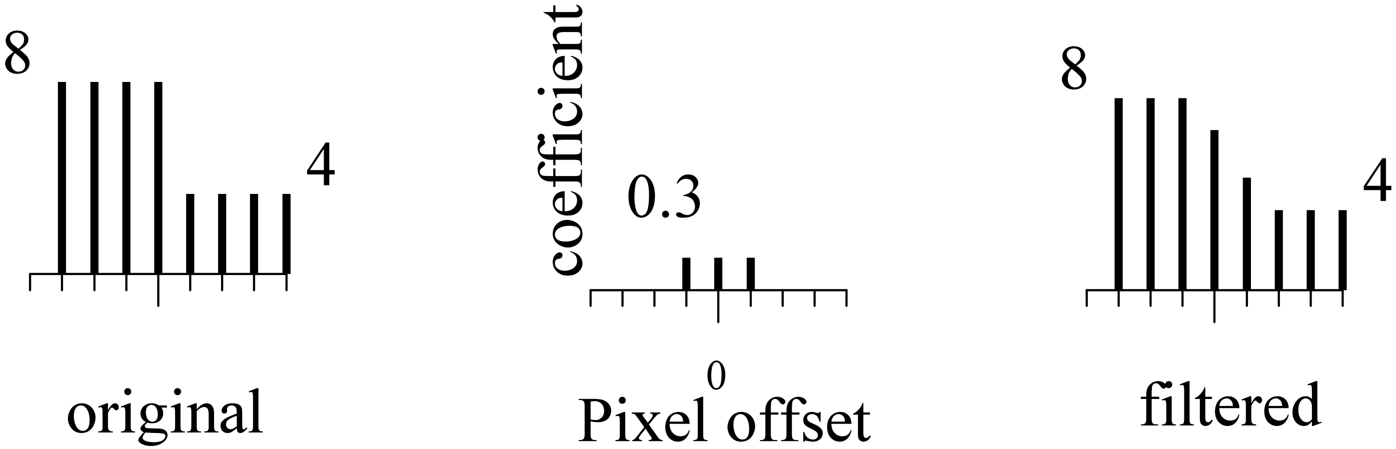


Blur examples

impulse

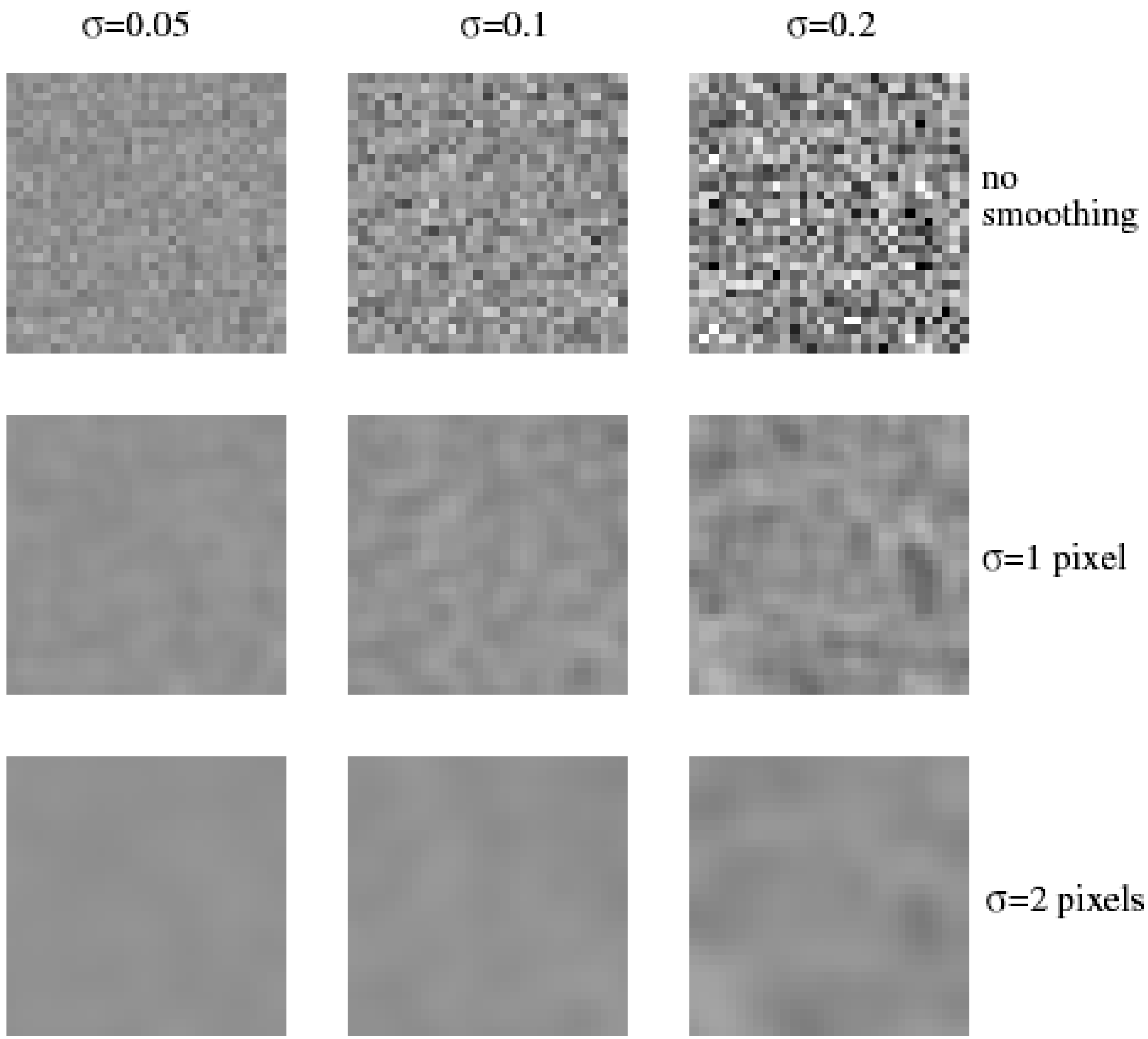


edge



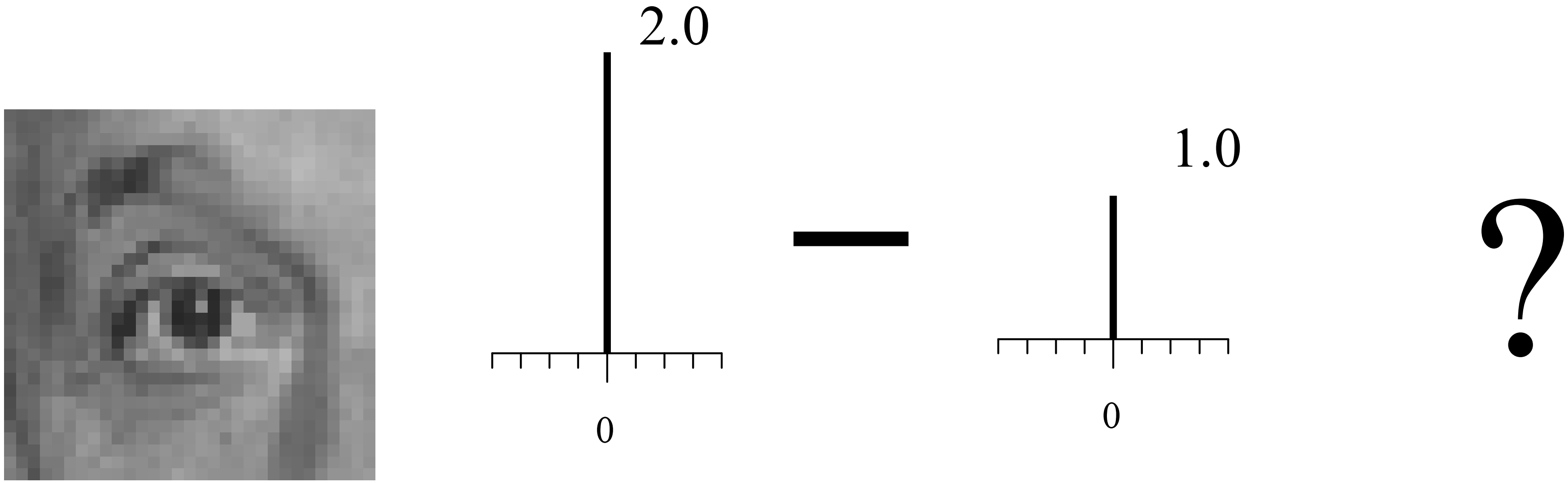
Smoothing reduces noise

- Generally expect pixels to "be like" their neighbours
  - surfaces turn slowly
  - relatively few reflectance changes
- Generally expect noise processes to be independent from pixel to pixel
- Implies that smoothing suppresses noise, for appropriate noise models
- Scale
  - the parameter in the symmetric Gaussian
  - as this parameter goes up, more pixels are involved in the average
  - and the image gets more blurred
  - and noise is more effectively suppressed



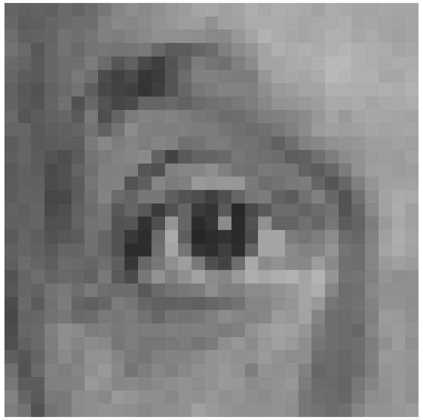
The effects of smoothing
Each row shows smoothing with gaussians of different width; each column shows different realisations of an image of gaussian noise.

Linear filtering (warm-up slide)

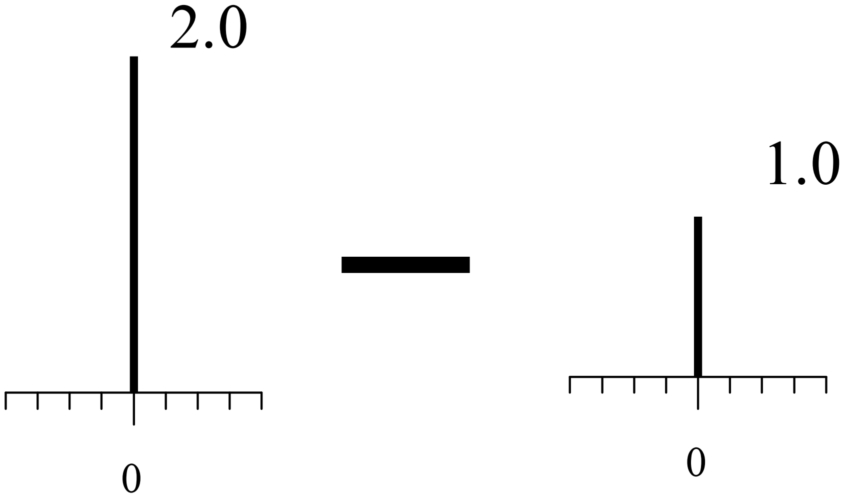


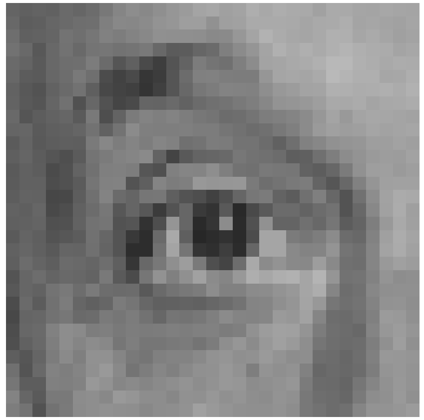
original

Linear filtering (no change)



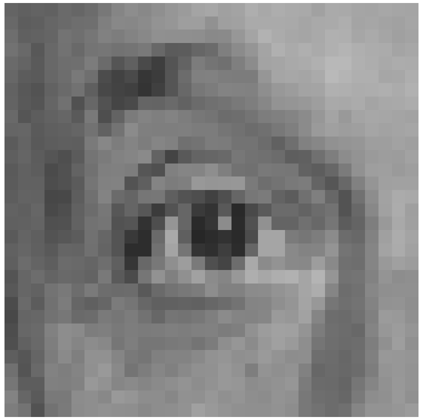
original



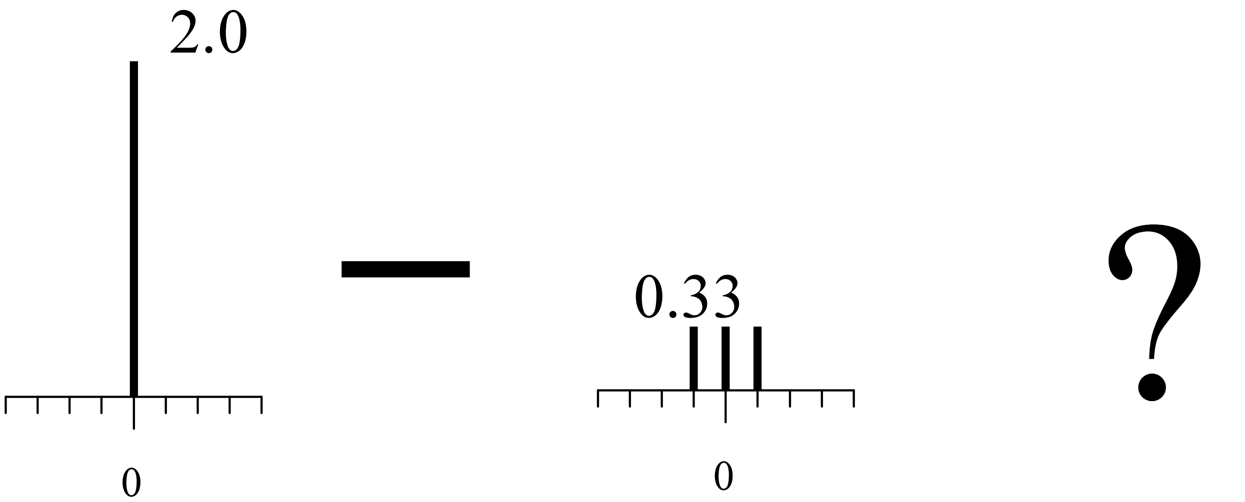


Filtered
(no change)

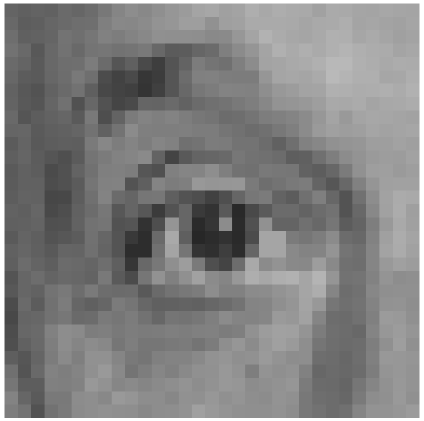
Linear filtering



original



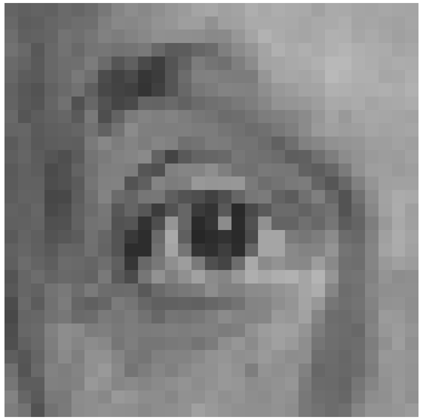
(remember blurring)



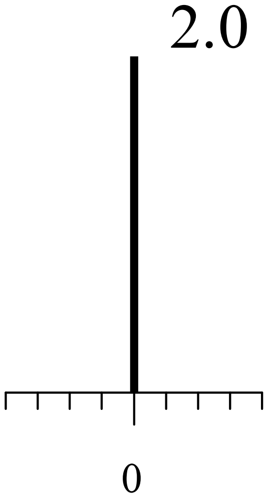
original

Blurred (filter applied in both dimensions).

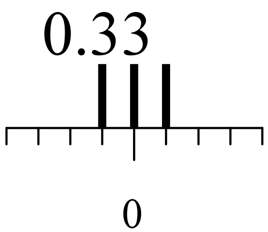
Linear filtering



original

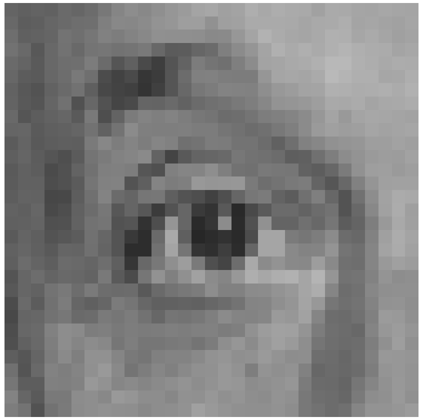


−

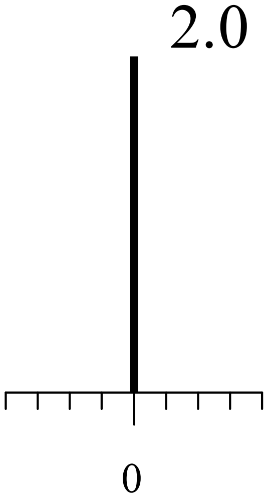


?

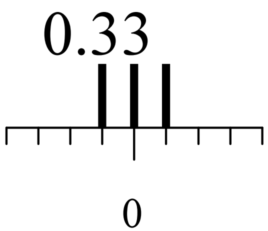
Sharpening

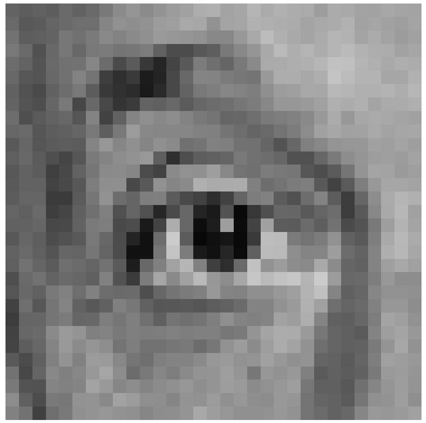


original



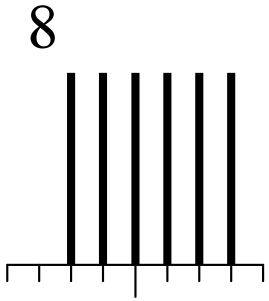
−



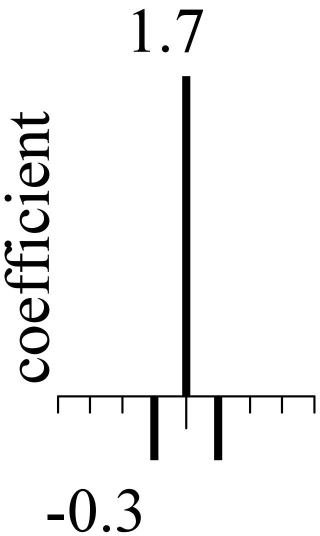


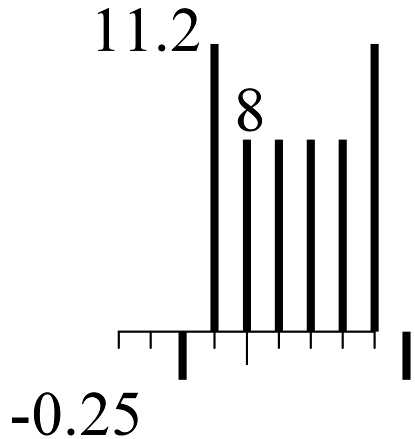
Sharpened
original

Sharpening example



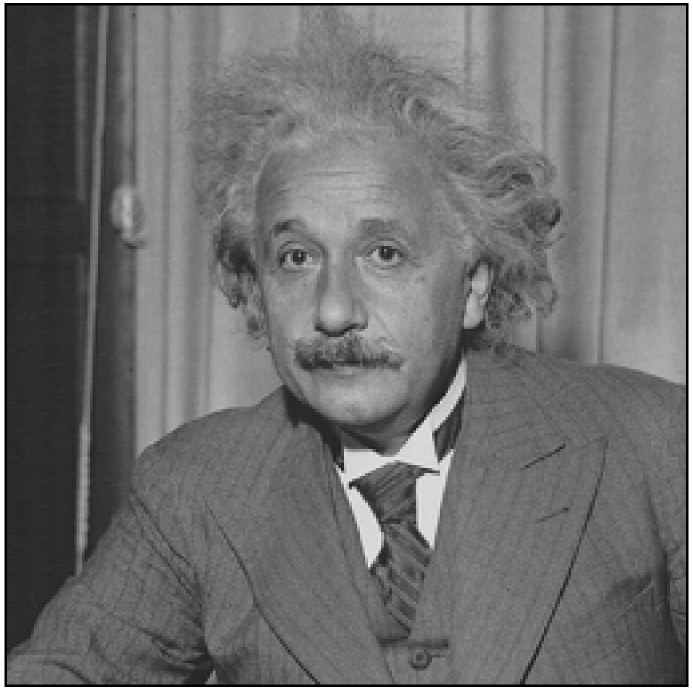
original



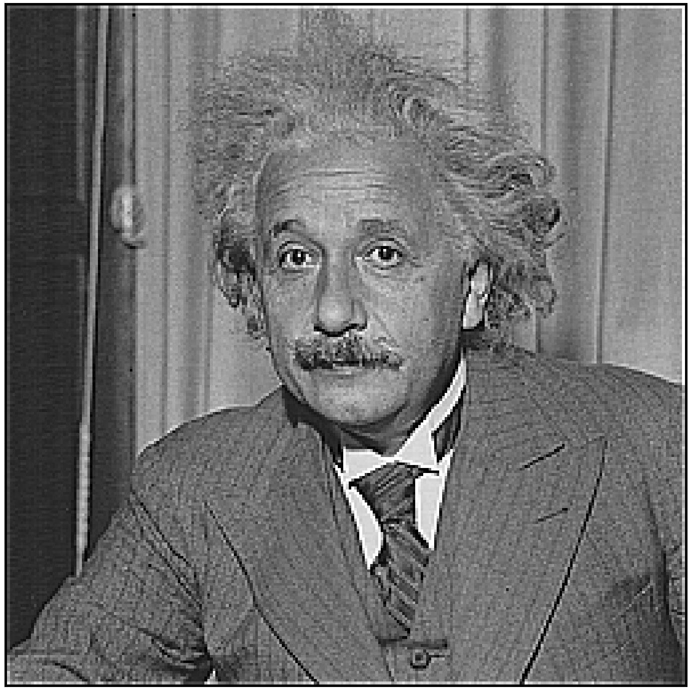


Sharpened
(differences are
accentuated; constant
areas are left untouched).

Sharpening



before



after

Gradients and edges

- Points of sharp change in an image are interesting:
  - change in reflectance
  - change in object
  - change in illumination
  - noise
- Sometimes called **edge points**

- General strategy
  - linear filters to estimate image gradient
  - mark points where gradient magnitude is particularly large wrt neighbours (ideally, curves of such points).

Linear image transformations

- In analyzing images, it's often useful to make a change of basis.

transformed image

$$\vec{F} = U\vec{f}$$

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

Self-inverting transforms

Same basis functions are used for the inverse transform

$$\vec{f} = U^{-1}\vec{F}$$

$$= U^{+}\vec{F}$$

U transpose and complex conjugate

An example of such a transform: the Fourier transform

discrete domain

Forward transform

$$F[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k,l] e^{-\pi i \left(\frac{km}{M} + \frac{\ln}{N} \right)}$$

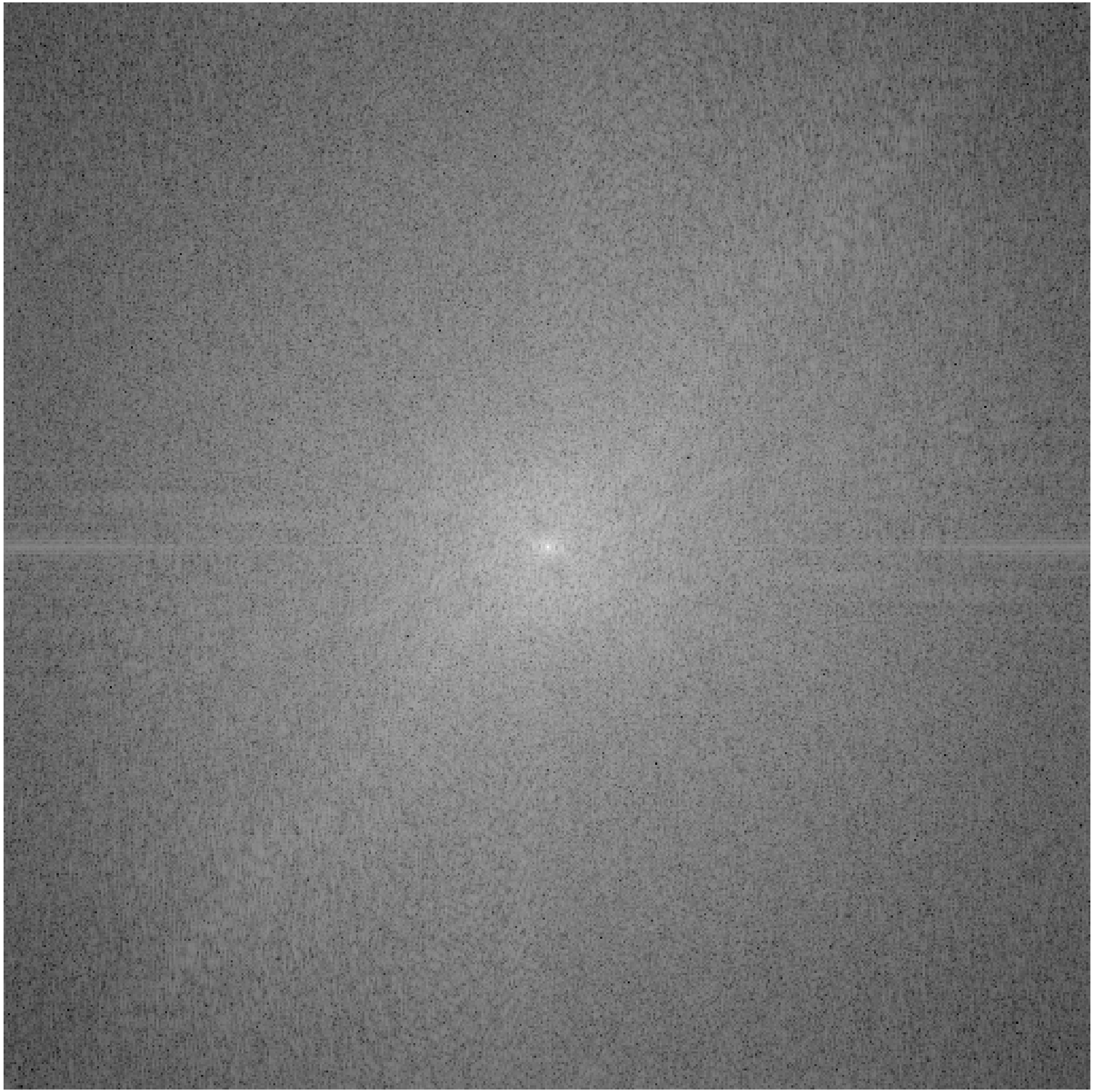
Inverse transform

$$f[k,l] = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[m,n] e^{+\pi i \left(\frac{km}{M} + \frac{\ln}{N} \right)}$$

Phase and Magnitude

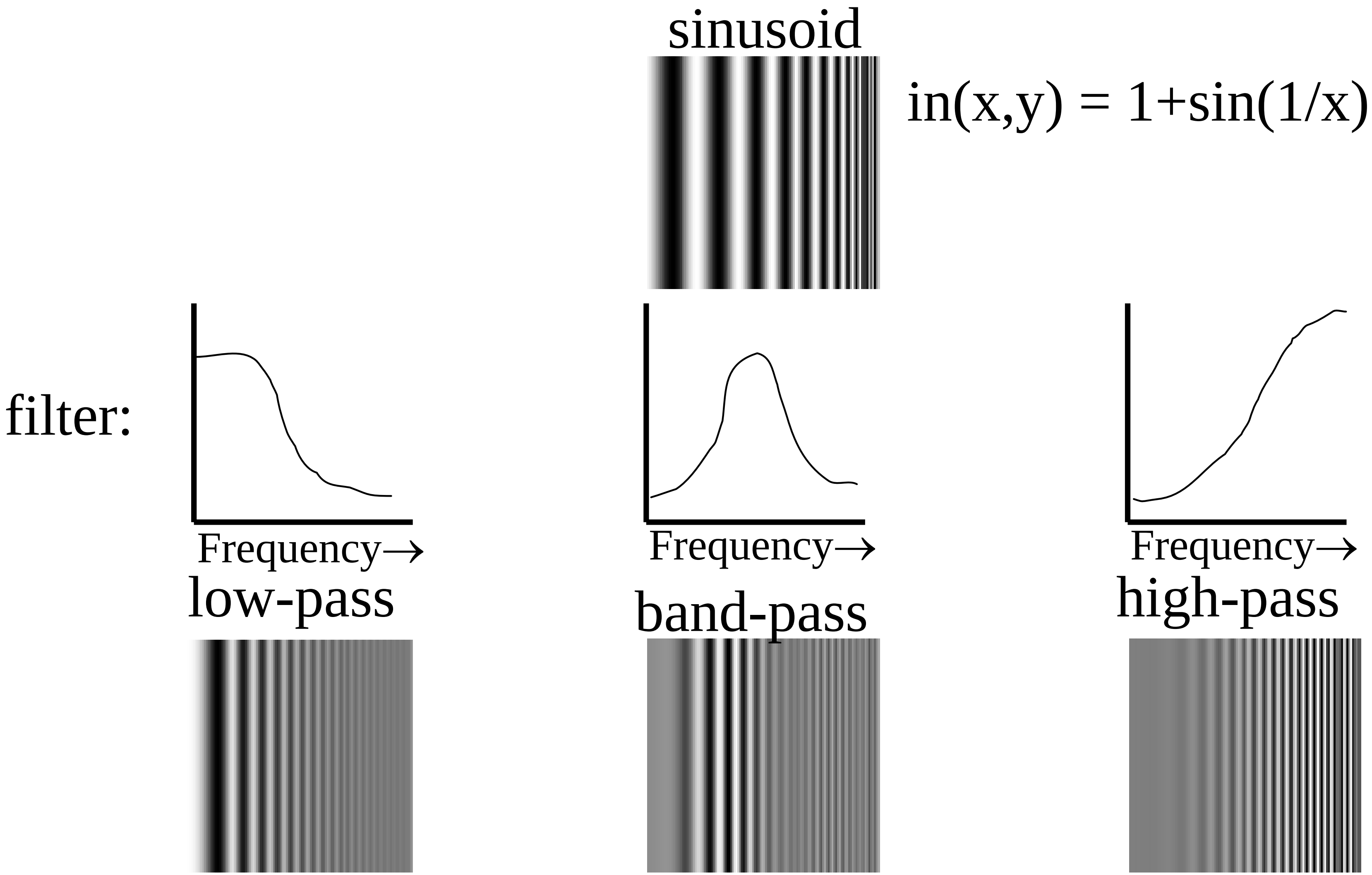
- Fourier transform of a real function is complex
  - difficult to plot, visualize
  - instead, we can think of the phase and magnitude of the transform
- Phase is the phase of the complex transform
- Magnitude is the magnitude of the complex transform
- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn’t
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

This is the magnitude transform of the cheetah pic

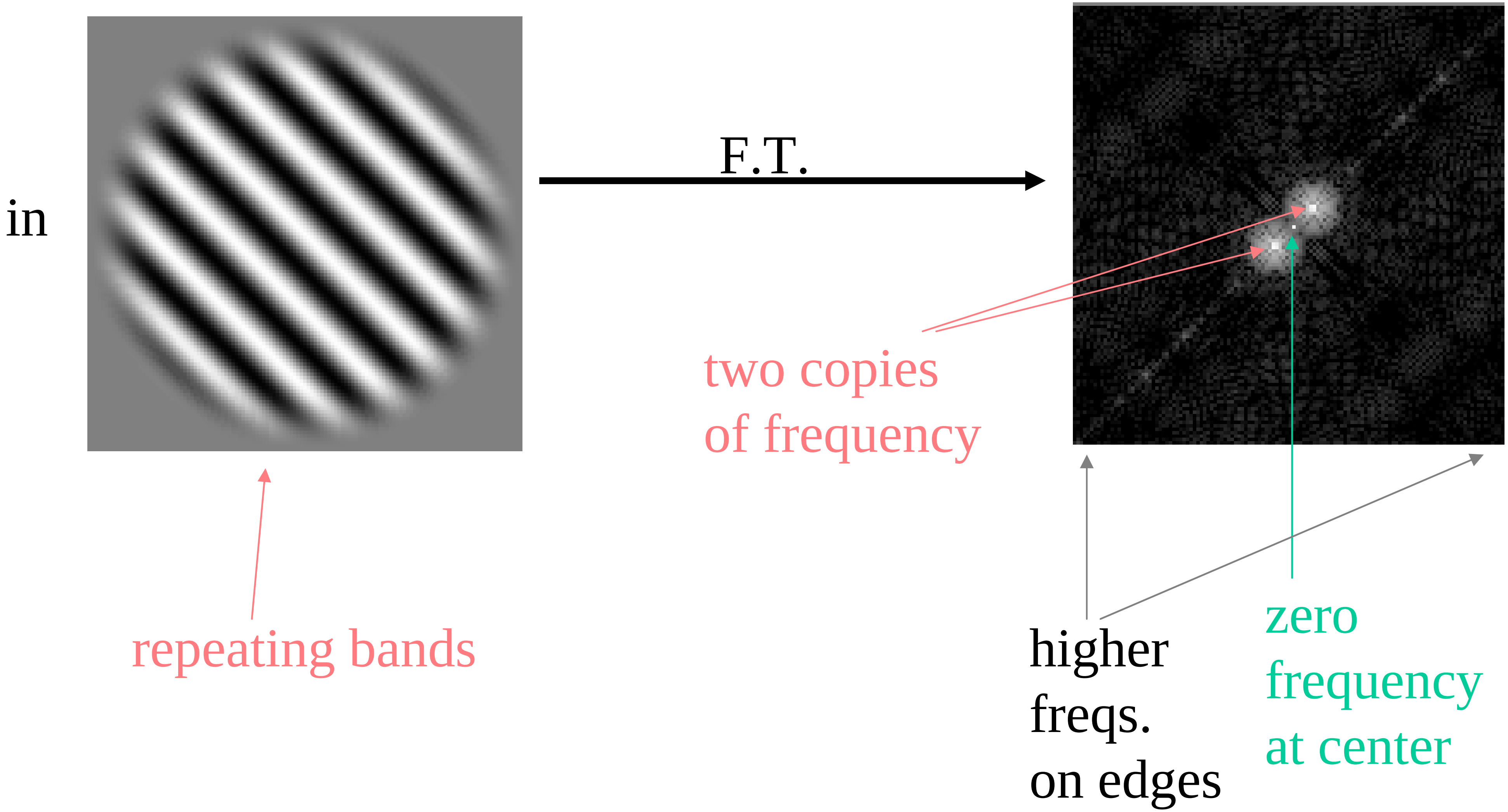


This is the magnitude transform of the zebra pic

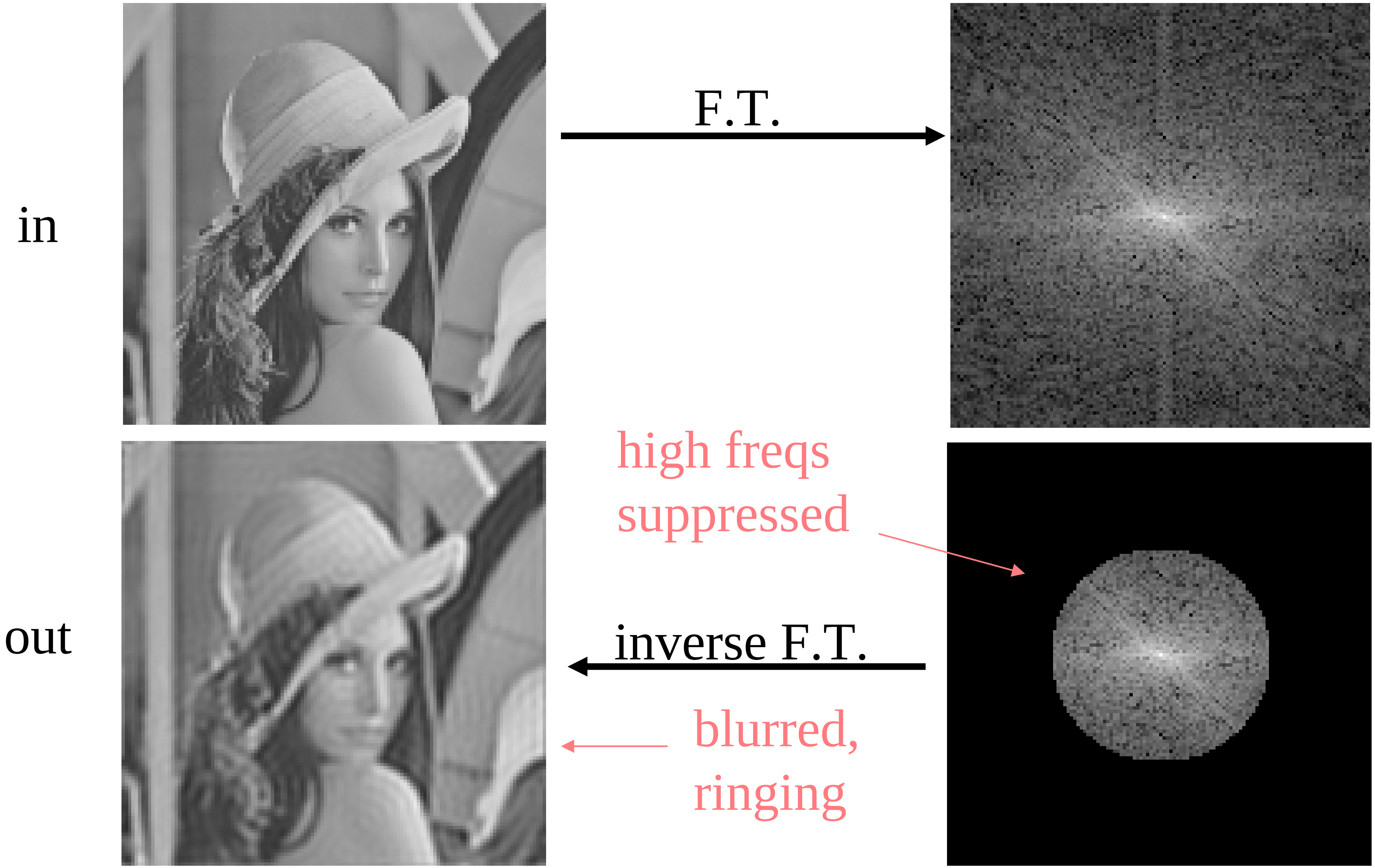
Frequency filtering



Fourier Transform



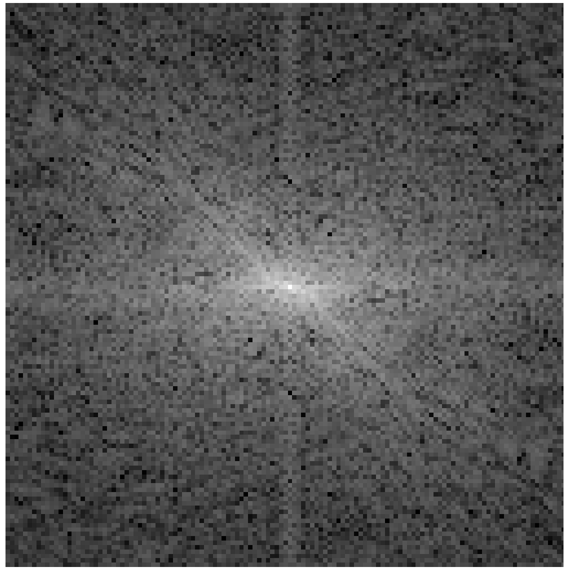
Fourier Low-pass filter



Fourier High-pass filter

in

F.T.



out

low freqs
suppressed

inverse F.T.

edges
detected