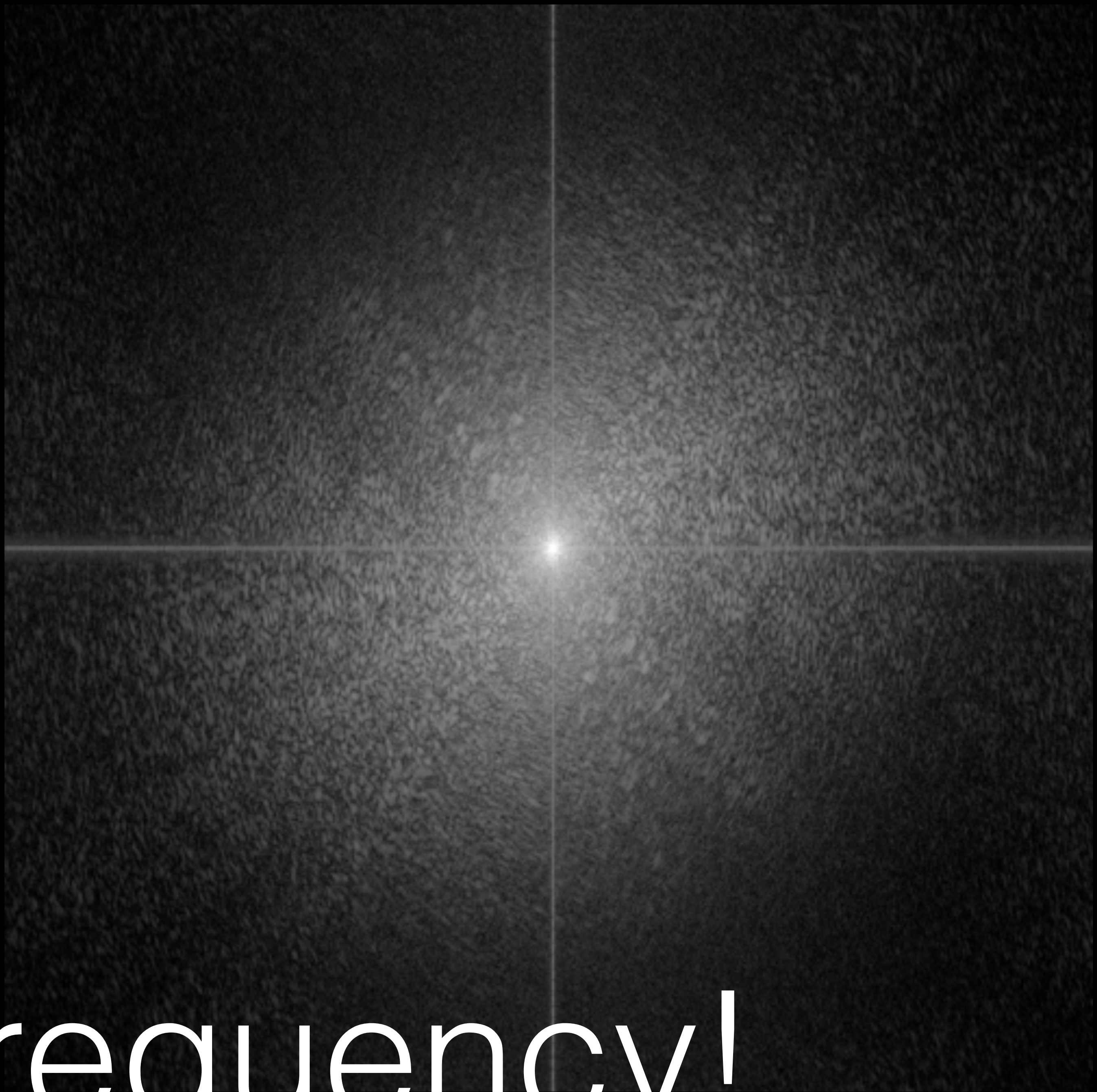
Digital Image Processing

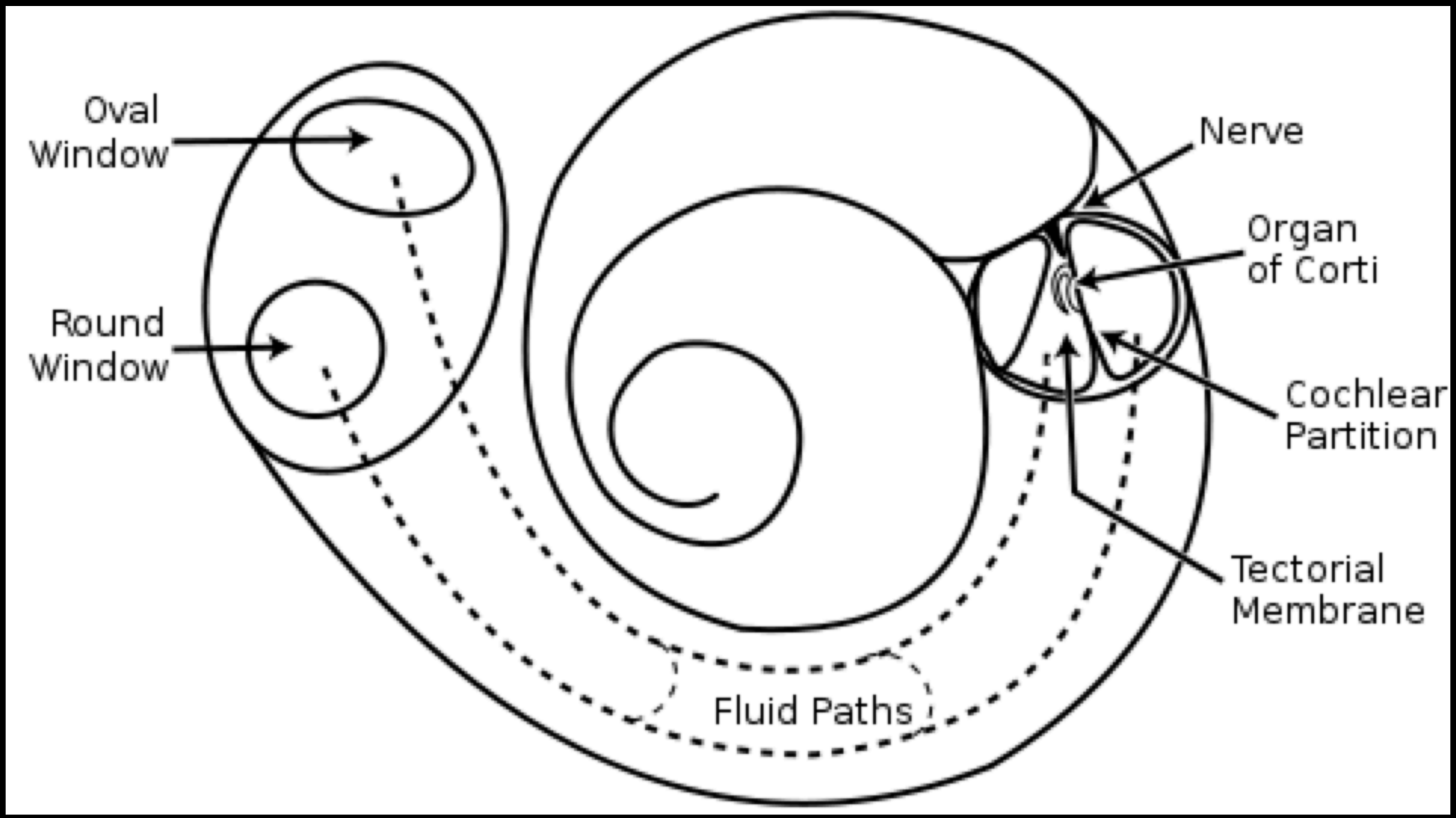
- Today: Chapter 2 from the course text book (Gonzales and Woods, Digital Image Processing)
- Linear and nonlinear operations
- Interpolation
- Geometrical transformations (Affine transformations)
- The Fourier Transform

Transform to Fourier (Frequency) space



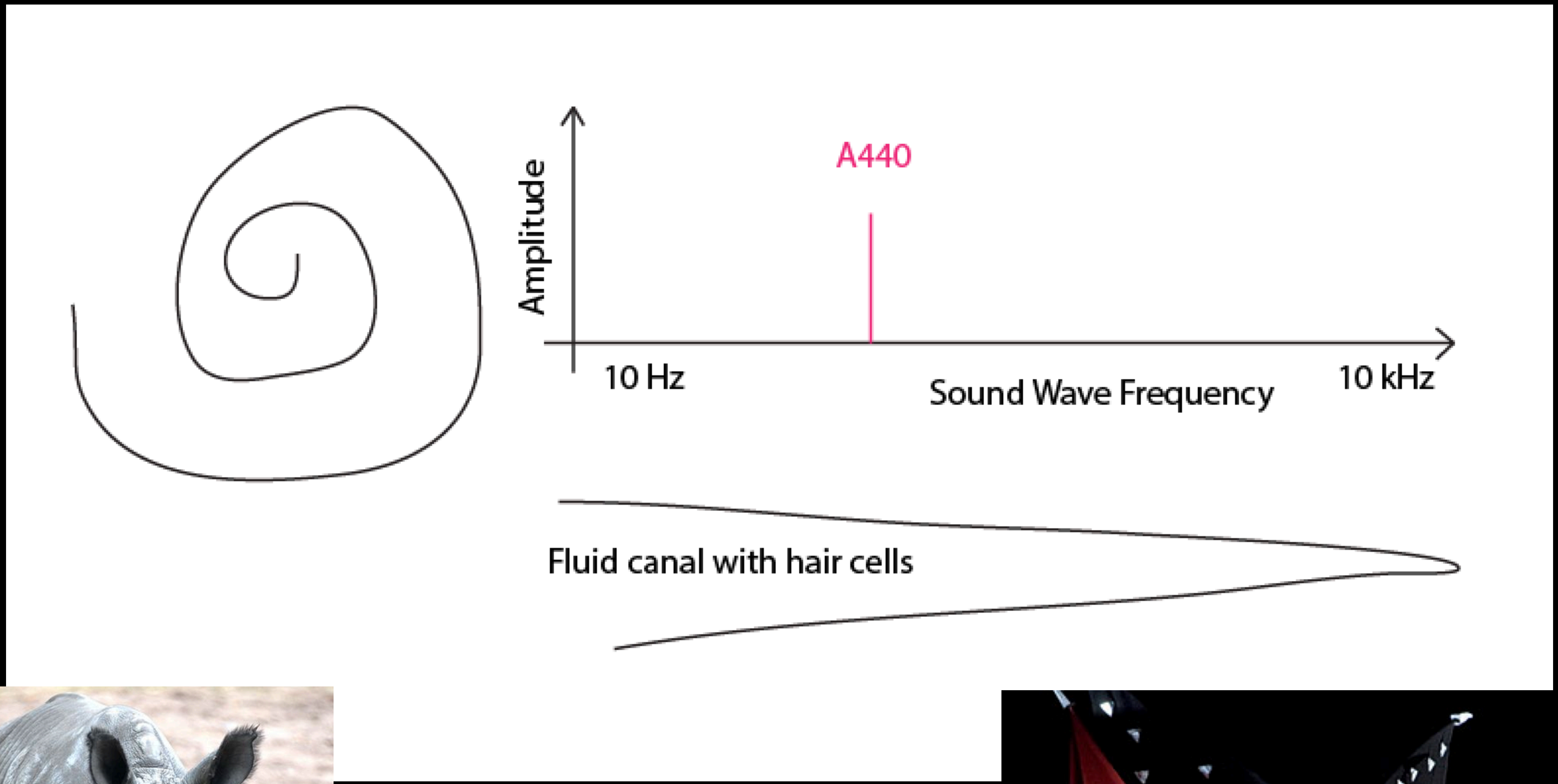
Spatial Frequency!

Lets start with 1D



- Cochlea of the human ear

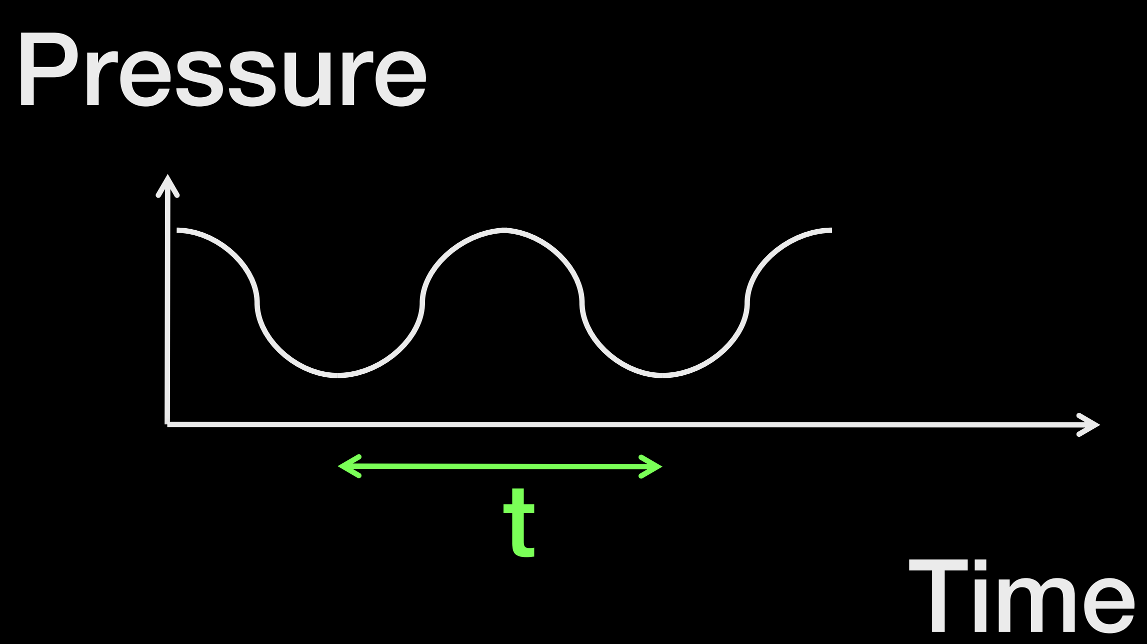
Frequency plot

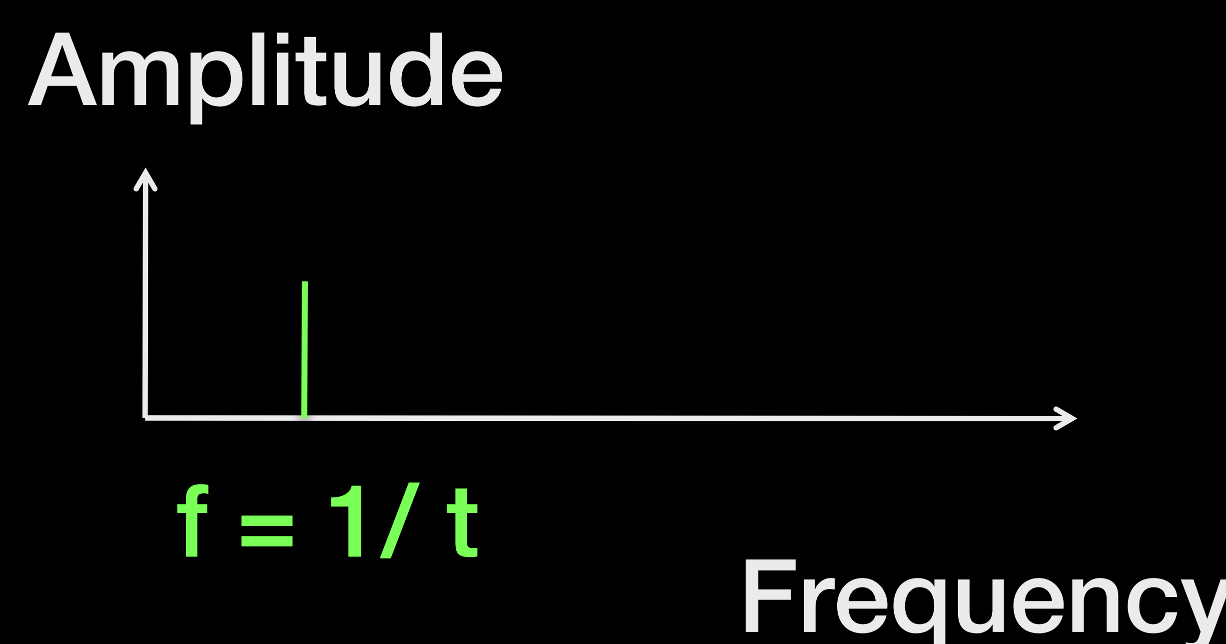


$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\,dt$$

$$\mathcal{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft}\,df$$

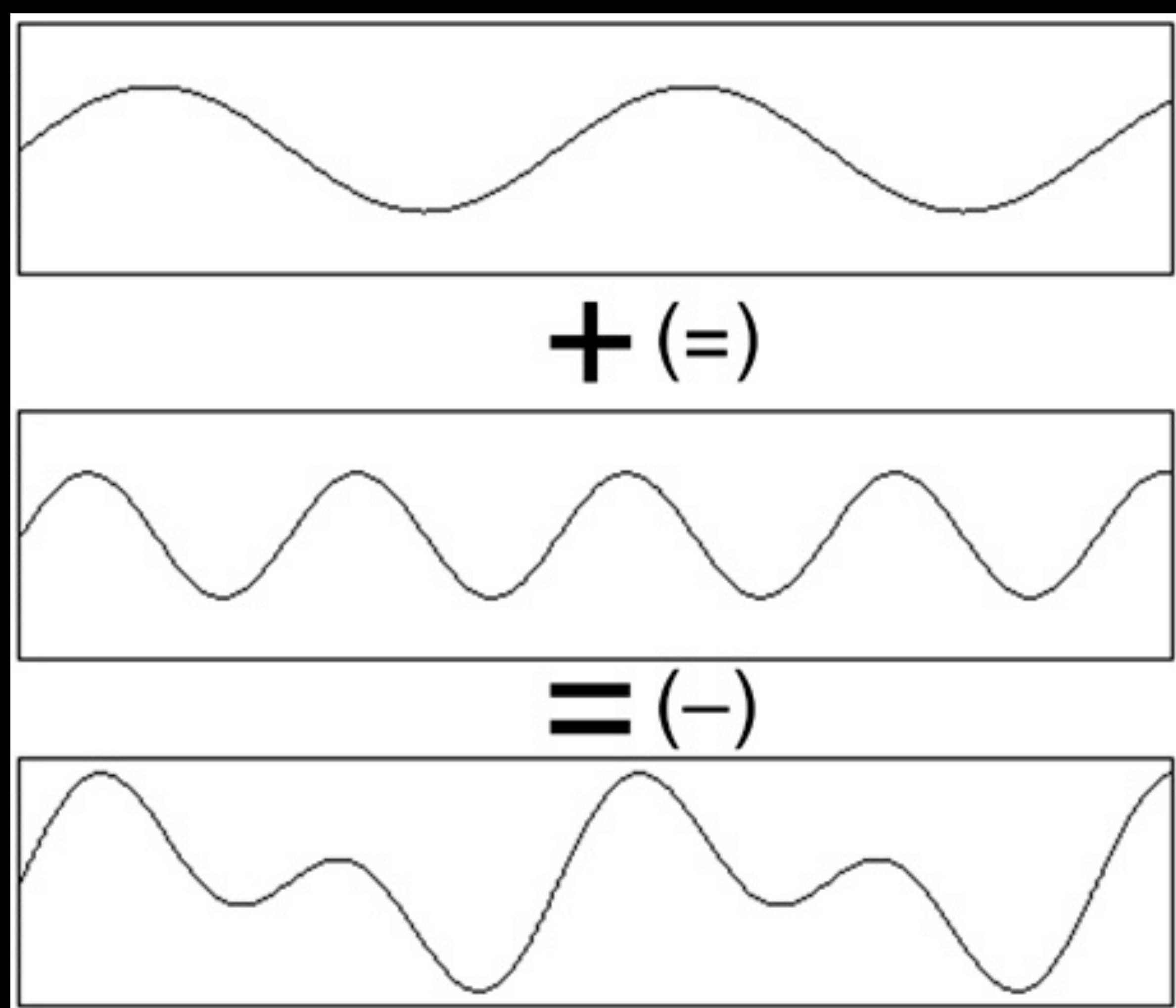
Time and space

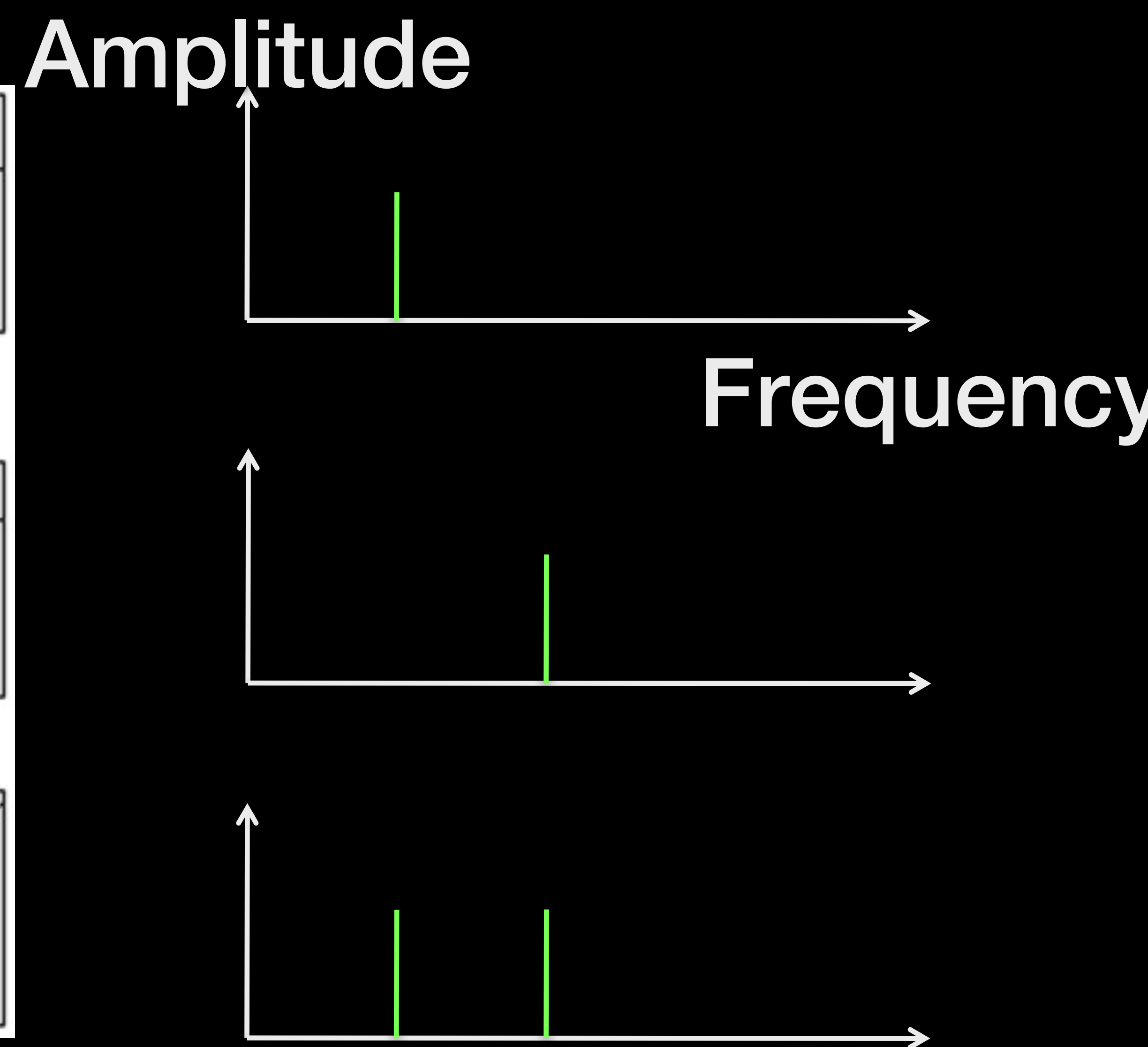




We can Fourier Transform back and forth

Fourier series

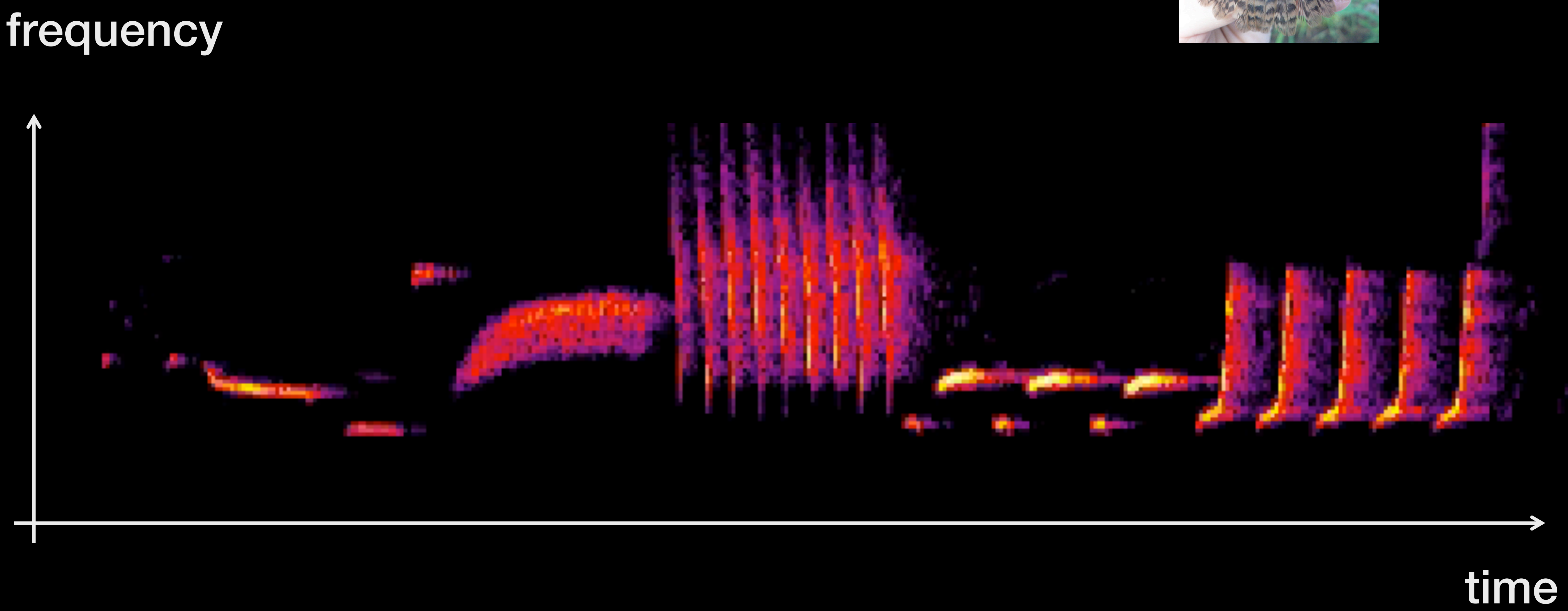




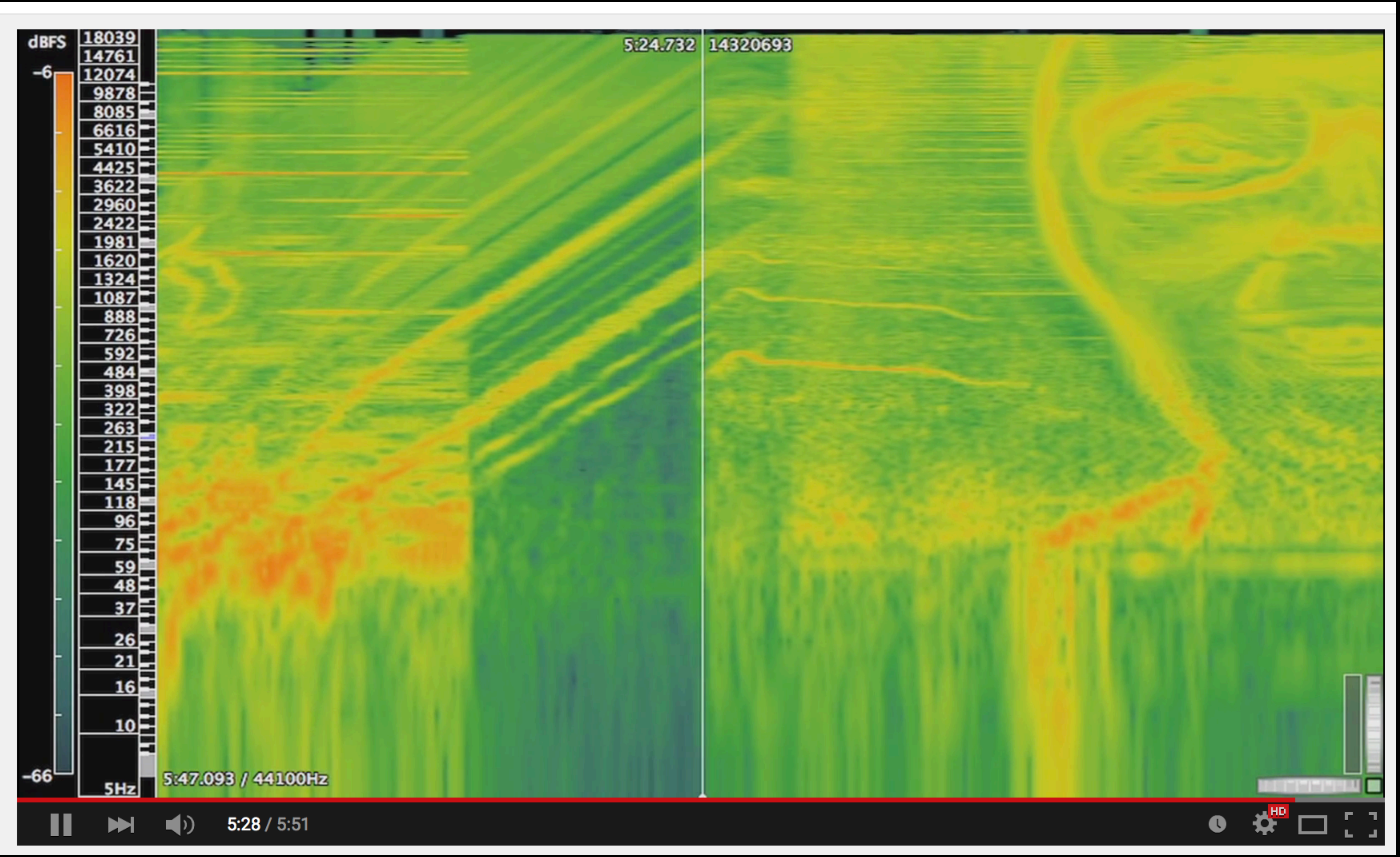
Pressure *vs* Time

Amplitude *vs* Frequency

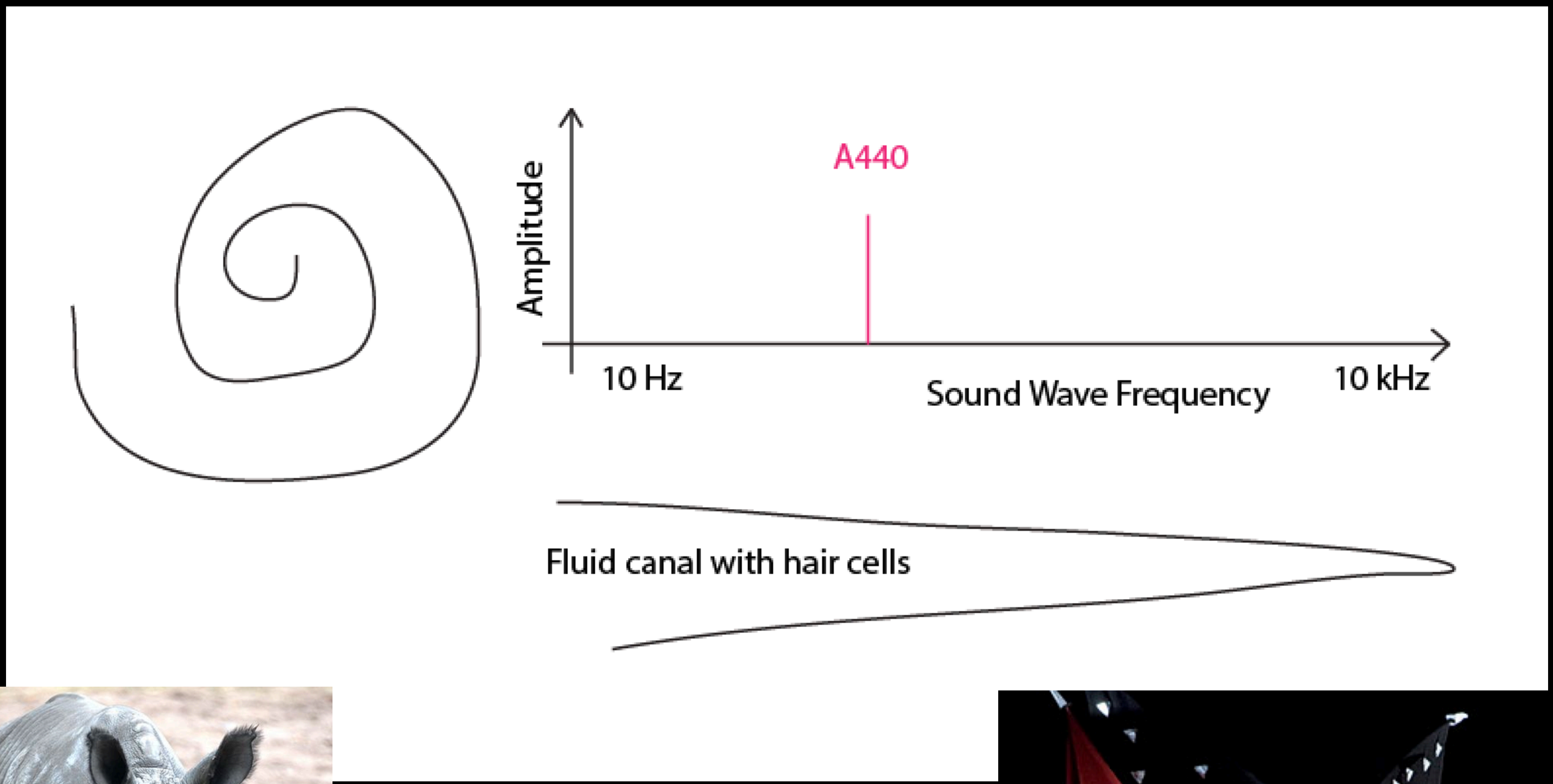
Short-time Fourier spectrogram



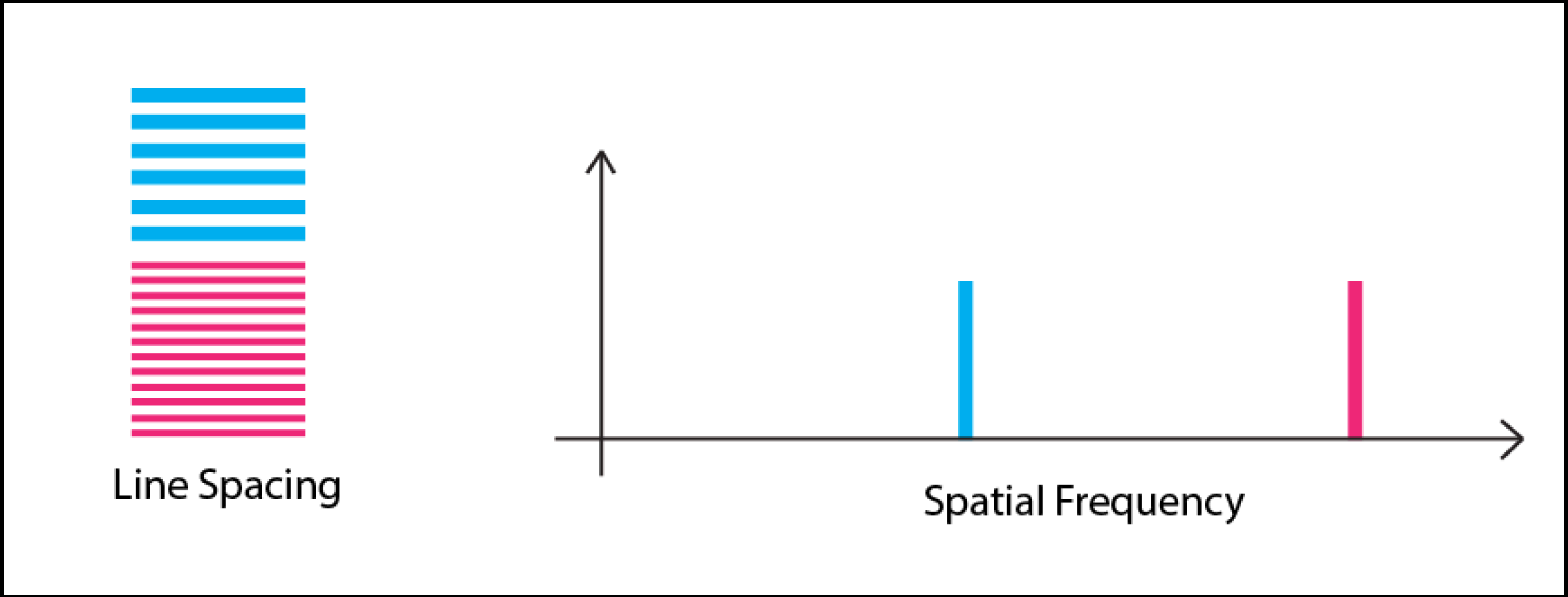
Aphex Twin "Mathematical Equation"



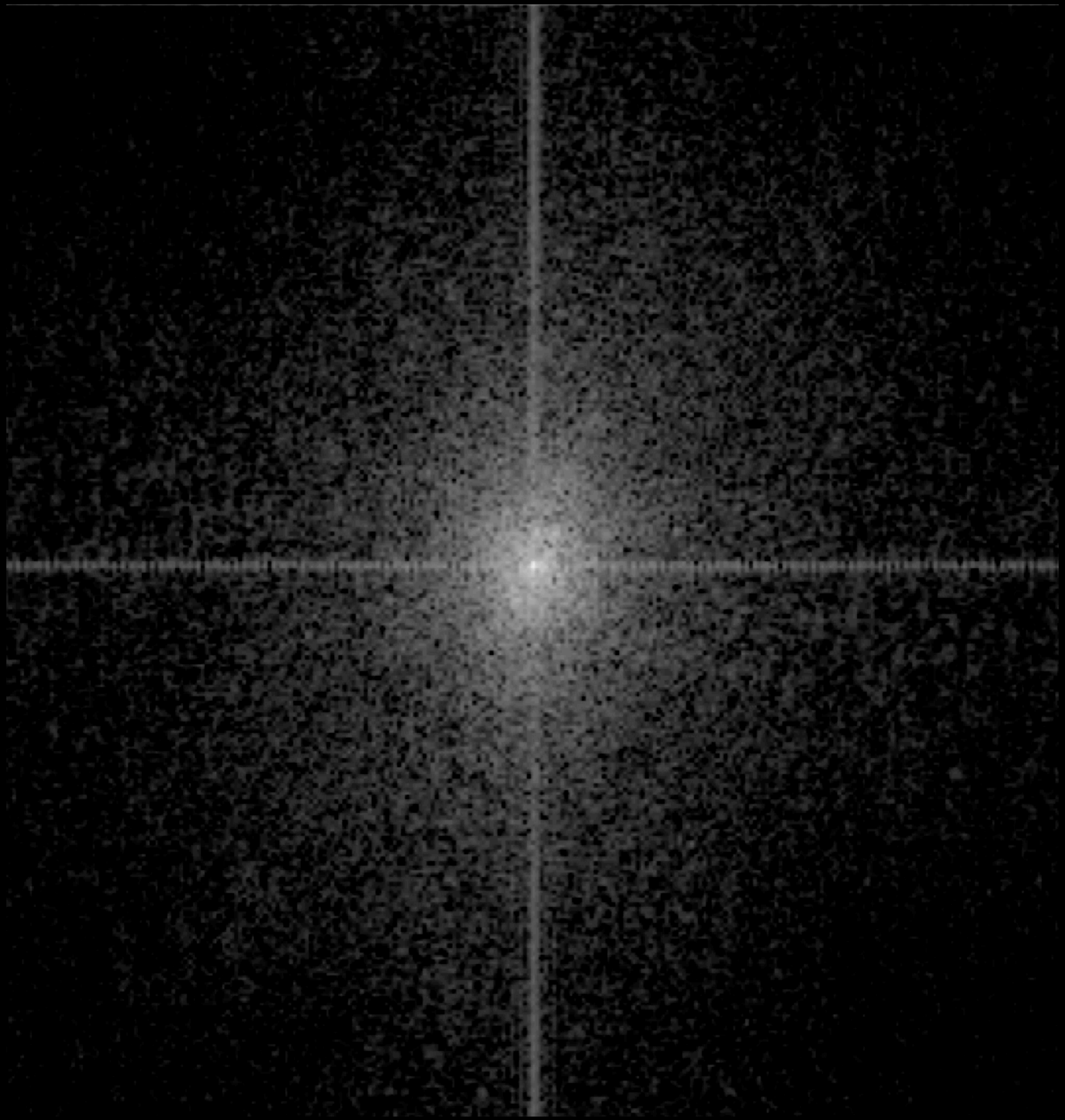
Lets go back to the simplest way to plot frequency of sound:



1D spatial frequency

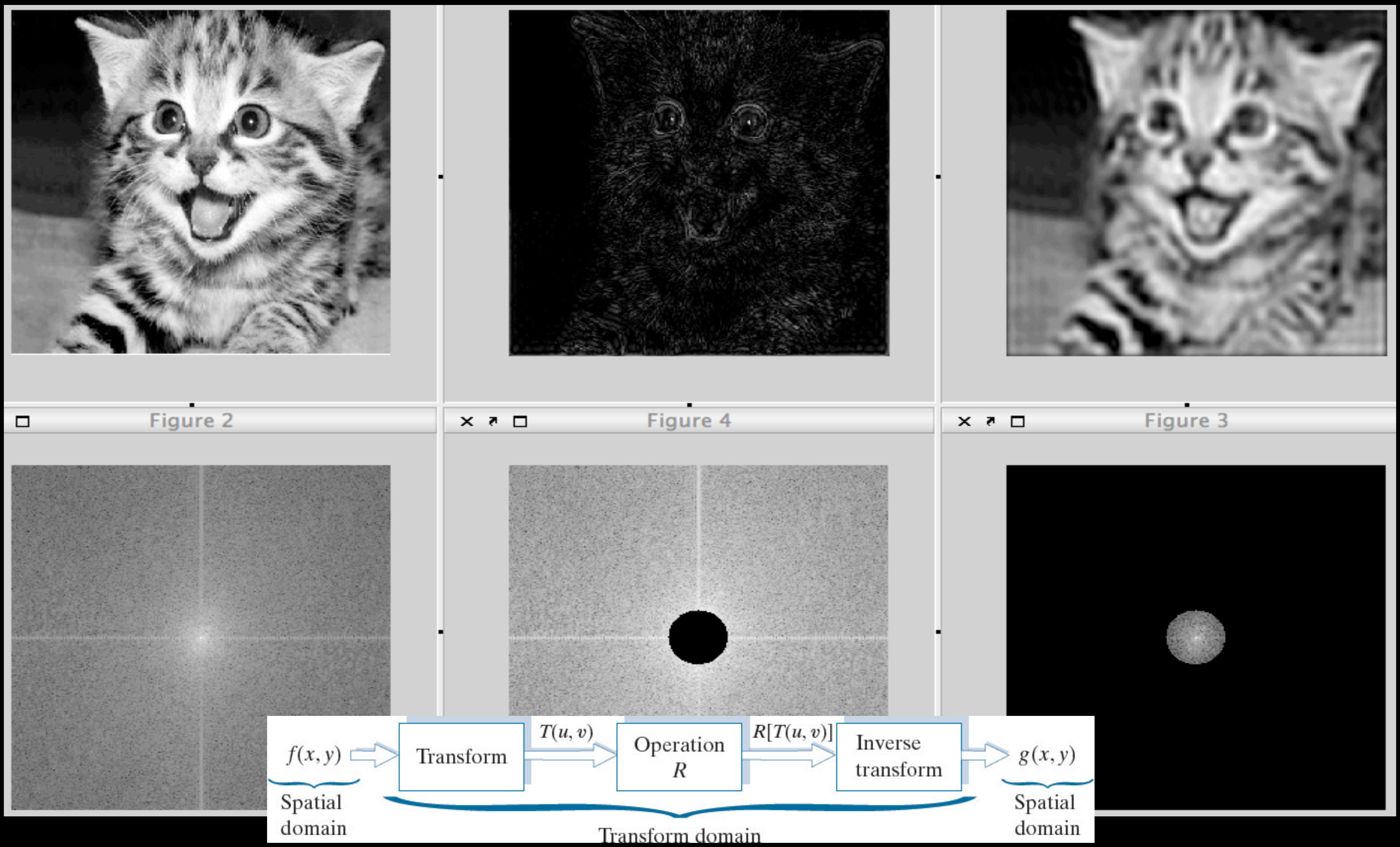


2D spatial frequency



$$y(k_1, k_2) = \iint f(x_1, x_2) \mathrm{e}^{-i2\pi(k_1 x_1 + k_2 x_2)}\, dx_1 dx_2$$

Spatial filtering



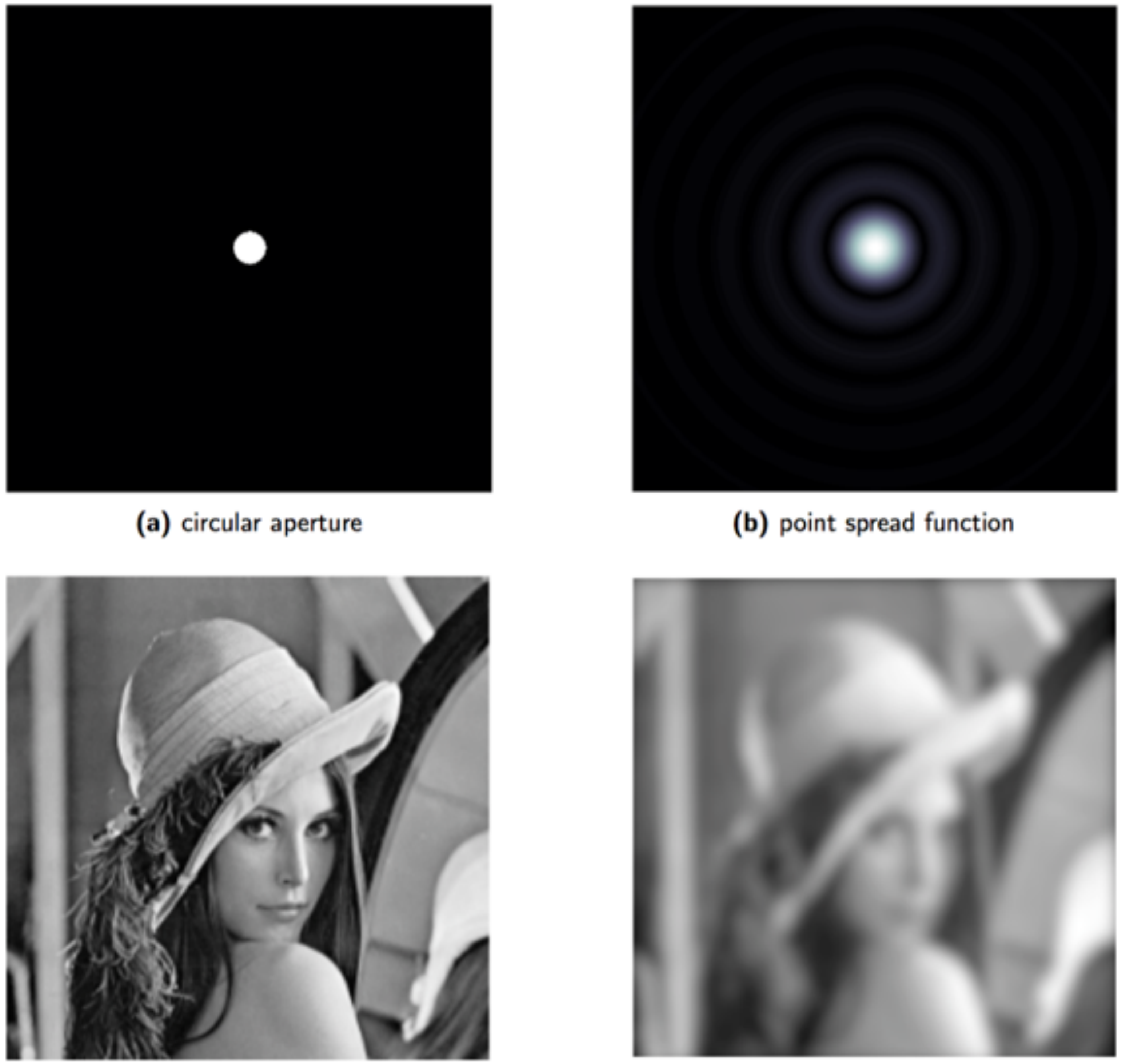
Tip:

- To get rid of ringing and artifacts when you do spatial filtering in Fourier space: don't apply a discrete binary mask, but apply a butterworth or similar filter to smooth out the sharp edge of the mask

Point Spread Function (PSF)

- Imaging systems like cameras and microscope cannot image with infinitely small resolution
- A point (infinitely small spot) in real space thus becomes a blurred spot in image space
- The size and shape of this blur spot is described by the PSF
- The image is the PSF convolved by each point source in the object

The convolution clearly has a blurring effect, explained by the averaging process. But we can also understand this as the result of a low-pass filter. Indeed, in the spatial domain, $\hat{P}$ corresponds to P, which is the indicator function of a circular lens aperture. Considered as a transfer function, P eliminates all frequency content for $\|\underline{\omega}\| > a$. Consequently, we observe a true low-pass version of the original image.



(a) circular aperture

(b) point spread function

(c) original image

(d) convolved with PSF

Figure 3: The Lena image convolved with the point spread function of a circular aperture. The result is a low-pass filtered version of the original image, corresponding to the local averaging performed by the convolution. The Airy disk, whose width is determined by the first zero of the Bessel function J_1, determines the region of greatest contribution to each local average.

Convolution

- Convolution in real space is a multiplication in Fourier space - simpler operation
- Convolution is kind of a “drag-and-stamp” operation

3 Convolutions

The **convolution** product in $\mathbb{R}$ is defined as

$$(f * g)(t) = \int f(u)g(t-u)\,du.$$

The convolution product is

(i) **Commutative:** $f * g = g * f$ for any f, g.

(ii) **Associative:** $(f * g) * h = f * (g * h)$ for any f, g, h.

For fixed h we can define the **convolution operator** $f \overset{L}{\mapsto} g = h * f$. Roughly speaking, the output g is the local averages of the input f, weighted by the function h. For instance, when h is a Gaussian curve, g is a smoothed version of f.

A convolution operator is a TIO. Indeed,

$$\begin{aligned}
(Lf_\tau)(t) &= \int_{\mathbb{R}} f(u-\tau)h(t-u)\,du \\
&= \int_{\mathbb{R}} f(u)h((t-\tau)-u)\,du \\
&= (f * h)(t-\tau) \\
&= (Lf)_\tau(t).
\end{aligned}$$

Conversely, every TIO is a convolution operator. To see this, we write

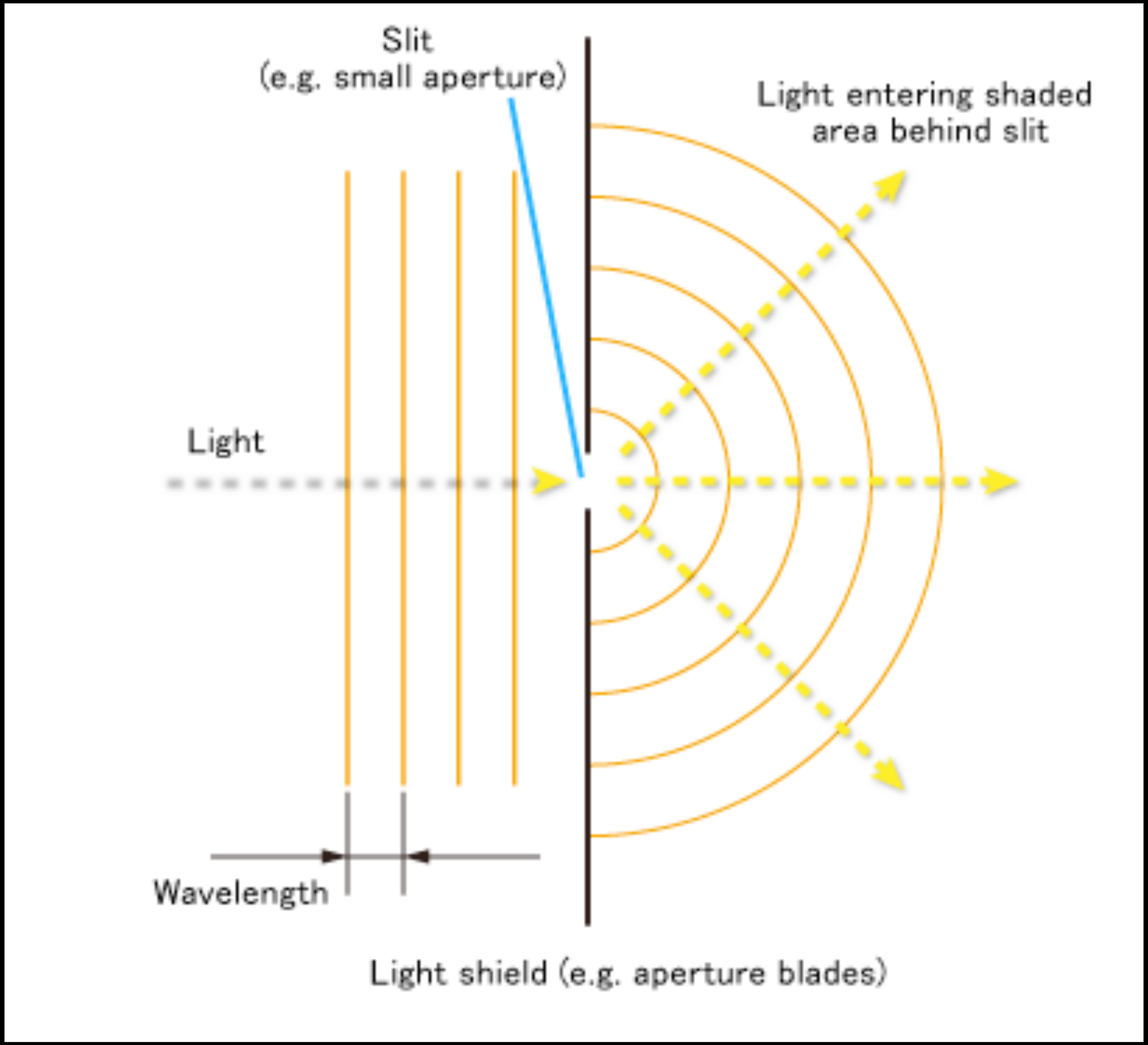
$$g(t) = \int h(t,u)f(u)\,du.$$

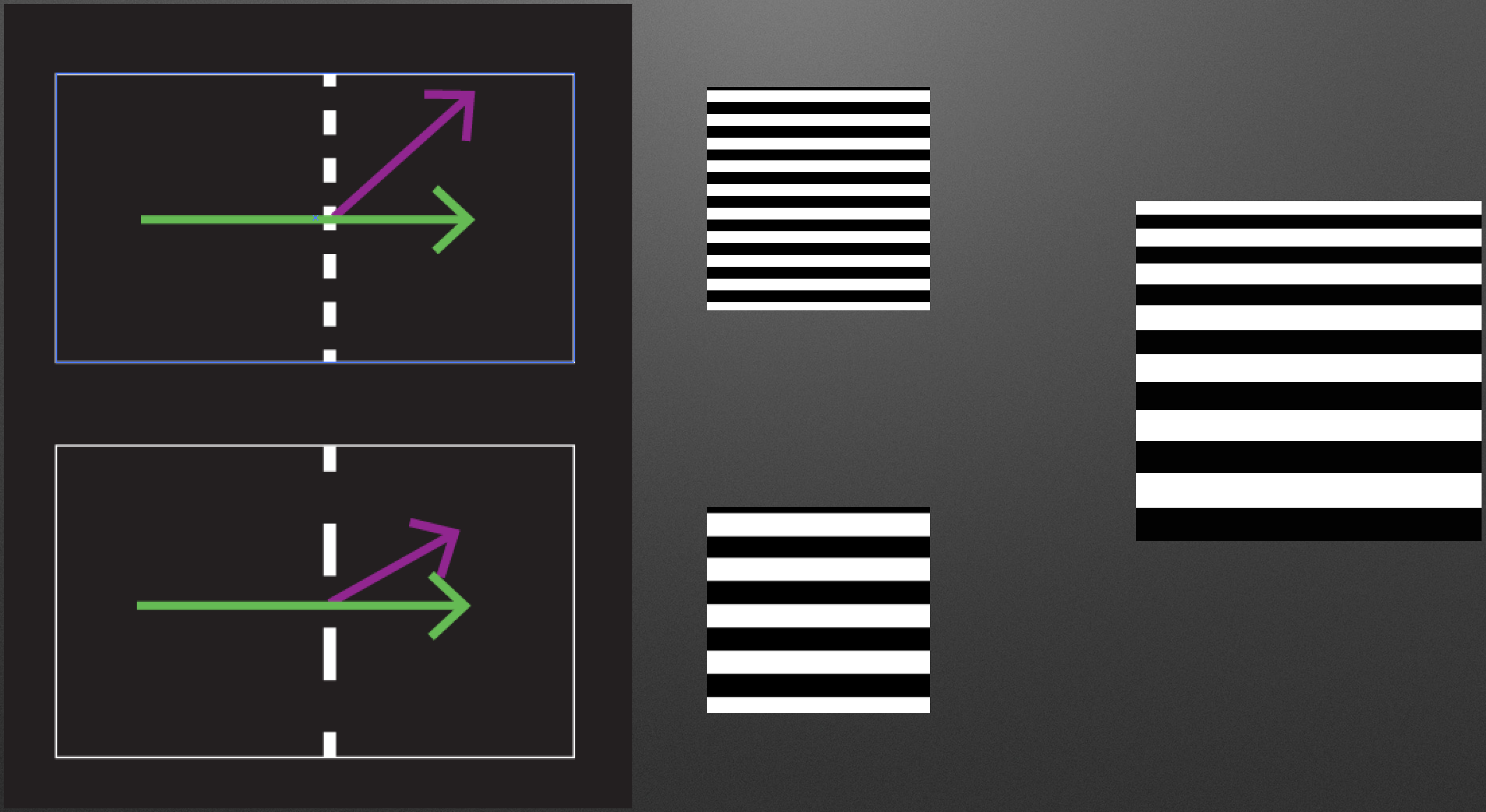
This is the continuous analog of a matrix-vector product. To prove the claim, consider on one hand

$$g_\tau(t) = (Lf)_\tau(t) = \int h(t-\tau, u)f(u)\,du. \tag{1}$$

- Convolution in real space corresponds to multiplication in Fourier space!

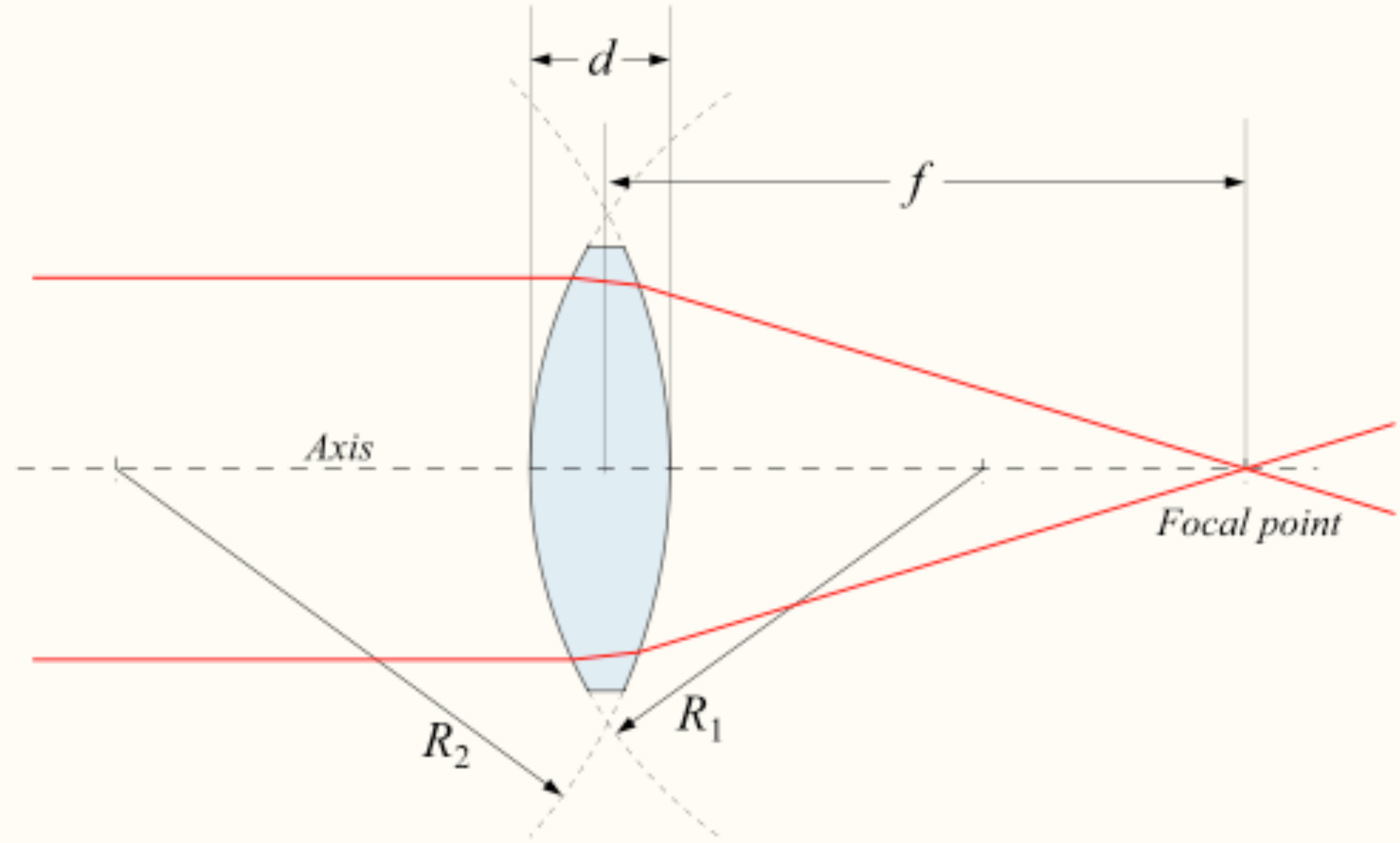
Image formation: Diffraction





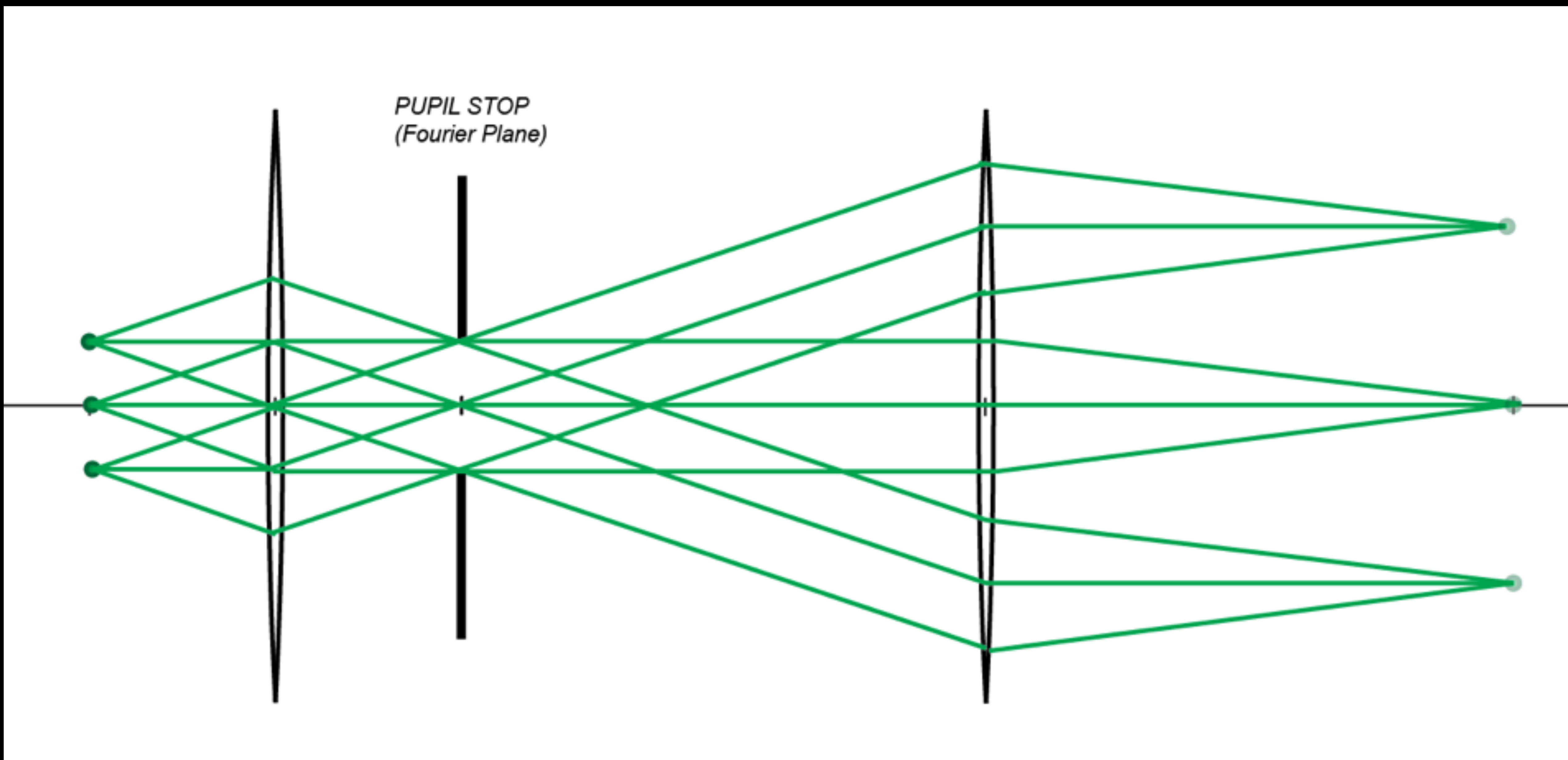
- Fine spatial frequencies diffract light stronger: this is why the finer spatial frequencies end up at the edges in Fourier space

Single Lens

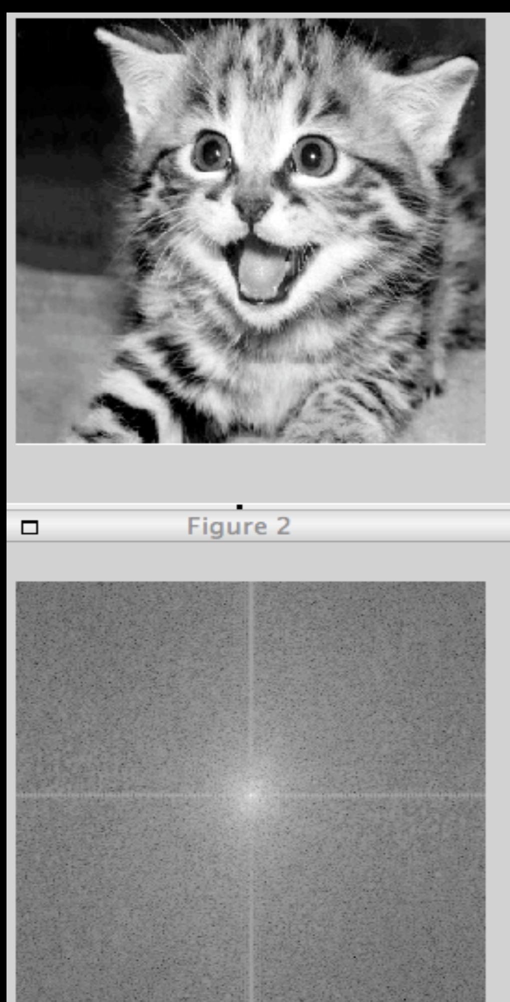


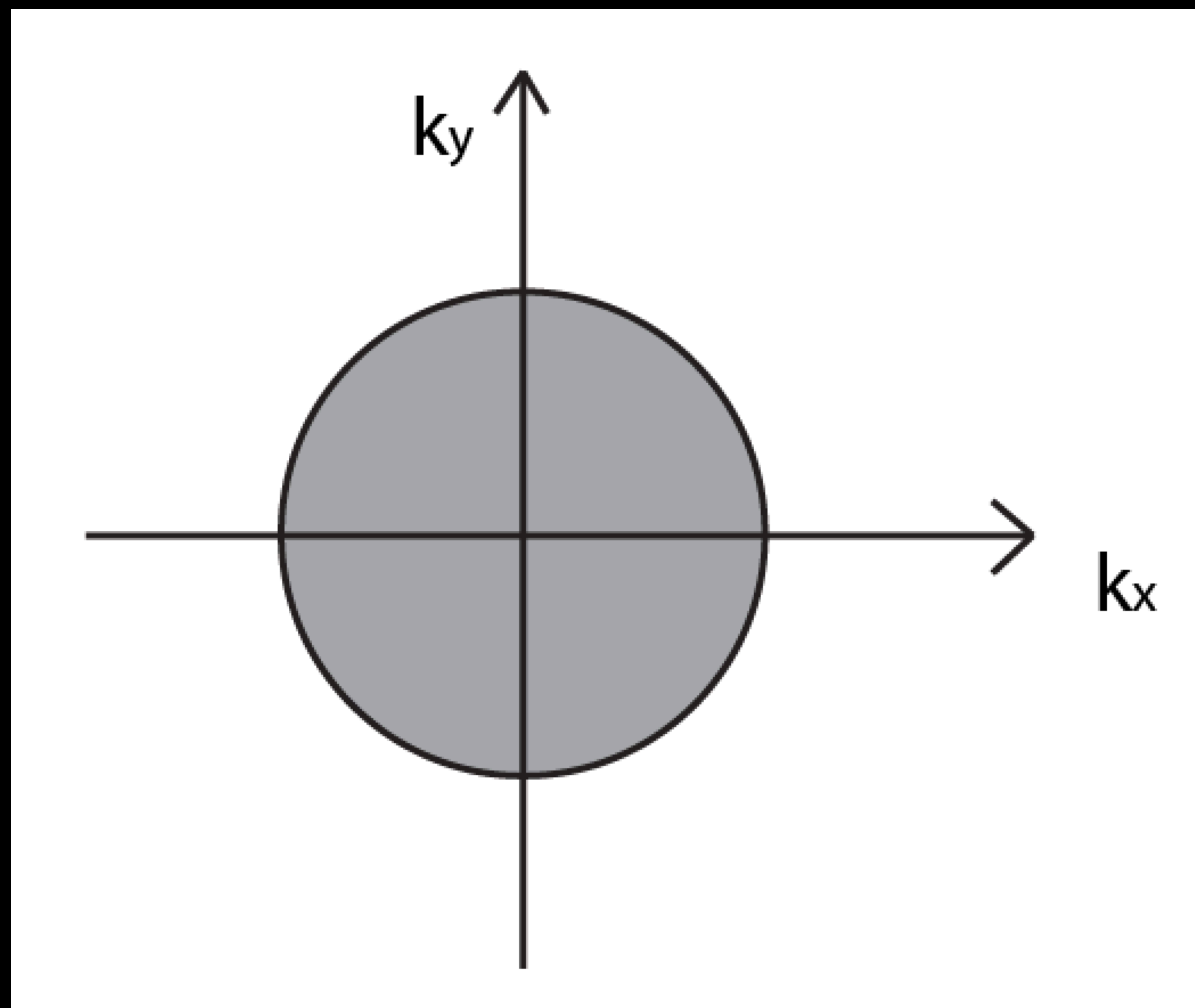
Positive (converging) lens

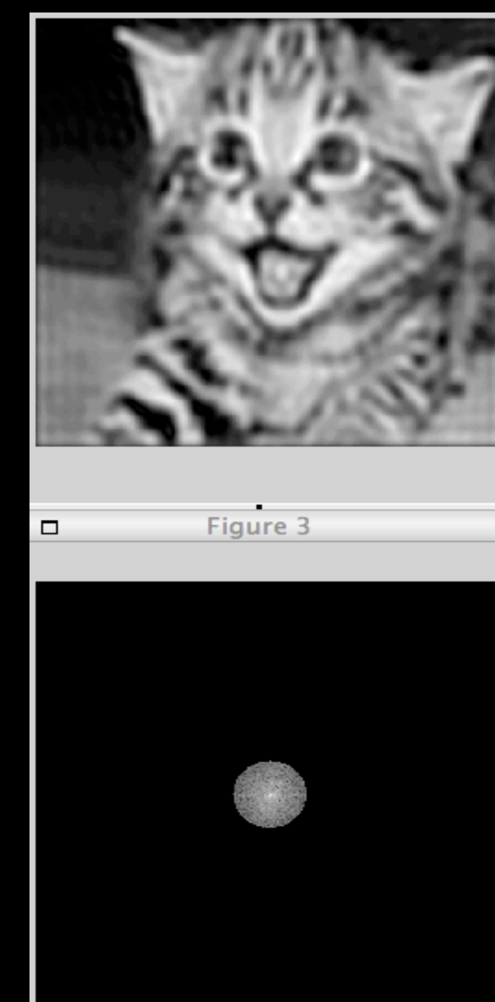
Two-lens imaging system



Optics: frequency space OTF support corresponds to the physical aperture size



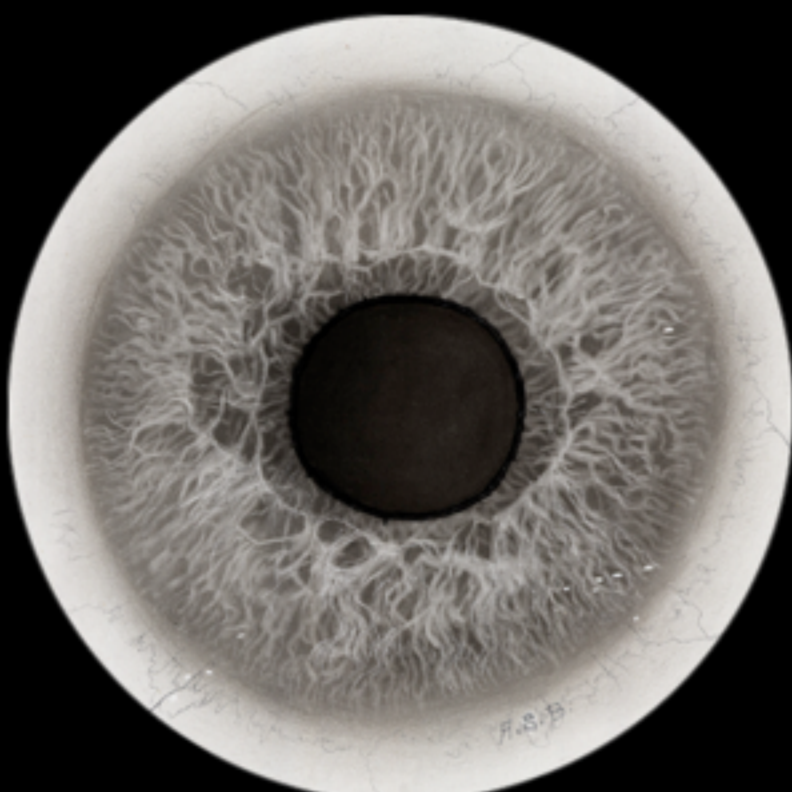




OTF = Optical Transfer Function

(k_x, k_y) are the spatial coordinates in frequency (Fourier) space

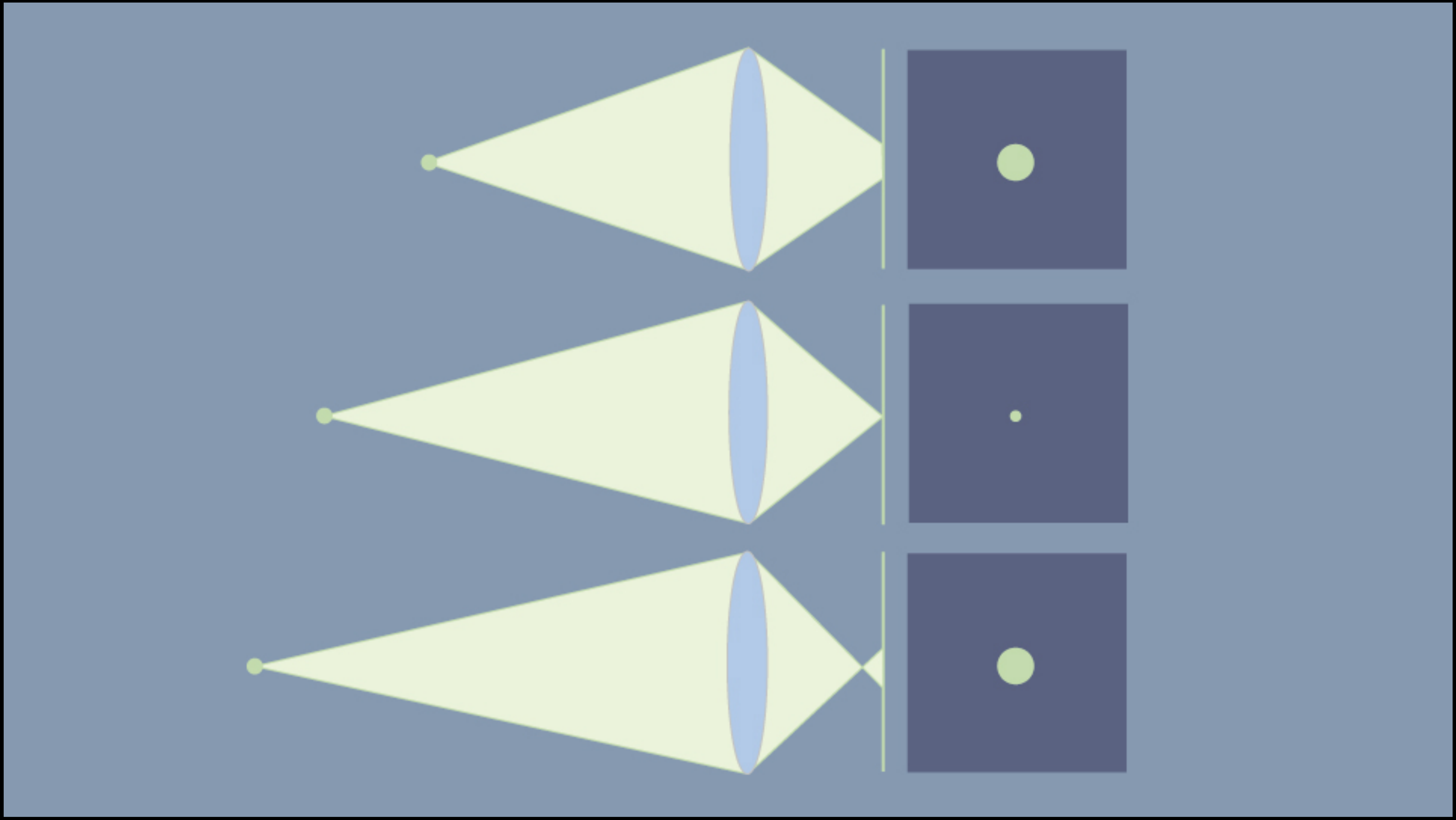
Aperture also determines the depth of field



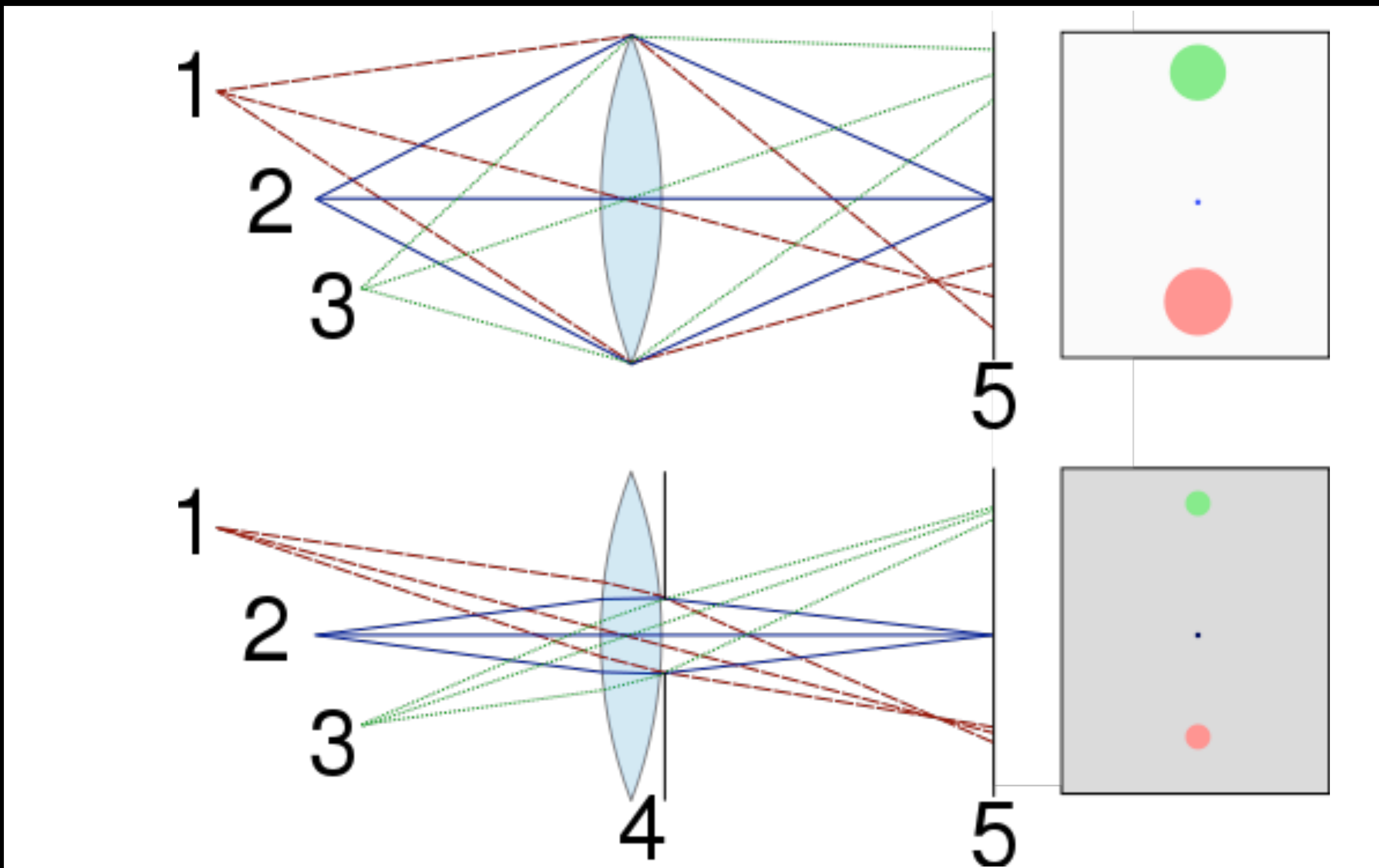
f/1.8 f/2.8 f/4.0 f/5.6

f/8 f/11 f/16 f/22

Circle of Confusion



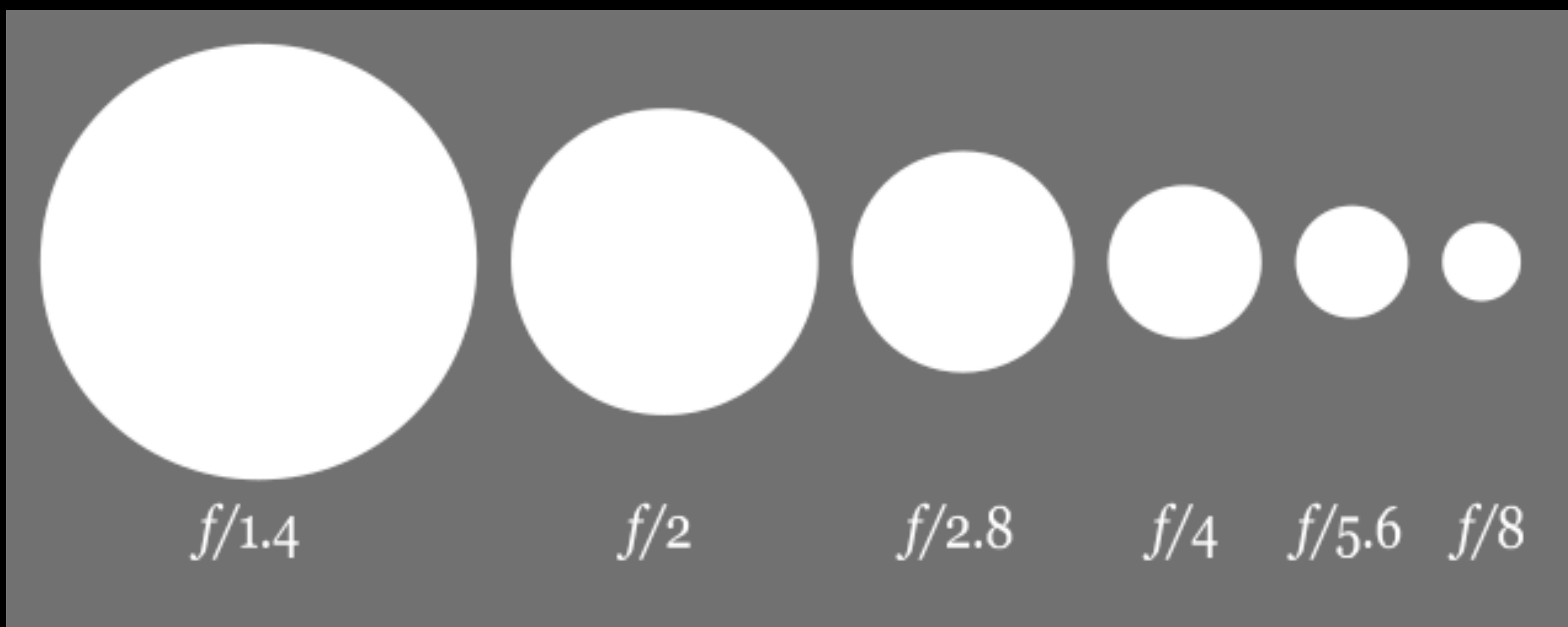
Large aperture = less DOF



F# ("F-number") = f/D

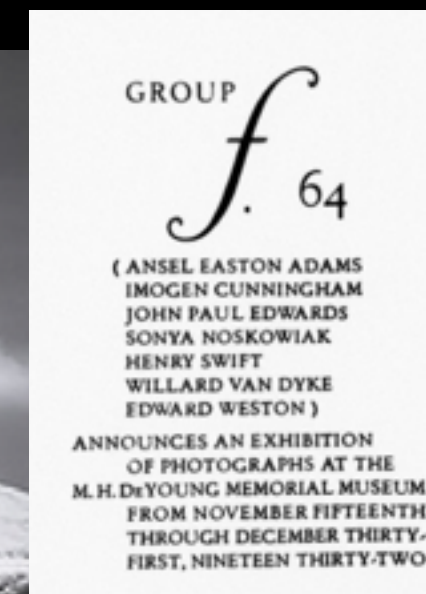
Determined by ratio of lens focal length *f* and lens aperture Diameter *D*

YOU DON'T NEED TO MEMORIZE THIS DEFINITION,
ITS JUST FOR REFERENCE



$$f/1 = f/(\sqrt{2})^0,\ f/1.4 = f/(\sqrt{2})^1,\ f/2 = f/(\sqrt{2})^2,\ f/2.8 = f/(\sqrt{2})^3\ \ldots$$

Small aperture


GROUP f. 64

(ANSEL EASTON ADAMS
IMOGEN CUNNINGHAM
JOHN PAUL EDWARDS
SONYA NOSKOWIAK
HENRY SWIFT
WILLARD VAN DYKE
EDWARD WESTON)

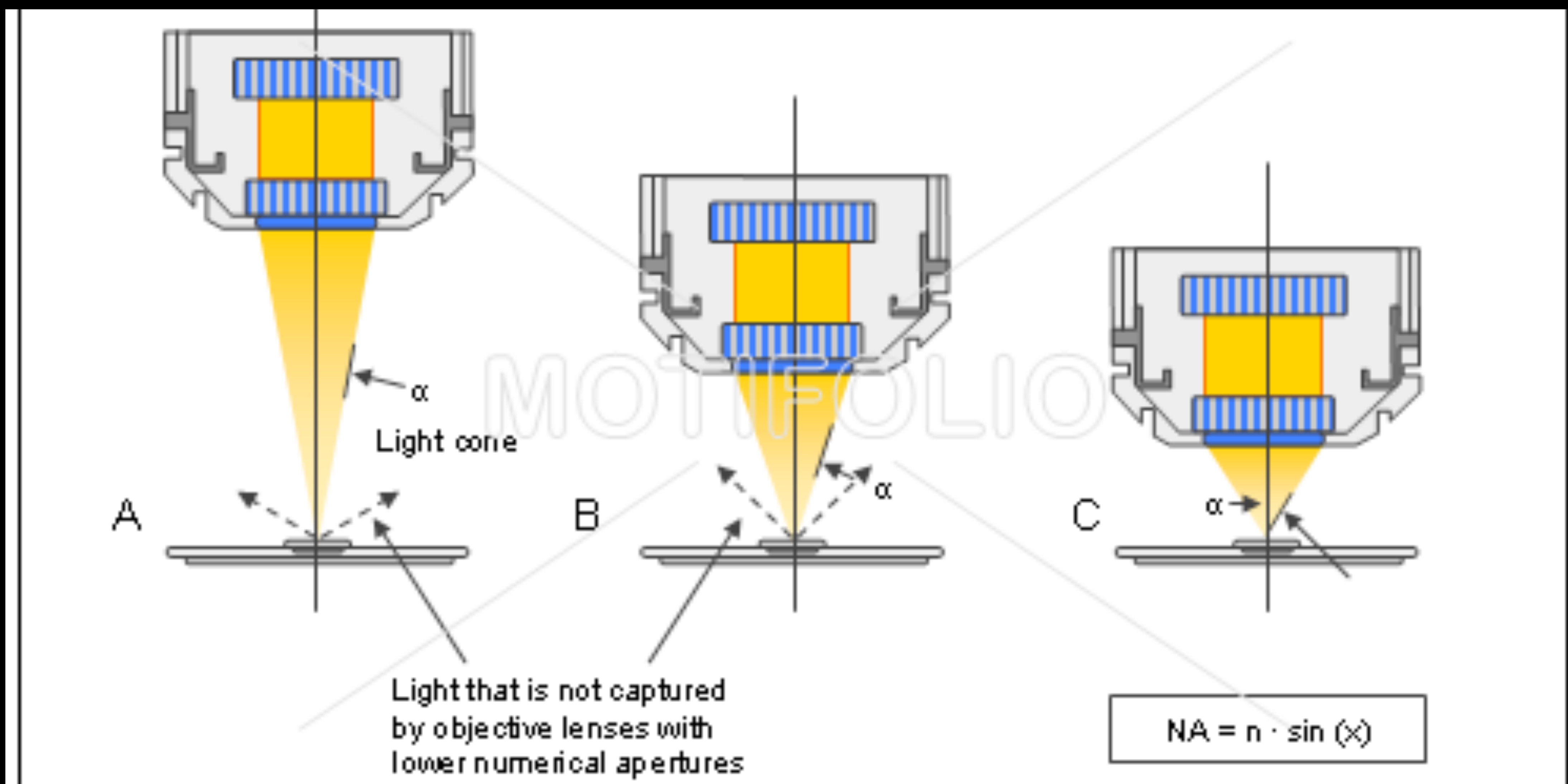
ANNOUNCES AN EXHIBITION
OF PHOTOGRAPHS AT THE
M. H. DeYOUNG MEMORIAL MUSEUM
FROM NOVEMBER FIFTEENTH
THROUGH DECEMBER THIRTY-
FIRST, NINETEEN THIRTY-TWO

…closed aperture - very large depth of field

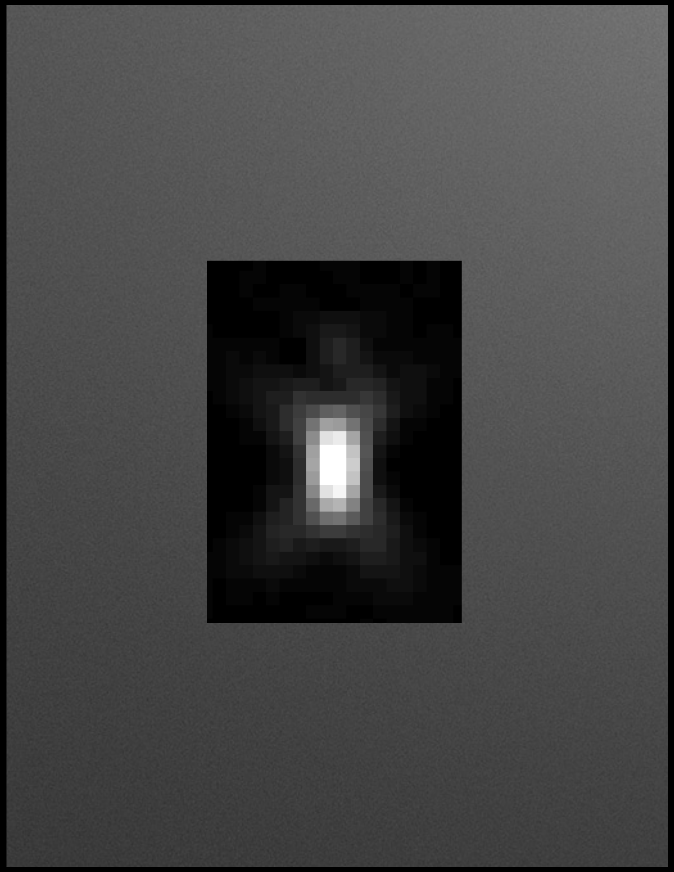
Large aperture

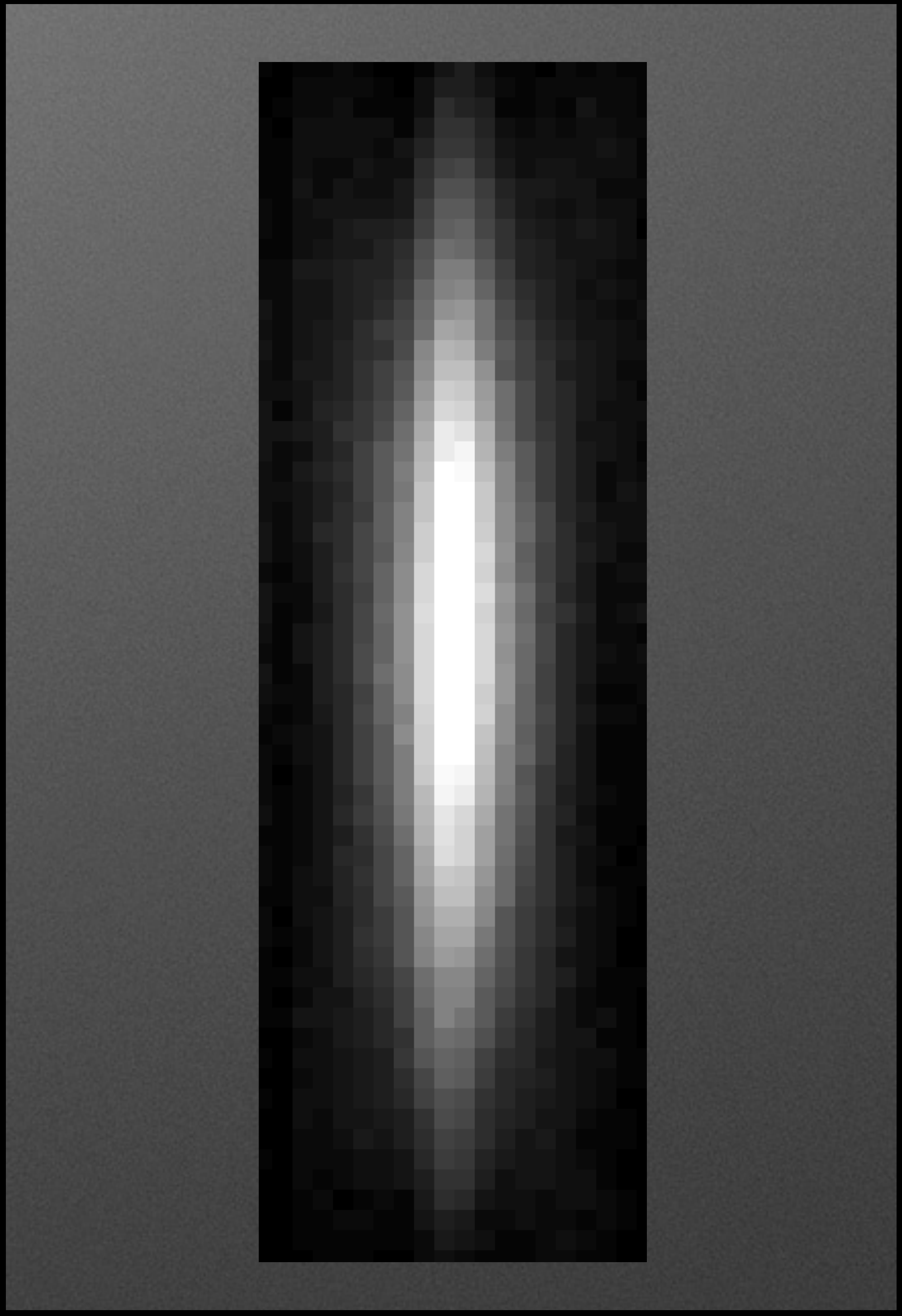
Microscope Objective

Numerical Aperture (NA) in a microscope can be super large -> really high resolution



Axial PSF measurements in a microscope





Operations

- Element wise (pixel by pixel) *vs.* Matrix operations
- Single-pixel vs neighborhood:
  - Single-pixel: grab *e.g.* the value of the one nearest pixel)
  - Neighborhood (calculate and use e.g. the average / max / min value of the nearest neighbors)

Resize with single nearest neighbor interpolation

- Take an image of size 500x500 pixels.
- Resize this to 750x750 pixels:
    - Shrink the 750x750 grid to overlay the original and **select the value** of each pixel in the new image to that of the **nearest pixel** of the original image
- Works, but introduces artifacts such as distortion of straight edges

Neighbors of a pixel

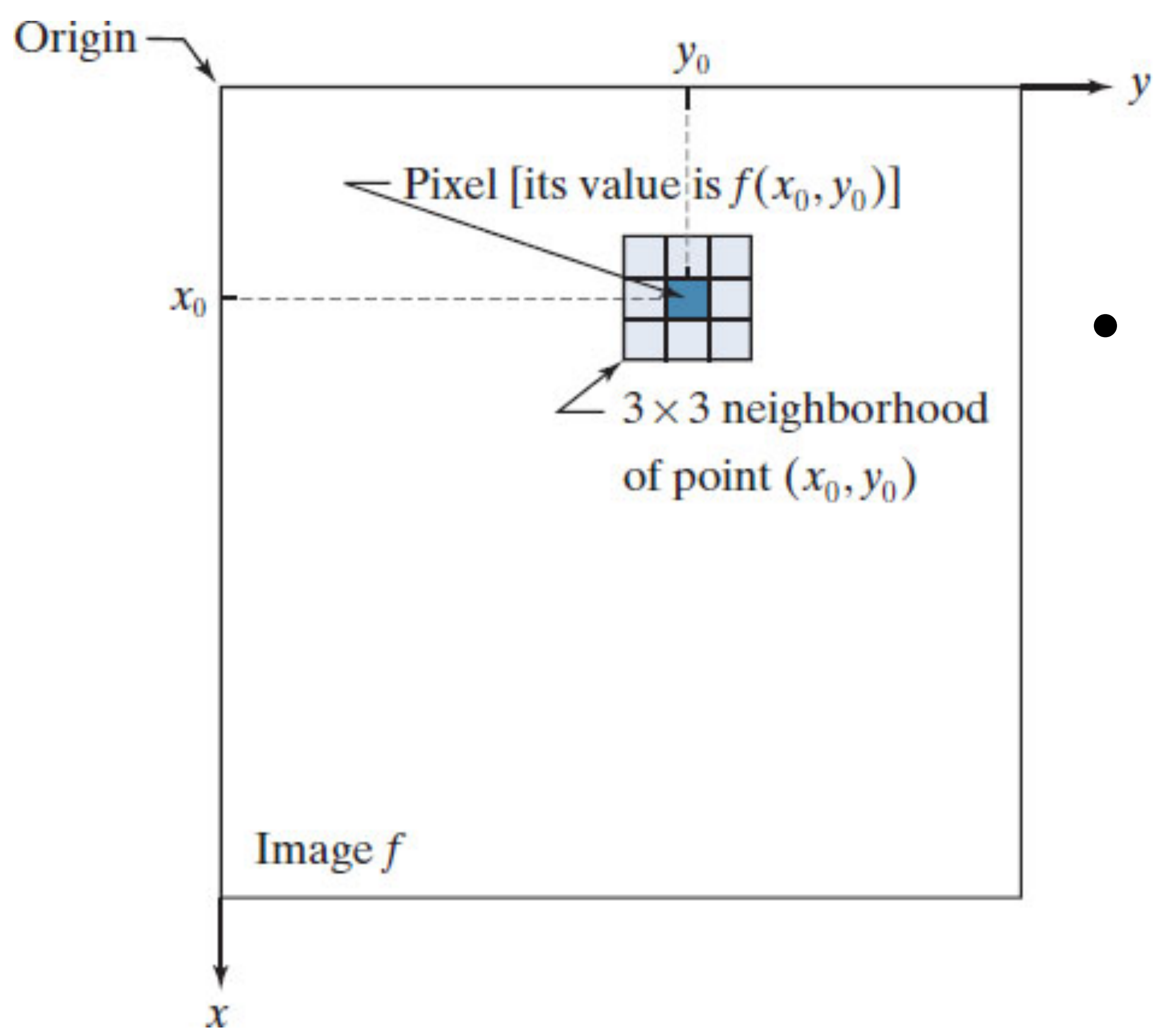
- Pixel p at coordinates (x,y) has two **horizontal** and two **vertical** neighbors with coordinates:

  4-neigborhood: (x+1,y), (x-1,y), (x,y+1), (x,y-1)

- Pixel p at coordinates (x,y) also has four **diagonal** neighbors with coordinates:

  (x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)

Open or Closed



- Neighborhood of pixel p is said to be **open** if it doesn't contain p and **closed** if it contains p

Bilinear interpolation

- Use the four nearest neighbors to estimate the intensity at a given location.

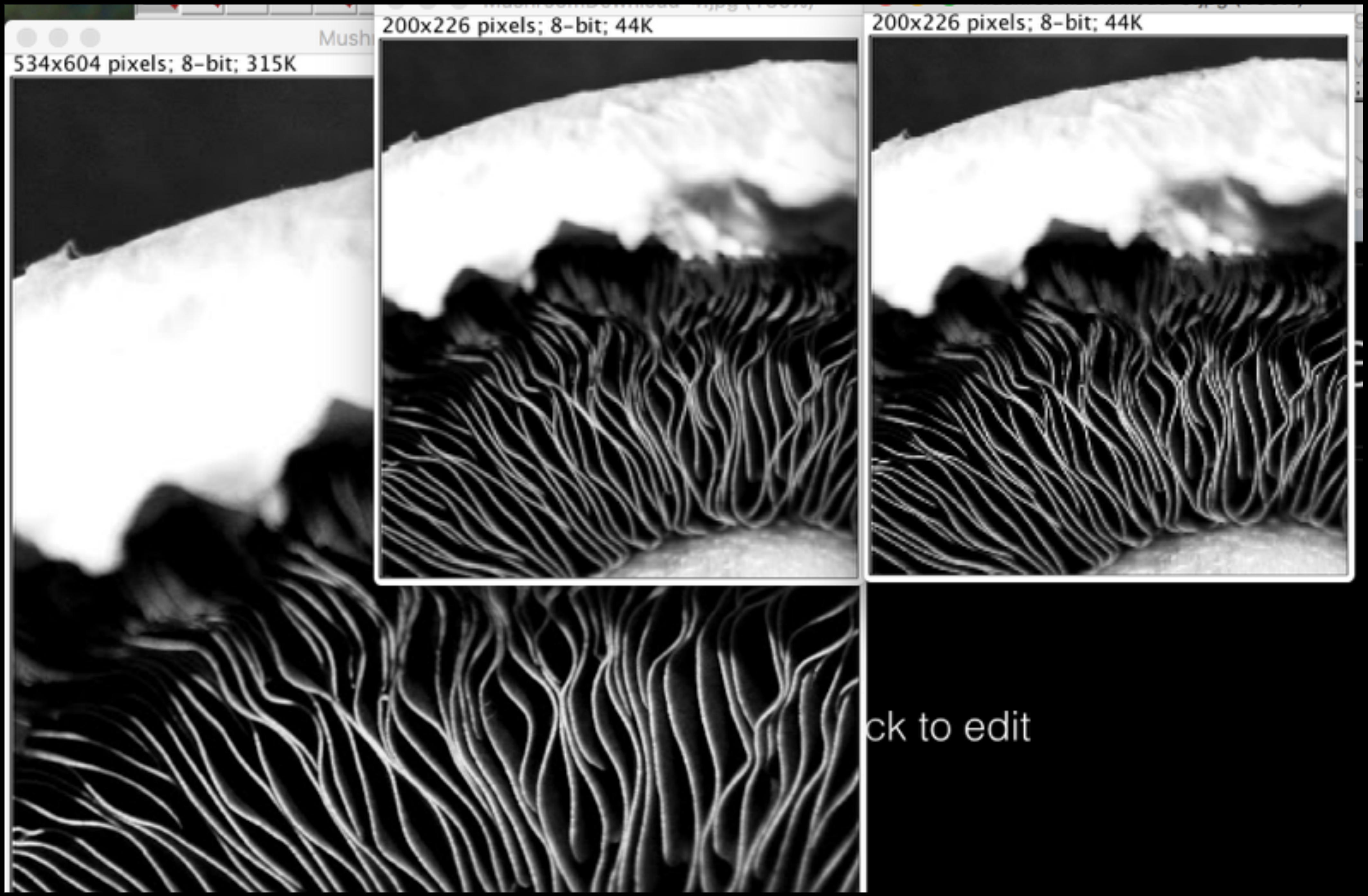
- *Note: “linear” in bilinear refers to lines, not linearity, it is not a linear operation since it multiplies coordinates*

Bicubic interpolation

- Sixteen nearest neighbors:

- *Subpixel accuracy fills the gaps*

Average when downsizing



- Edges lose contrast if you average but result is smoother

Median

- Lets do this as an exercise!

Geometrical transformations

- 2D affine transformations: “rubber-sheet”
  - Include scaling, translation, rotation and shearing
  - Preserve points, straight lines and planes

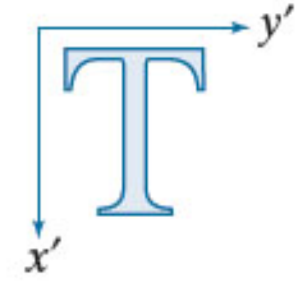
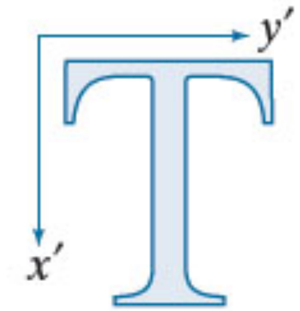
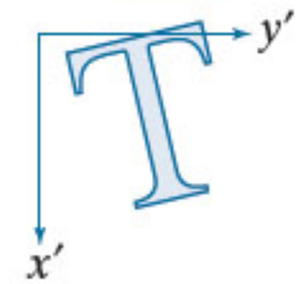
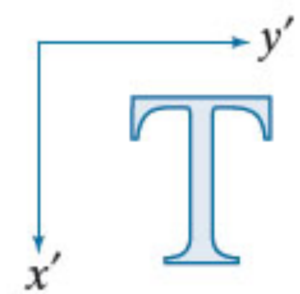
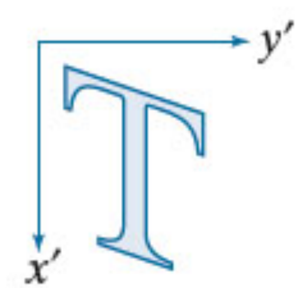
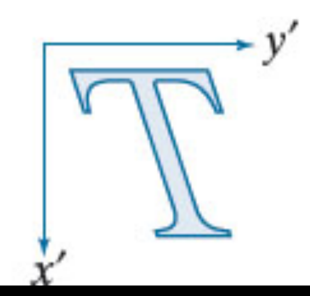
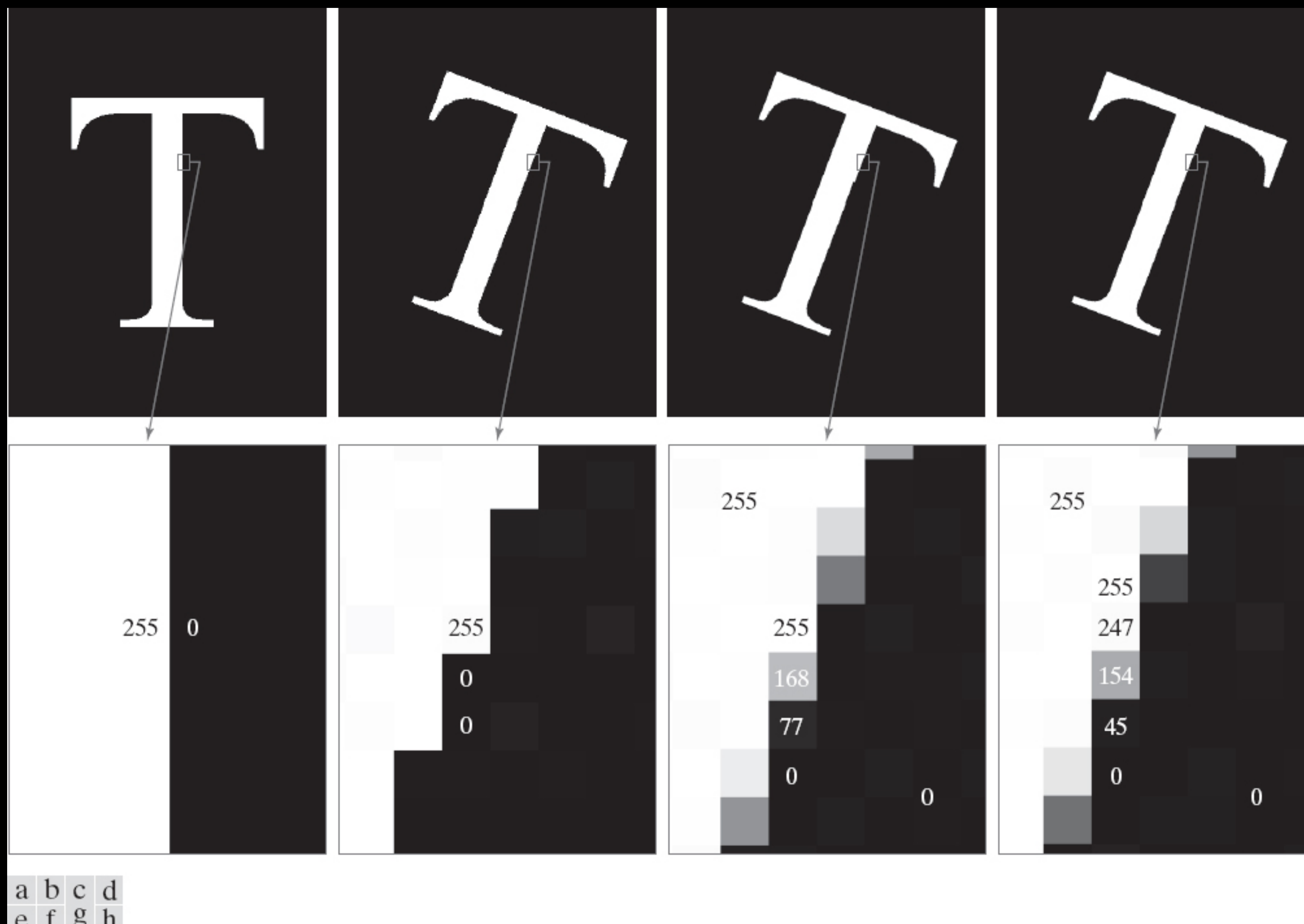
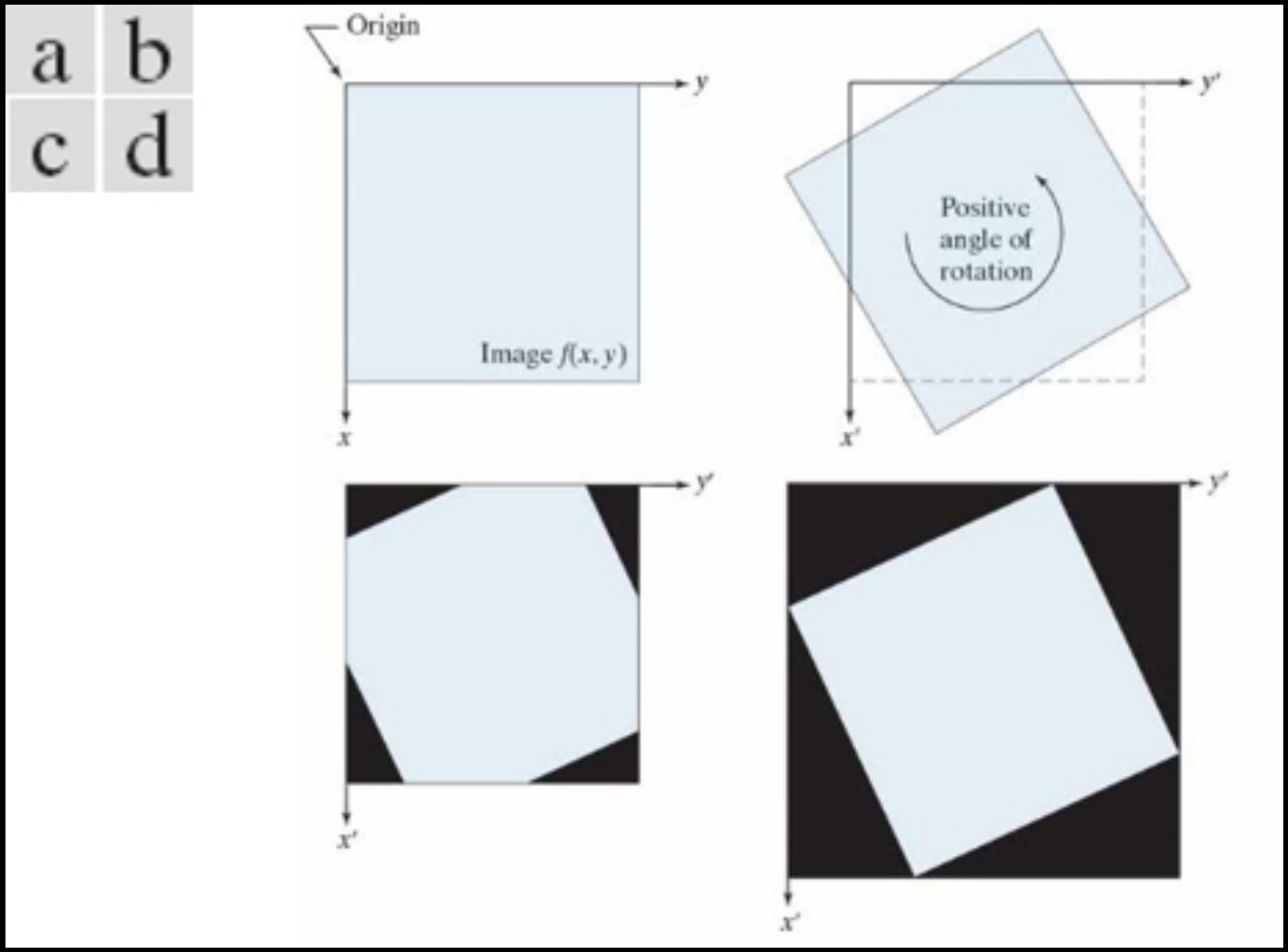
Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to –1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

Image rotation and intensity interpolation




a b
c d
Origin
y
x
Image f(x, y)
y'
x'
Positive angle of rotation
y'
x'
y'
x'

Distortion and registration errors

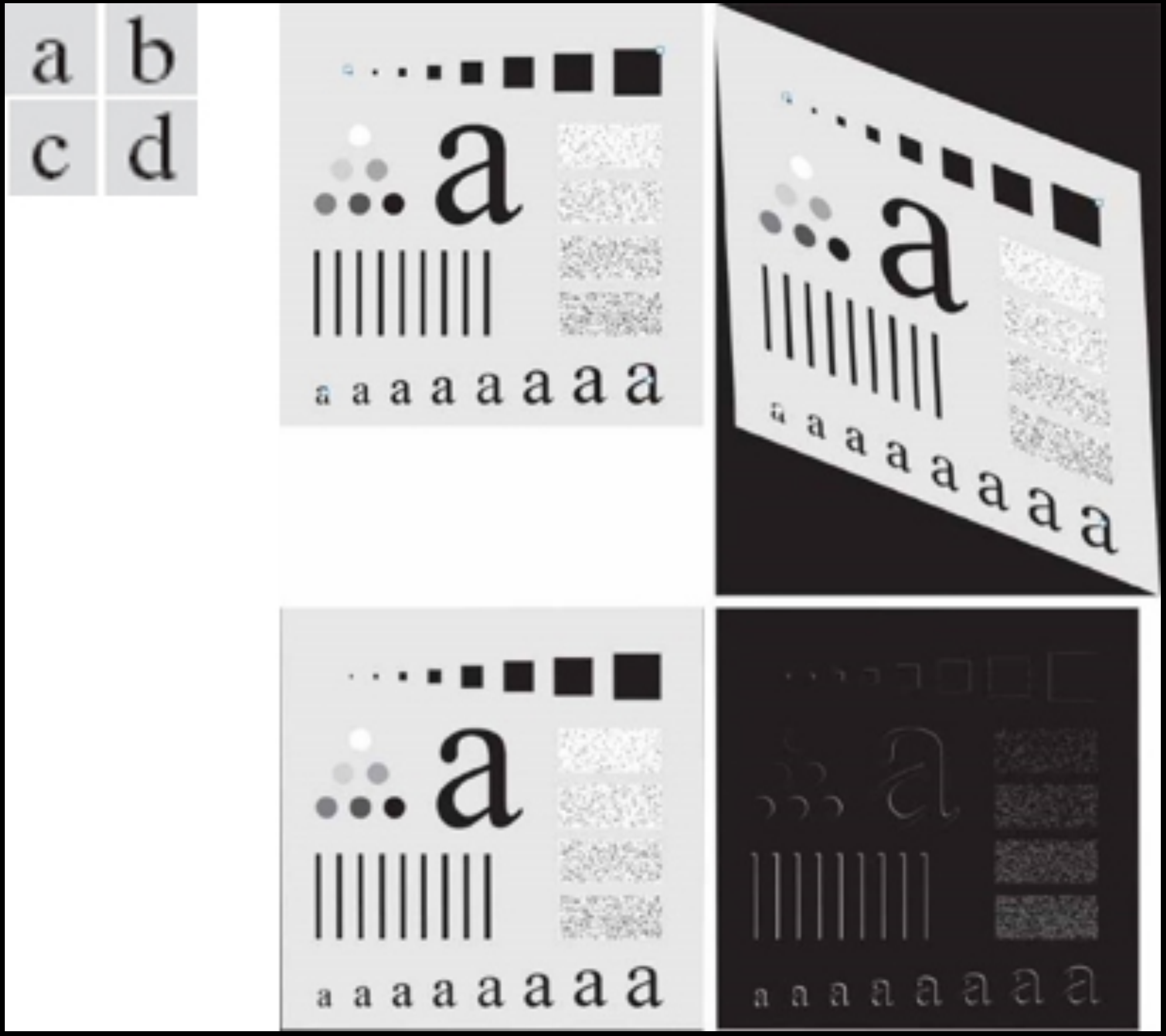


Image Registration

- Put fiducial markers in the object before imaging (or in the original image before transforming)
- This way you have ground truth in a few locations

Linear vs. Nonlinear

- Linear *versus* nonlinear operations
  - Additivity and homogeneity:
  - H[*a* f(x,y) + *b* f(x,y)] =
    H[*a* f(x,y)] + H[*b* f(x,y)] =
    a g(x,y) + b g(x,y)

For example:

- Addition is linear
- Operation to find max pixel in matrix is nonlinear

Interpolation

- Using known data to estimate values at unknown locations, used e.g. when:
  - Zooming / Resizing images
  - Rotating and other affine transformations

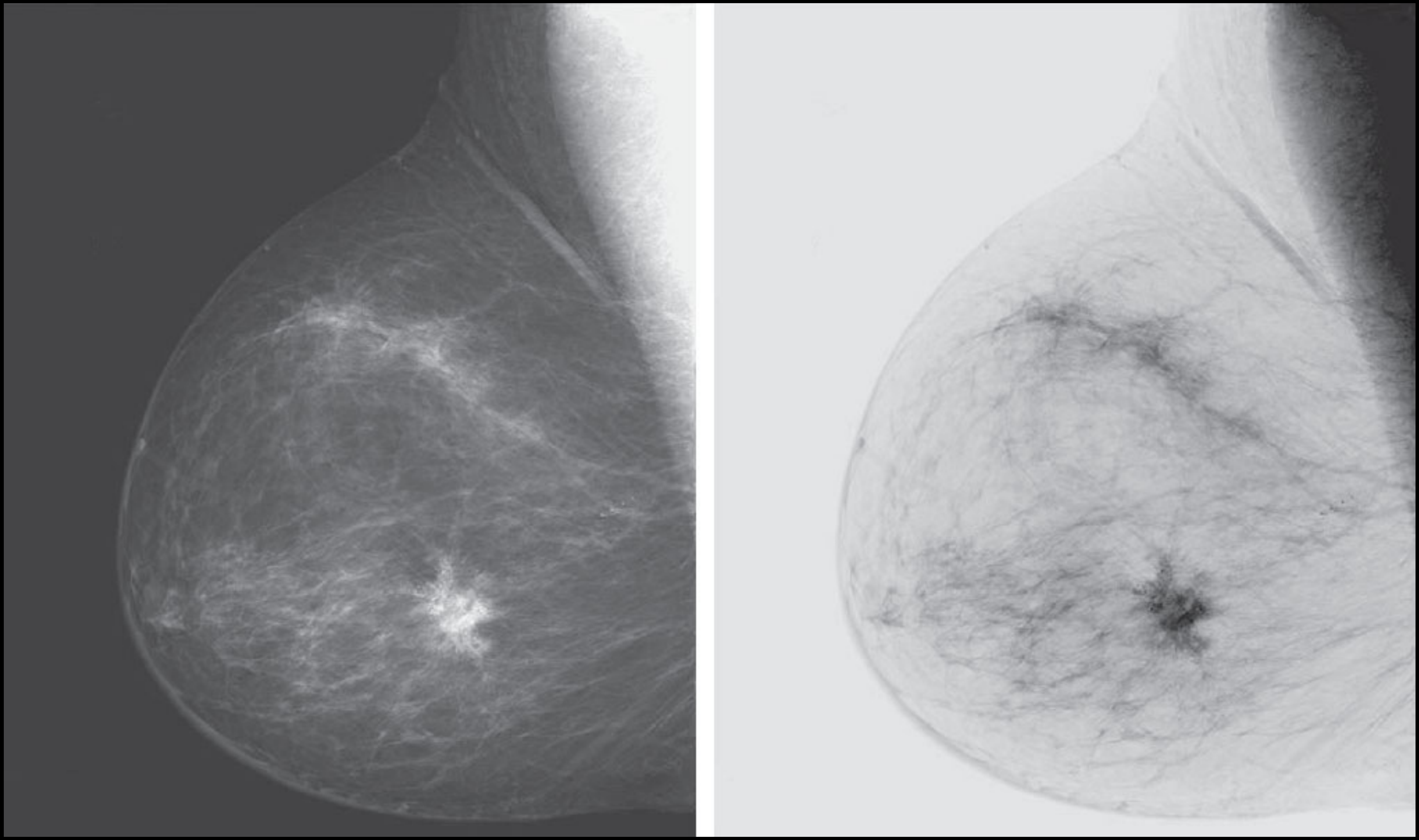
Chapter 3

Contrast and display scales

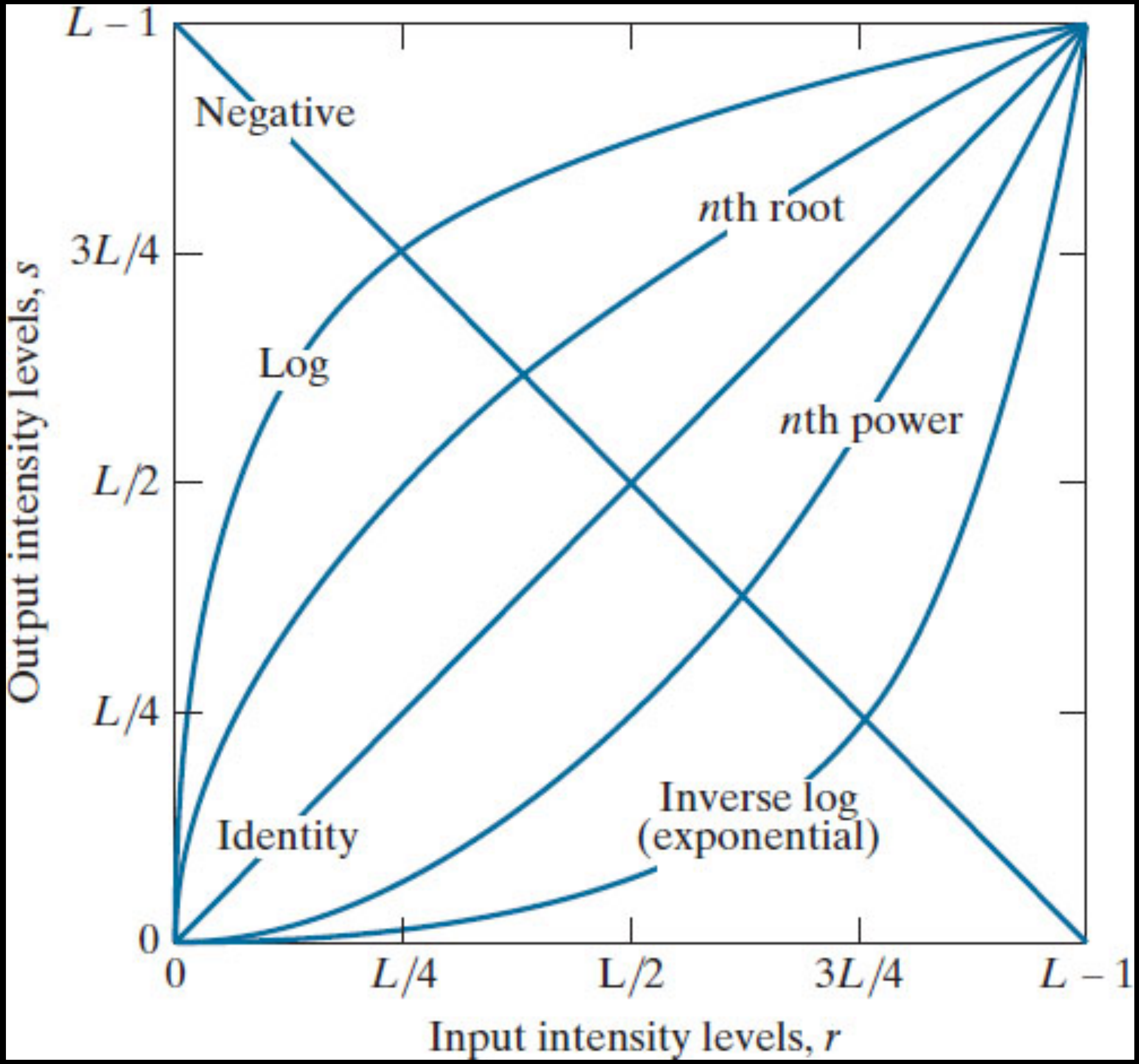
Threshold -> binarize

Inverting the image

Inverting can give better visibility for display



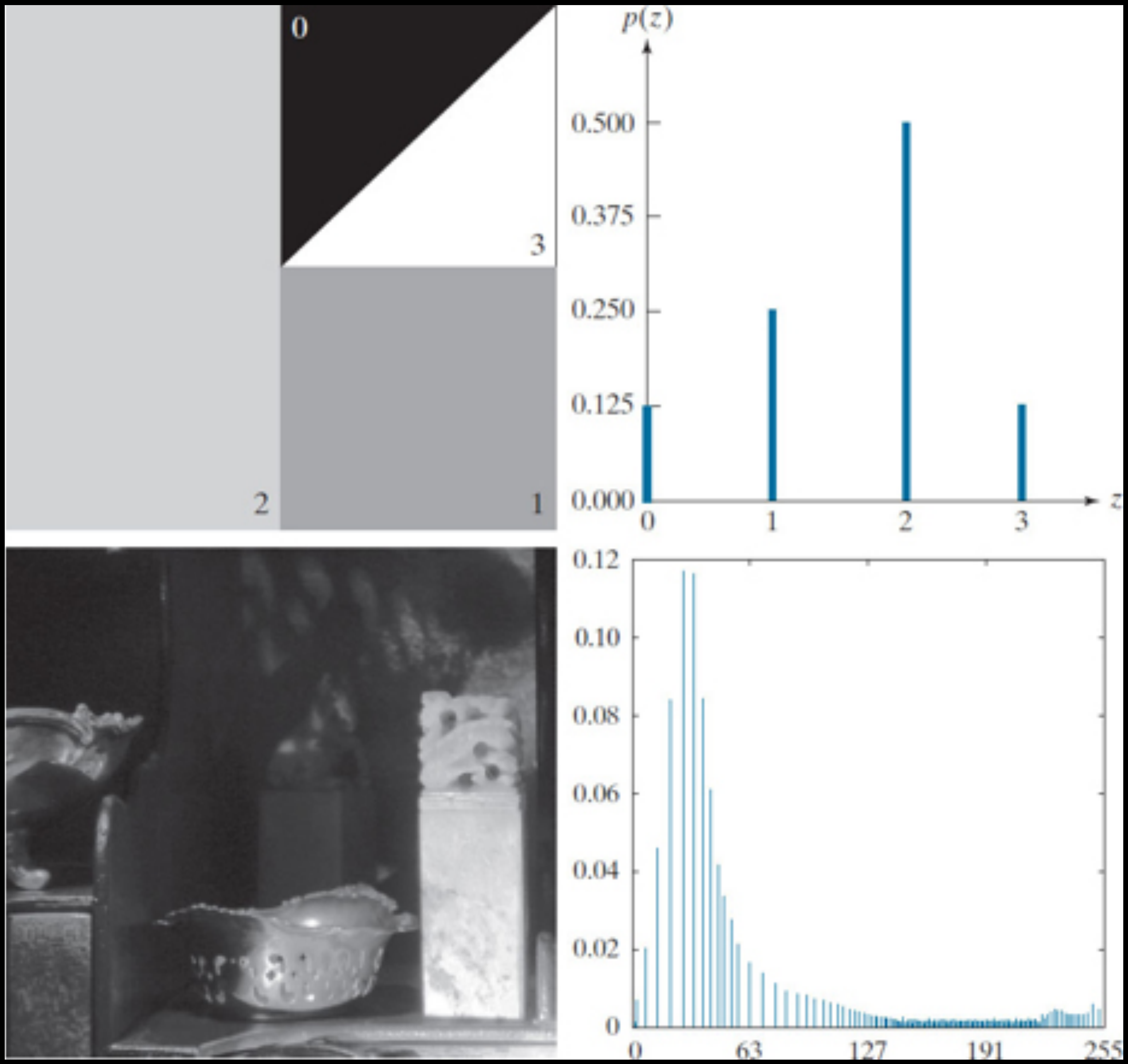
Scaling



L − 1
3L/4
L/2
L/4
0
Output intensity levels, s
Negative
Log
nth root
Identity
nth power
Inverse log
(exponential)
0
L/4
L/2
3L/4
L − 1
Input intensity levels, r

Log scale display

Histograms



0
3
2
1
p(z)
0.500
0.375
0.250
0.125
0.000
0
1
2
3
z
0.12
0.10
0.08
0.06
0.04
0.02
0
0
63
127
191
255

Measuring image contrast

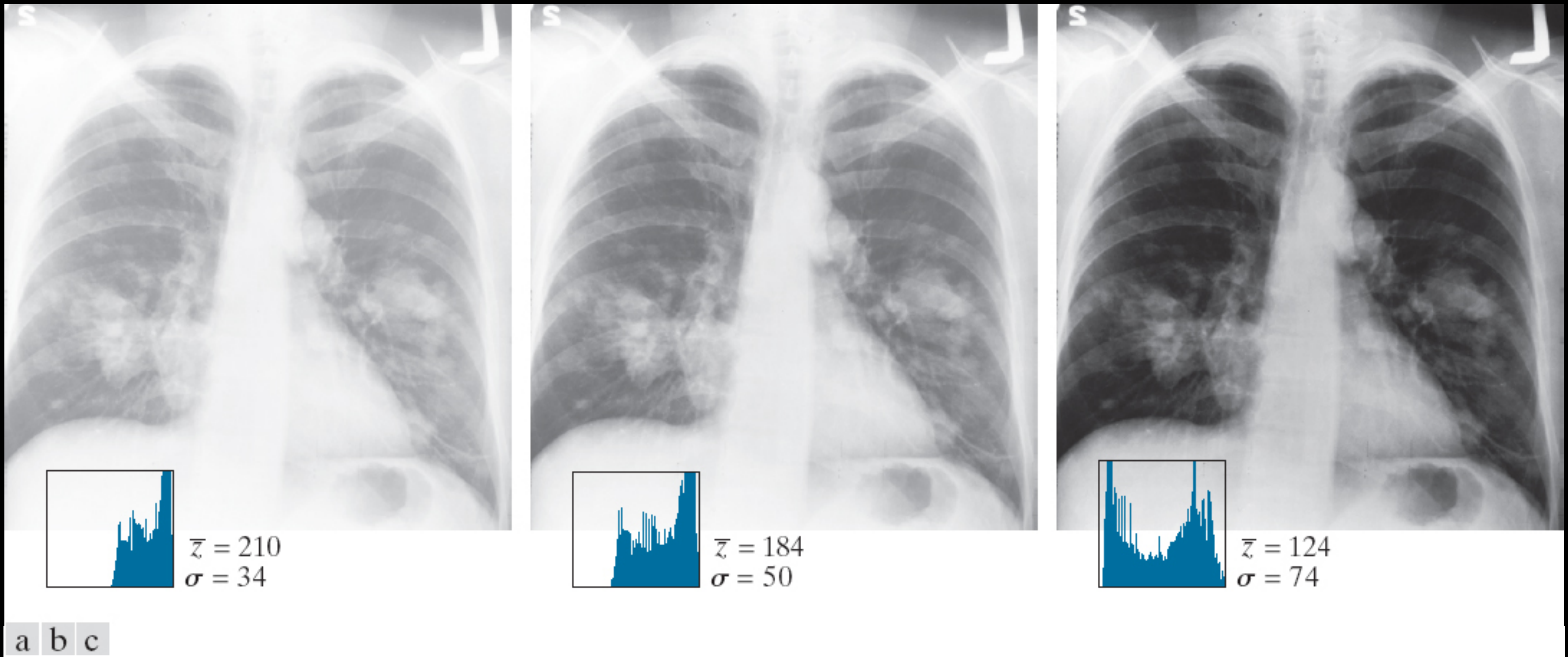


Illustration of the mean and standard deviation as functions of image contrast. (a)-(c) Images with low, medium, and high contrast, respectively. (Original image courtesy of the National Cancer Institute.)