

## Lecture 2 |

# Image Processing for Microscopy

Marco Cantoni

based on lecture notes from Prof. Vaso Tileli

# Today we will cover:

- Correcting defects (e.g. uneven illumination)
- Image **enhancement** (improving visualization using filters, kernels etc.)

histogram equalization, GAMMA value, kernel-filters

- Processing images in the frequency domain (FFT)
- Segmenting and thresholding
- Image measurements

# Correcting Defects

Image processing operations and procedures that can be applied to correct some of the defects or shortcomings in as-acquired images that may occur due to imperfect detectors, limitations of the optics, inadequate or **nonuniform illumination**, or an undesirable viewpoint.

**These are corrections applied **after** the image has been digitized and stored, and therefore are unable to deliver the highest quality result that could be achieved by optimizing or correcting the acquisition process in the first place.**

However, acquiring the optimum quality image can be impractical – reasons

- low dose imaging – noisy
- intensity variations – irregularities of surfaces
- outliers – x-rays, dead pixels

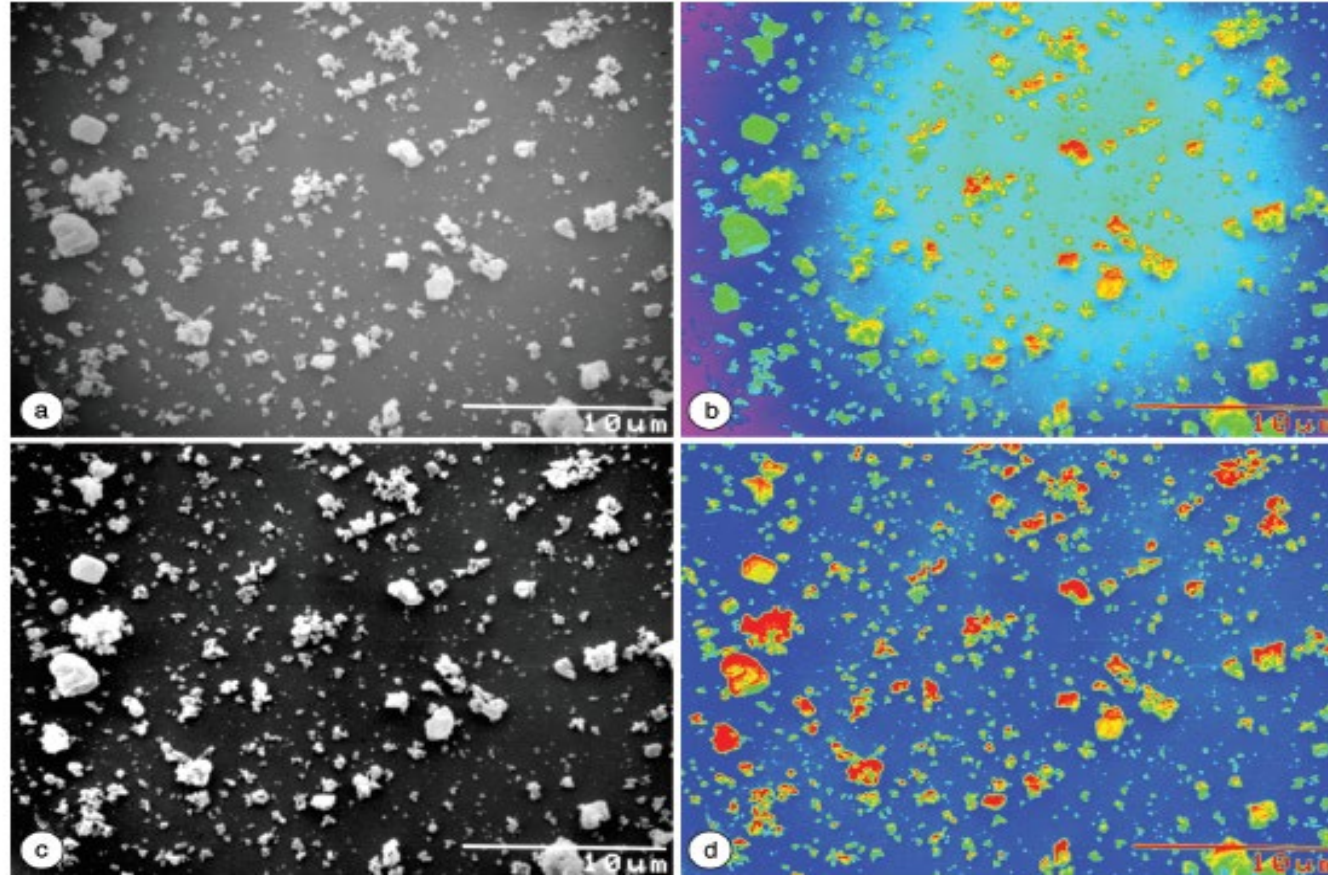
## Non-uniform illumination correction



*Figure 4.58 Macro image (a) with nonuniform illumination. The background image (b) was recorded under the same conditions with the specimen removed from the copy stand. Subtracting the background image pixel-by-pixel from the original and expanding the contrast of the result produces a leveled image (c).*

- Take an image from the background, and post-process the images - leveling

# Fitting a background function



*Figure 4.63 Leveling image contrast: (a) SEM image of particulates on a substrate; (b) false color applied to (a) to make the shading more visually evident; (c) image leveled by automatic polynomial fitting; (d) false color applied to (c).*

- Background points are selected automatically after enhancement using color
- The variation of the background is approximated by a polynomial
- Image is divided in many regions, and **darkest** pixels are found in region

# Image Enhancement

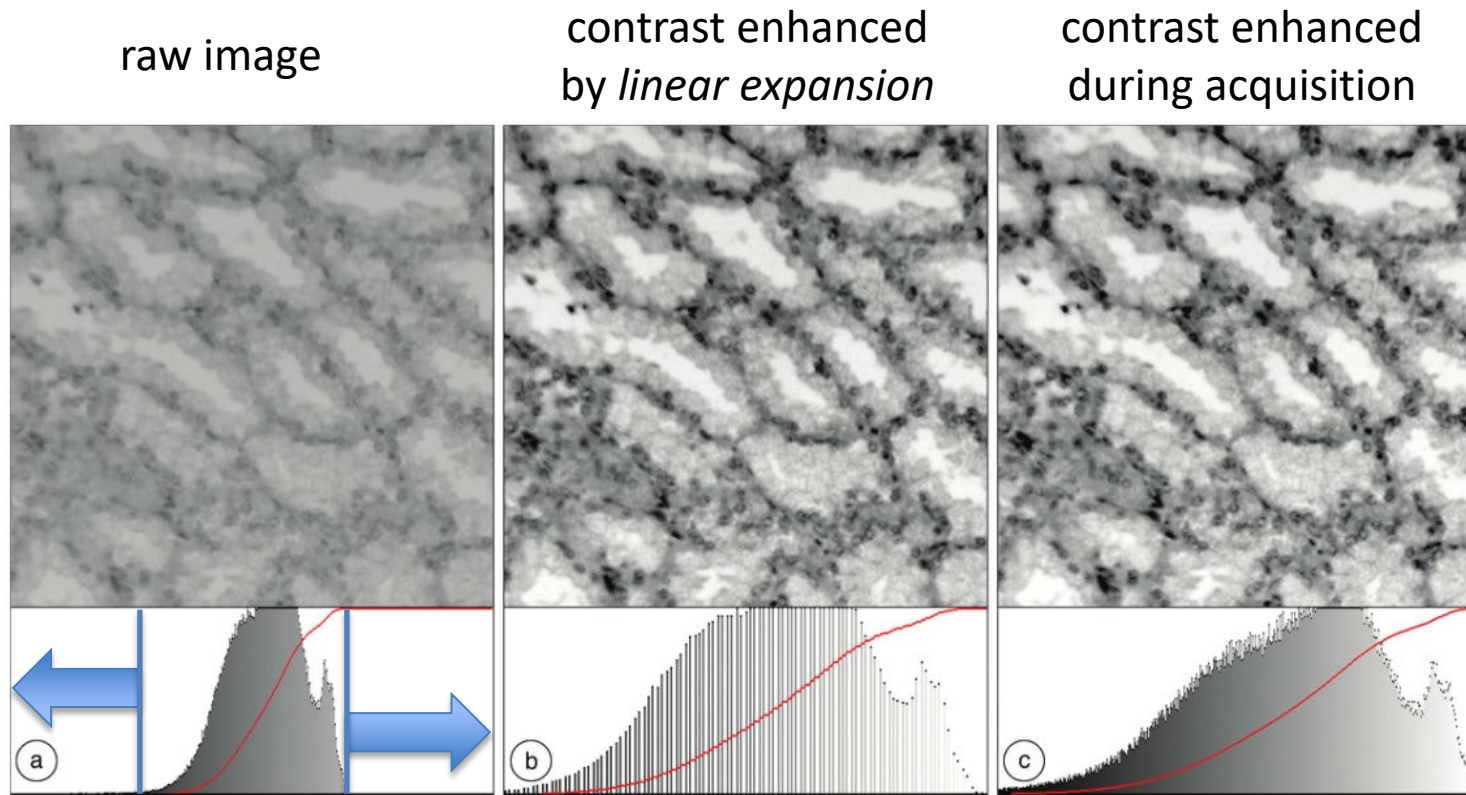
## Reasons to enhance image

- Make the image easier to visually examine and interpret
- Apply methods based on the physics of image interpretation
- Improve printing/documenting of images
- Allow automatic methods to perform measurements (segmentation, binarization)

# Contrast Expansion – Linear

Visibility of the structures present can be improved by stretching the contrast so that the values of the pixels are re-assigned to cover the entire available range

$$P' = 255 \cdot \frac{P - \text{Darkest}}{\text{Brightest} - \text{Darkest}}$$



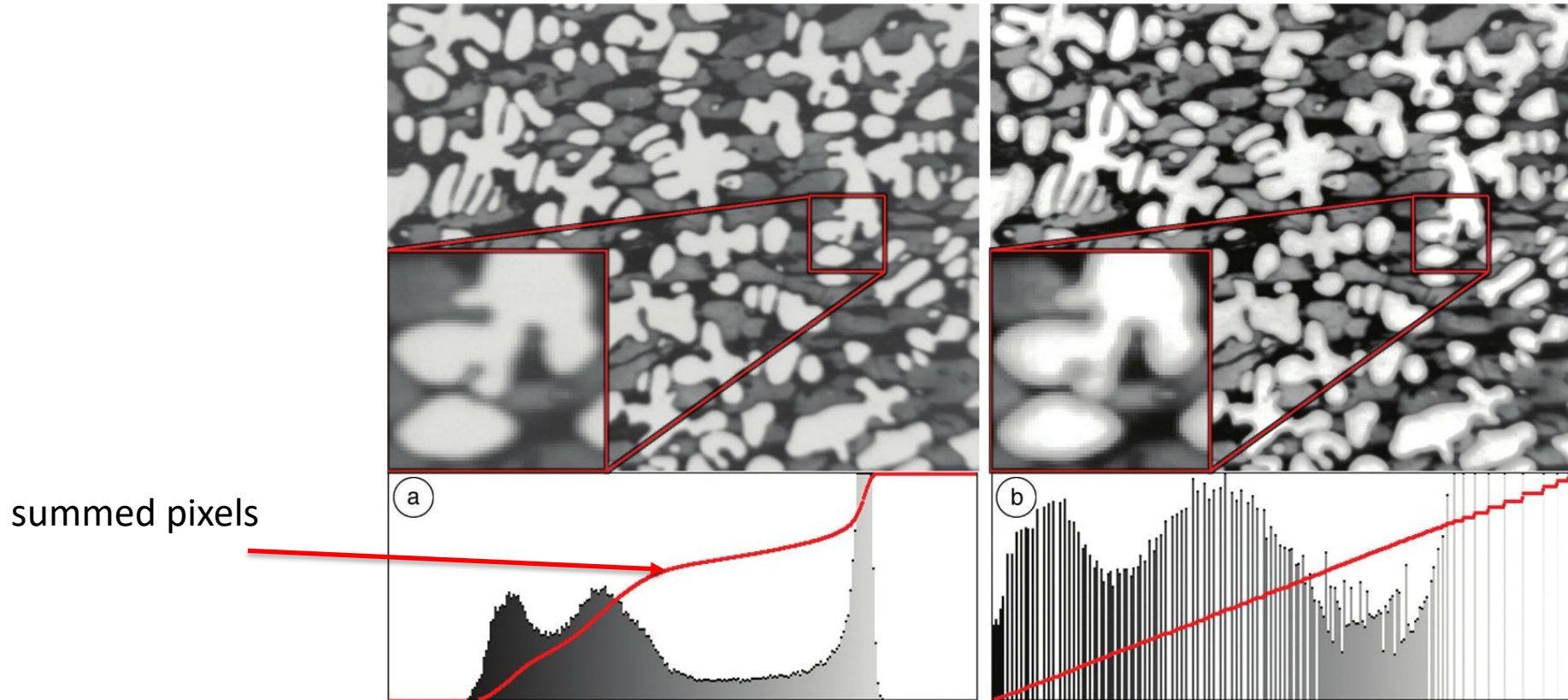
*Linear* contrast enhancement -> brightness and contrast, window levels



# Histogram Equalization

The data are plotted as a cumulative curve which is the integral or summation of the pixel values. Then this curve is used as the transfer function

$$f(j) = \frac{255}{T} \sum_{i=0}^j N_i$$



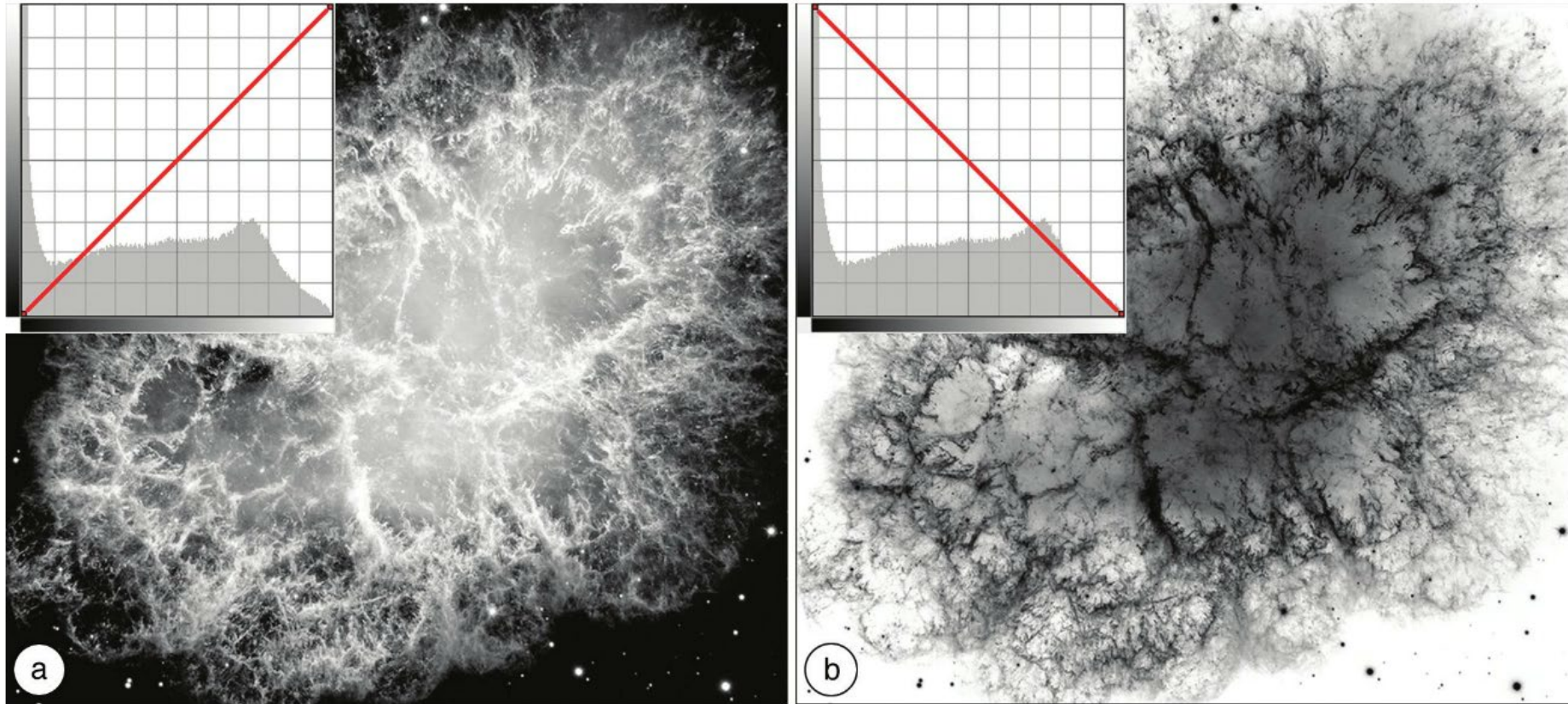
The rationale is to spread out the displayed brightness levels in the peak areas while compressing them in the valleys so that the same number of pixels in the display show each of the possible brightness levels



# Examples – Linear Manipulation

raw image

contrast **inverted** image



- The transfer function is a straight line
- Reversing the plot makes the human vision to discern details better

# Contrast Manipulation – *Non-Linear –Gamma Value*

The manipulation of pixel brightness can be described in terms of *a transfer function* relating the stored brightness value for each pixel to a displayed value

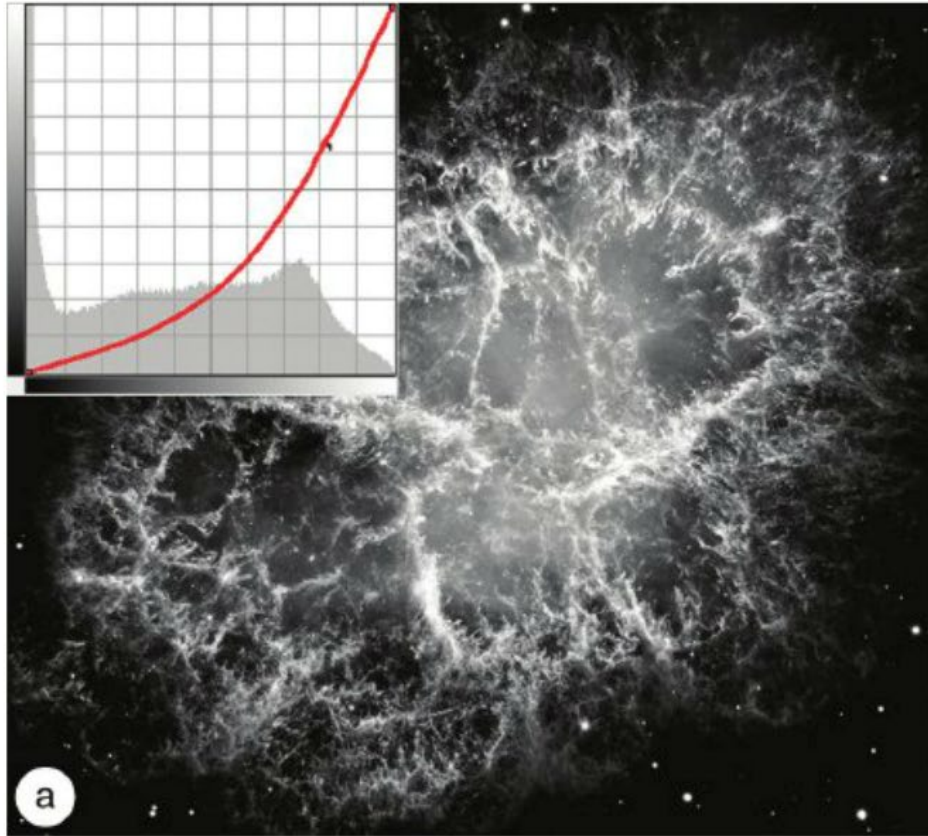
By altering the transfer function the contrast can be expanded so that some of the image details become more visible at the expense of compressing the contrast for others

It is all about manipulating the *Gamma value*

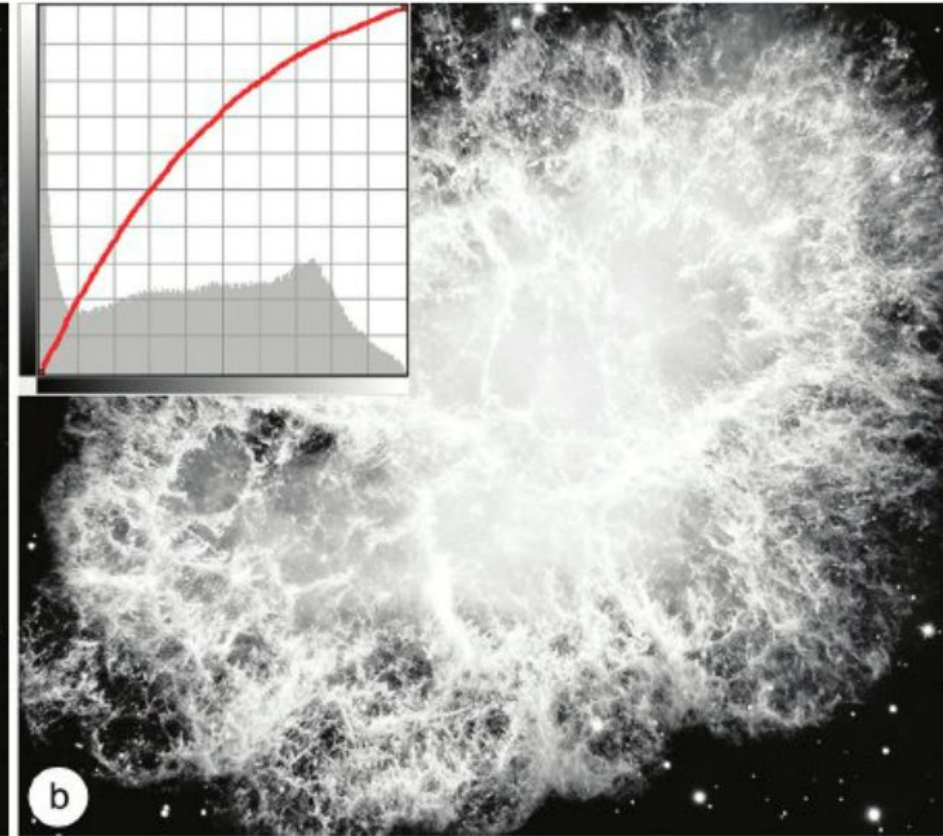
$$\left( \frac{\text{Display}}{255} \right) = \left( \frac{\text{Original}}{255} \right)^{\text{Gamma}}$$

# Examples – Non-Linear Manipulation

gamma > 1



gamma < 1



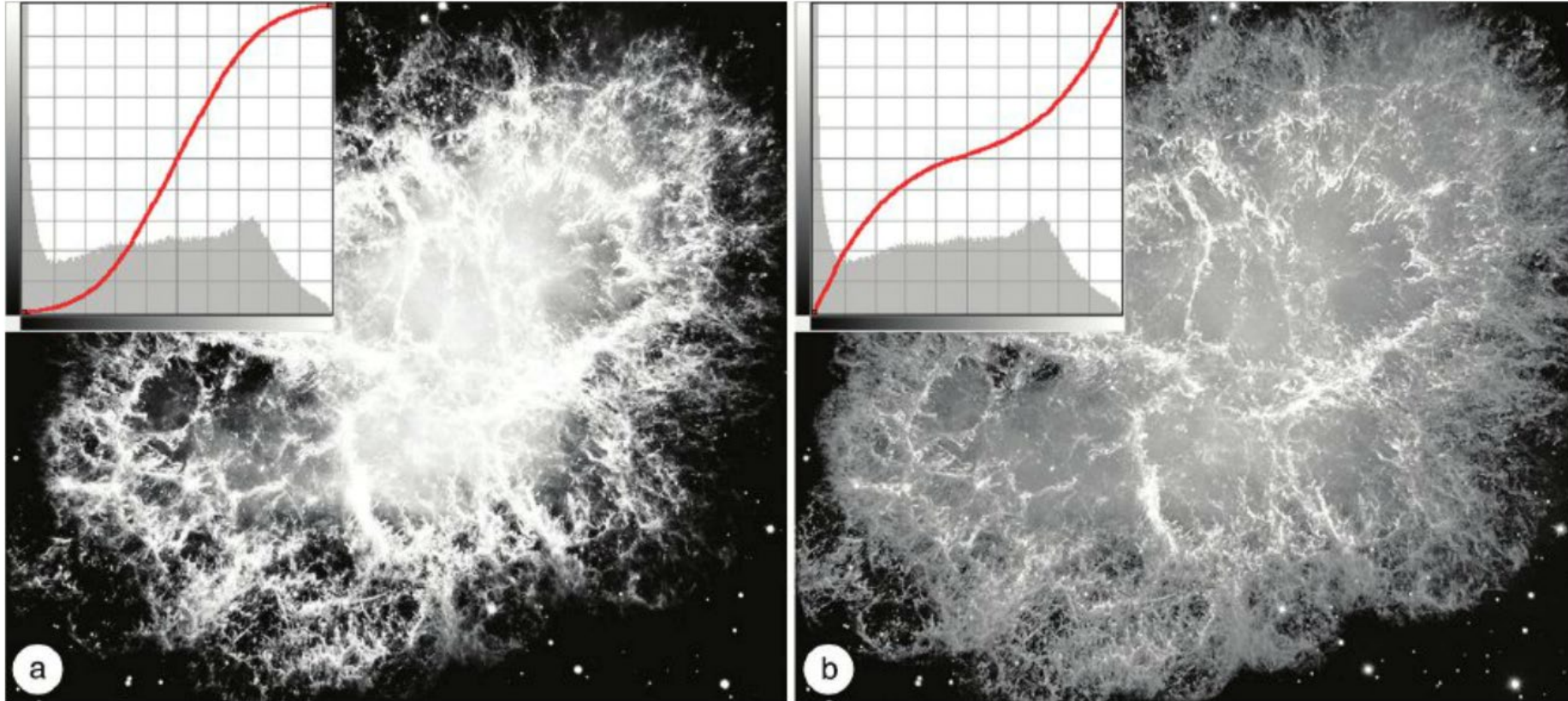
- (a) expansion for bright values – more details in the central area
- (b) expansion for the dark values – compresses the values in the bright areas



# Examples – Non-Linear Manipulation

expansion for middle grey values

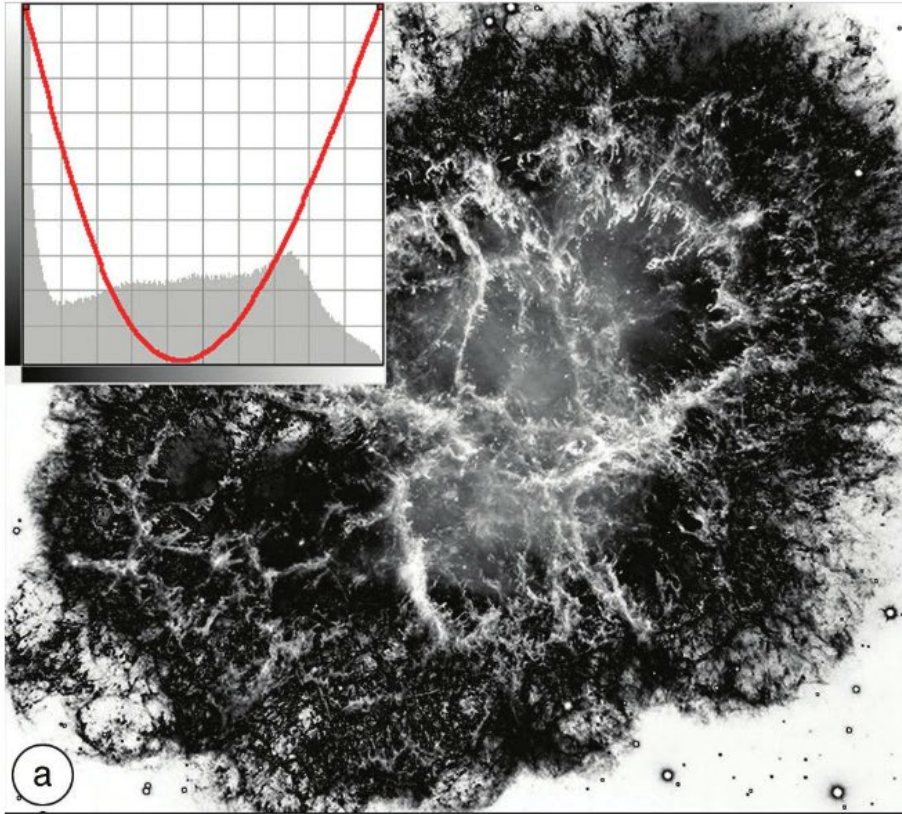
expansion for extreme grey values



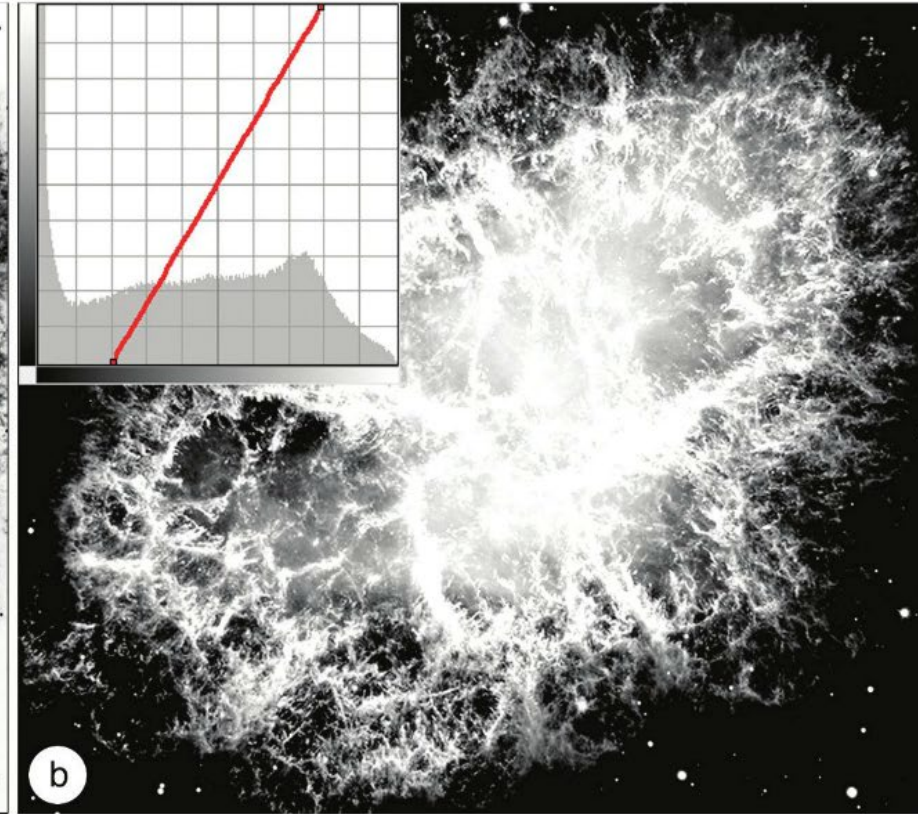
- Arbitrarily created transfer function can be more important or interesting
- (a) increased contrast in positive and negative regions
- (b) clipping extreme values to white and black expands the contrast in the middle

# Examples – Contrast Manipulation

partial reversal



clipping of histogram



Ultimately this is a flexible tool – the problem is that it is manual and depends on your judgment to determine which details are important – be aware of conscious or unconscious misconceptions on how the image should look like!



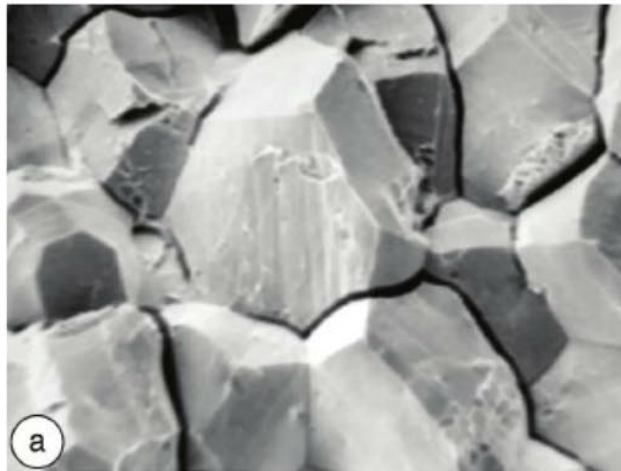
# Filters: Laplacian Sharpening

Local equalization of image contrast produces an increase in local contrast at boundaries, which has the effect of making easier for the viewer to see

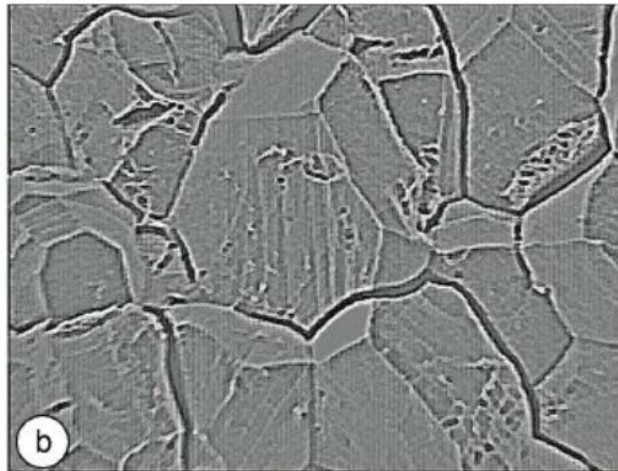
a 3x3 Laplacian convolution operator would be:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

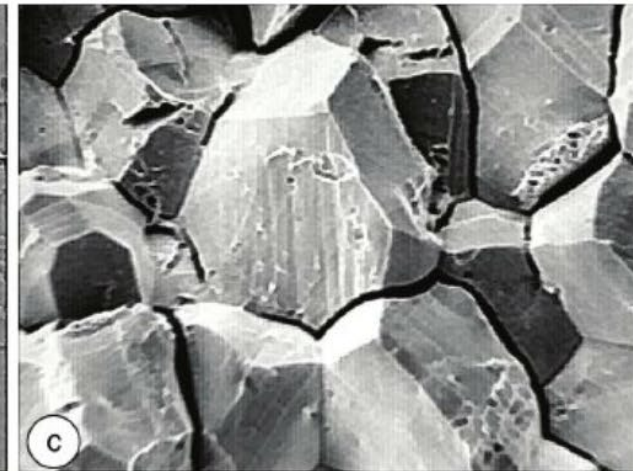
SEM raw image



Laplacian operator



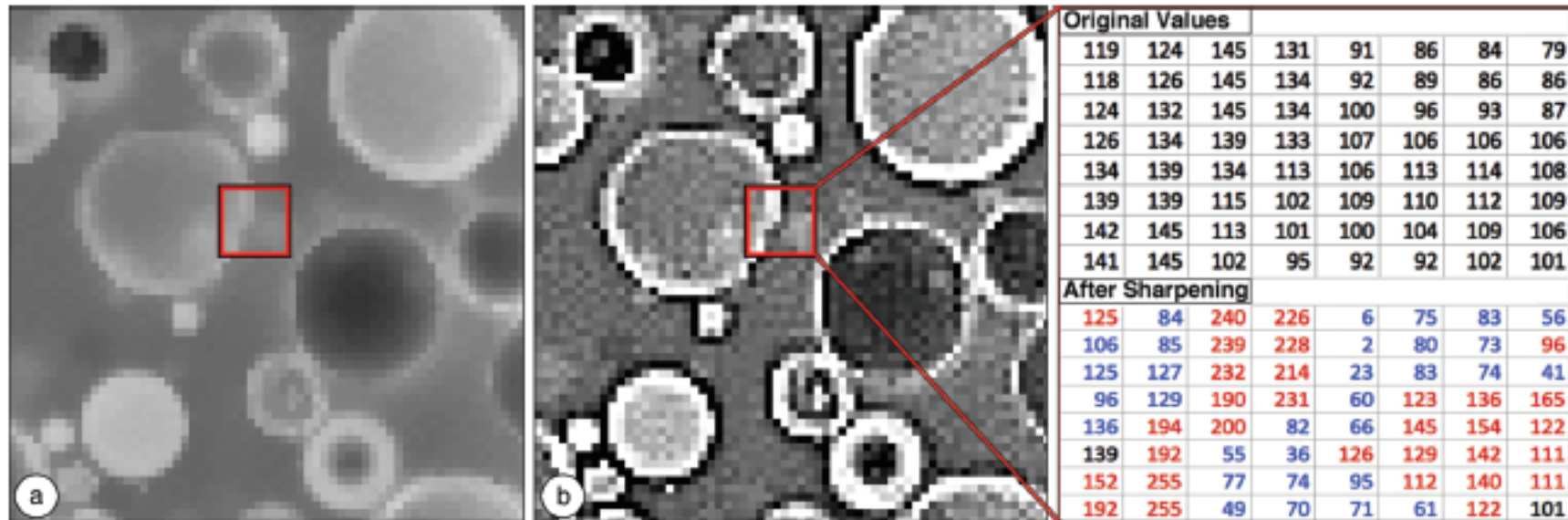
sharpening (a+b)



Unlike smoothing kernels, this operator represents the **second derivative** and it is not a weighted kernel, it highlights **DIFFERENCES**



# Sharpening kernel example



*Figure 5.23 Operation of the  $3 \times 3$  sharpening filter: (a) fragment of an original image (an epoxy containing bubbles); (b) after application of the filter the edges are delineated by bright and dark borders. The pixel values are shown for the outlined region. On the dark side of the edge, pixel values are reduced (blue), and on the bright side, they are increased (red).*

- The effect of the filter is to make the pixels on the dark side of an edge darker and those on the bright side brighter, thus increasing the edge contrast

# Non-symmetric Filters: Edge Detection

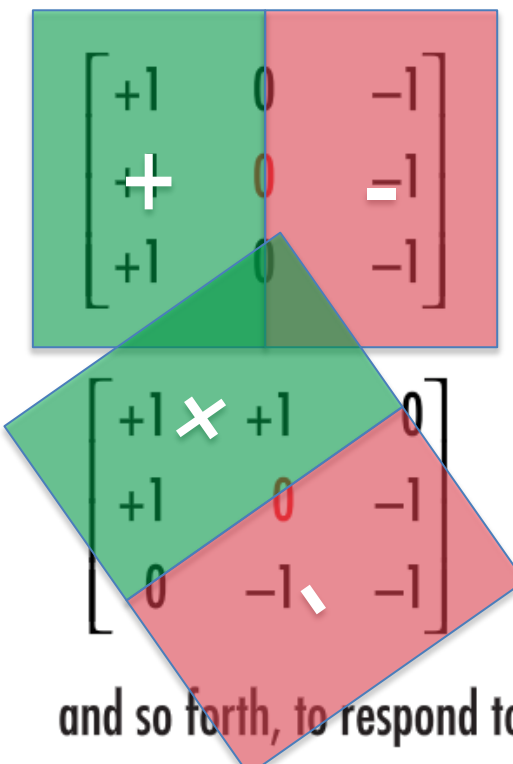
The process of locating boundaries with arbitrary orientation in two dimensional images -> it refers to taking **first derivatives** of brightness in different orientations

| (Robinson)                                                                         | (Sobel)                                                                            | (Prewitt)                                                                             | (Kirsch)                                                                             |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| $\begin{bmatrix} +1 & 0 & -1 \\ +1 & \mathbf{0} & -1 \\ +1 & 0 & -1 \end{bmatrix}$ | $\begin{bmatrix} +1 & 0 & -1 \\ +2 & \mathbf{0} & -2 \\ +1 & 0 & -1 \end{bmatrix}$ | $\begin{bmatrix} +1 & -1 & -1 \\ +1 & \mathbf{+2} & -1 \\ +1 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} +5 & -3 & -3 \\ +5 & \mathbf{0} & -3 \\ +5 & -3 & -3 \end{bmatrix}$ |
| $\begin{bmatrix} +1 & +1 & 0 \\ +1 & \mathbf{0} & -1 \\ 0 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} +2 & +1 & 0 \\ +1 & \mathbf{0} & -1 \\ 0 & -1 & -2 \end{bmatrix}$ | $\begin{bmatrix} +1 & +1 & -1 \\ +1 & \mathbf{+2} & -1 \\ -1 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} +5 & +5 & -3 \\ +5 & \mathbf{0} & -3 \\ -3 & -3 & -3 \end{bmatrix}$ |

and so forth, to respond to edges in various orientations.

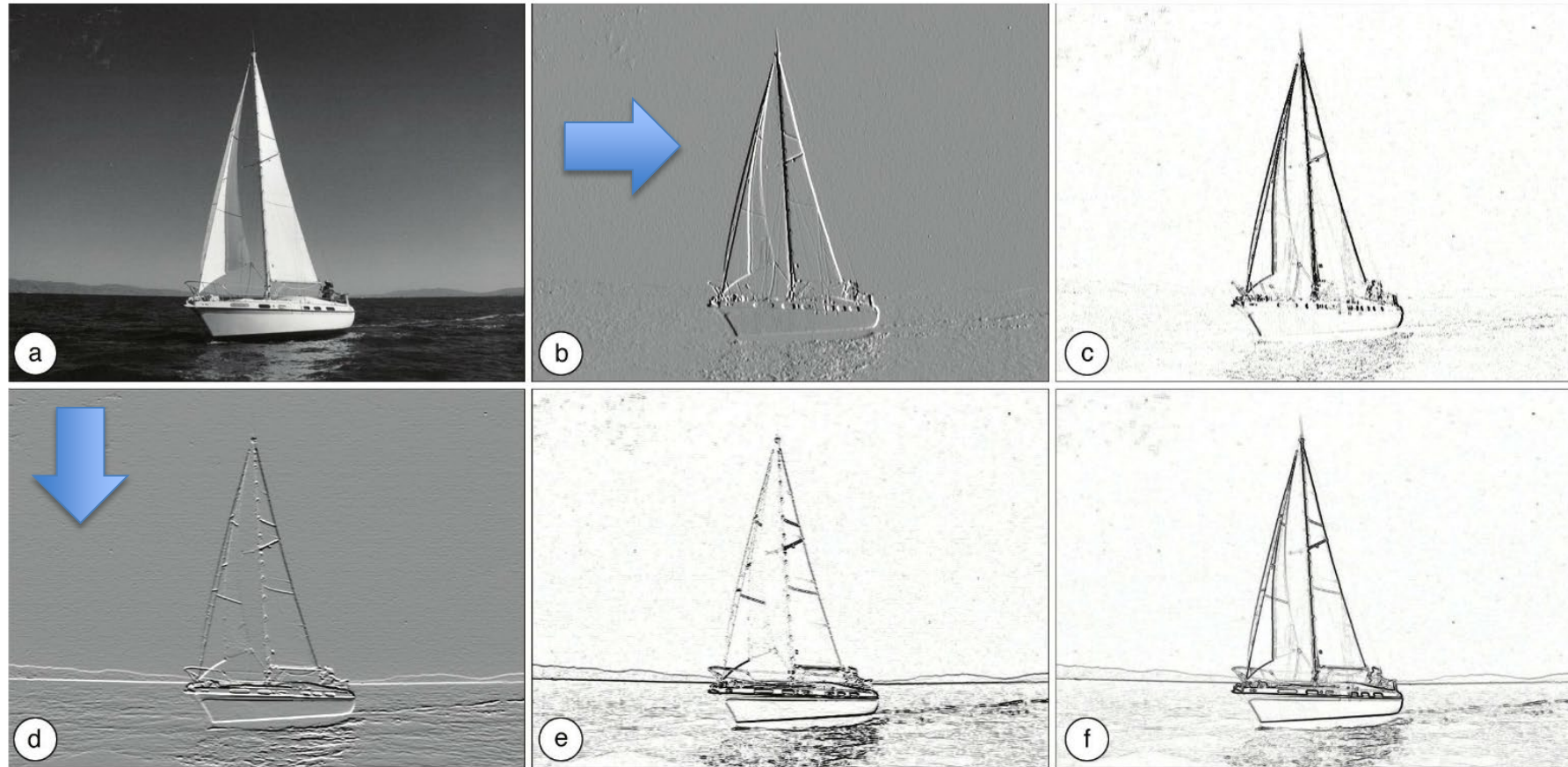
# Filters: Edge Detection

The process of locating boundaries with arbitrary orientation in two dimensional images -> it refers to taking **first derivatives** of brightness in different orientations

| (Robinson)                                                                                                                                                   | (Sobel)                                                                   | (Prewitt)                                                                    | (Kirsch)                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  $\begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix}$ | $\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$ | $\begin{bmatrix} +1 & -1 & -1 \\ +1 & +2 & -1 \\ +1 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} +5 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & -3 & -3 \end{bmatrix}$ |
| $\begin{bmatrix} +1 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$                                                                                    | $\begin{bmatrix} +2 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$ | $\begin{bmatrix} +1 & +1 & -1 \\ +1 & +2 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} +5 & +5 & -3 \\ +5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$ |

and so forth, to respond to edges in various orientations.

# Filters: Sobel Edge Detection



**Figure 5.38** Directional derivatives and the Sobel edge operator: (a) original image; (b) horizontal derivative (“embossed” image); (c) absolute value of the horizontal derivative; (d) vertical derivative (“embossed” image); (e) absolute value of the vertical derivative; (f) combining the derivative magnitudes by squaring, adding, and taking the square root (Sobel operator).



# Filters: Sobel Edge Detection – How it works

The operator consists of a pair of  $3 \times 3$  convolution kernels; One kernel is simply the other rotated by  $90^\circ$ . Two  $3 \times 3$  convolution kernels (shown below) are used to generate vertical and horizontal derivatives. The final image is produced by combining the two derivatives using the square root of the sum of the squares.

|    |   |    |
|----|---|----|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

G<sub>x</sub>

|    |    |    |
|----|----|----|
| +1 | +2 | +1 |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

G<sub>y</sub>

|                |                |                |
|----------------|----------------|----------------|
| P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> |
| P <sub>4</sub> | P <sub>5</sub> | P <sub>6</sub> |
| P <sub>7</sub> | P <sub>8</sub> | P <sub>9</sub> |

$$G = \sqrt{G_x^2 + G_y^2}$$

Using this information, we can also calculate the gradient's direction:

$$\Theta = \text{atan2}(G_y, G_x)$$

$$|G| = |G_x| + |G_y|$$

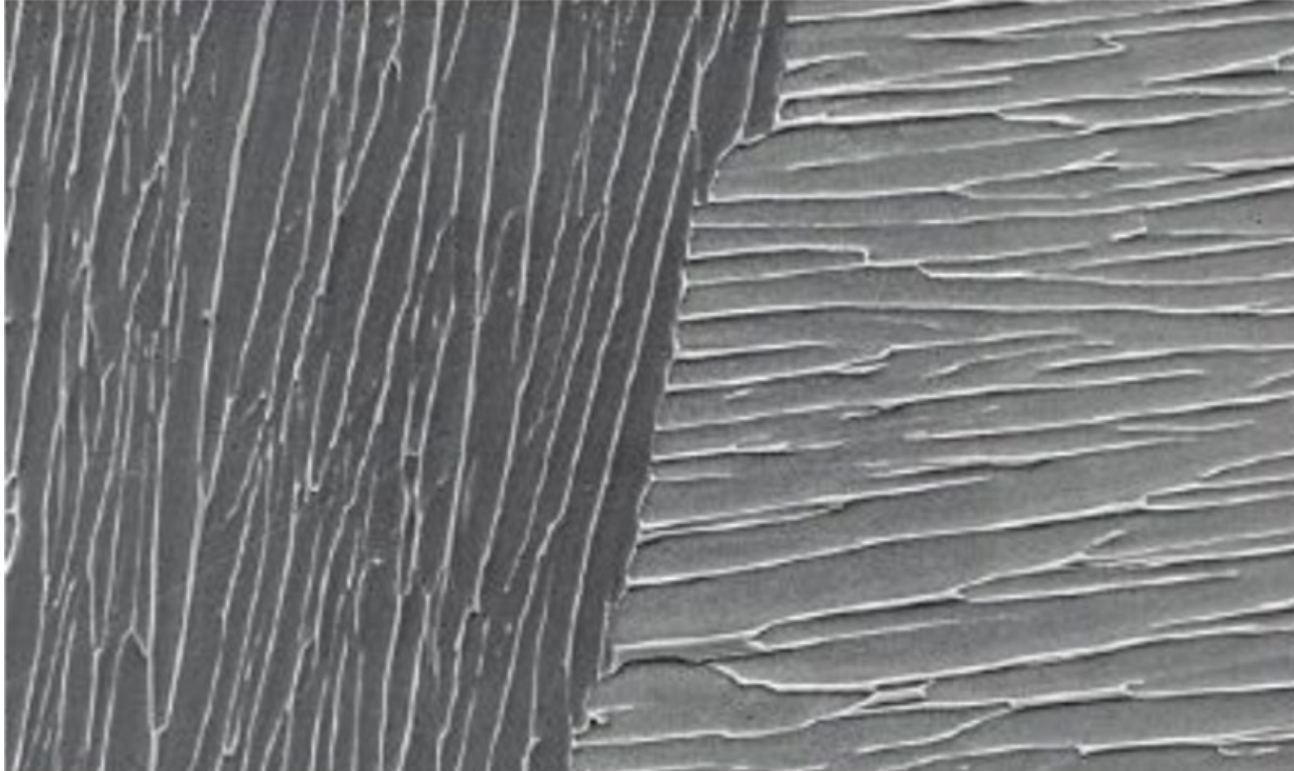
**Gradient: Amplitude G and direction  $\theta$**

$$|G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)|$$

In ImageJ this function is under “Find Edges”

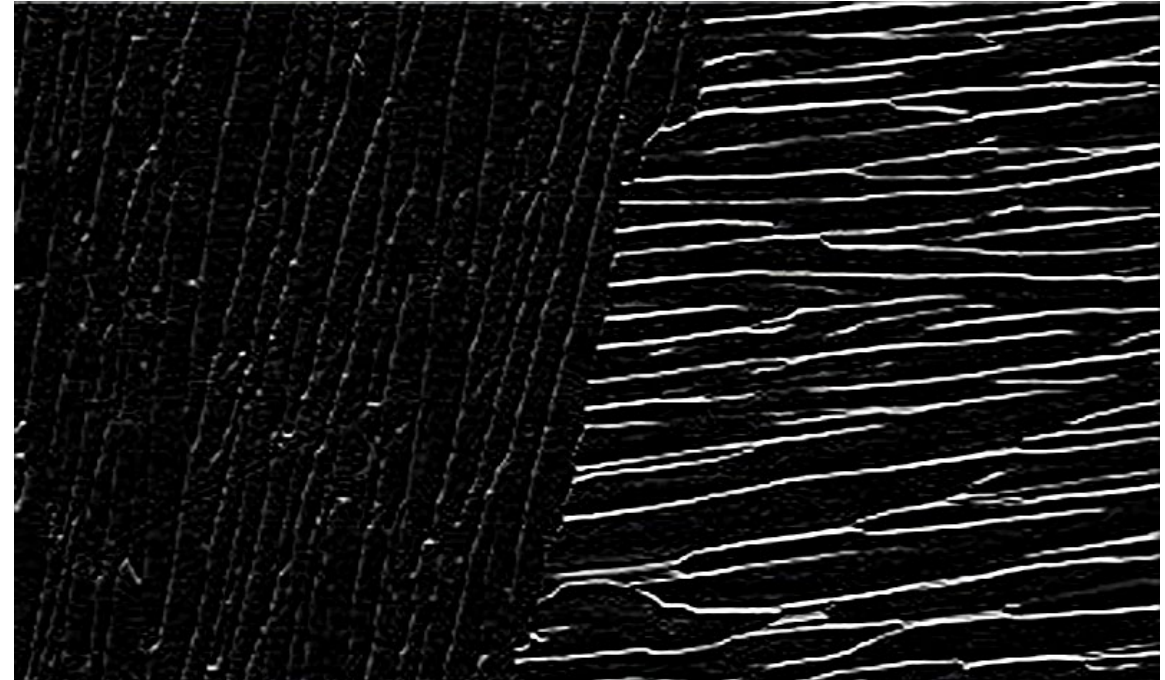
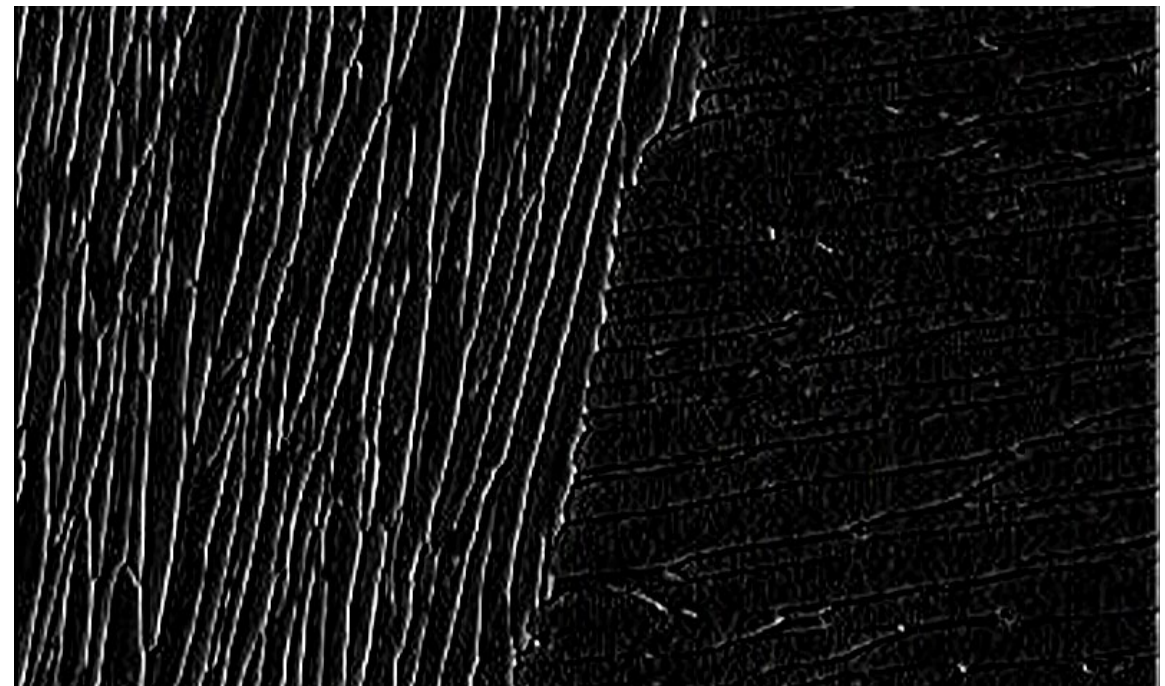
Gx=

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |



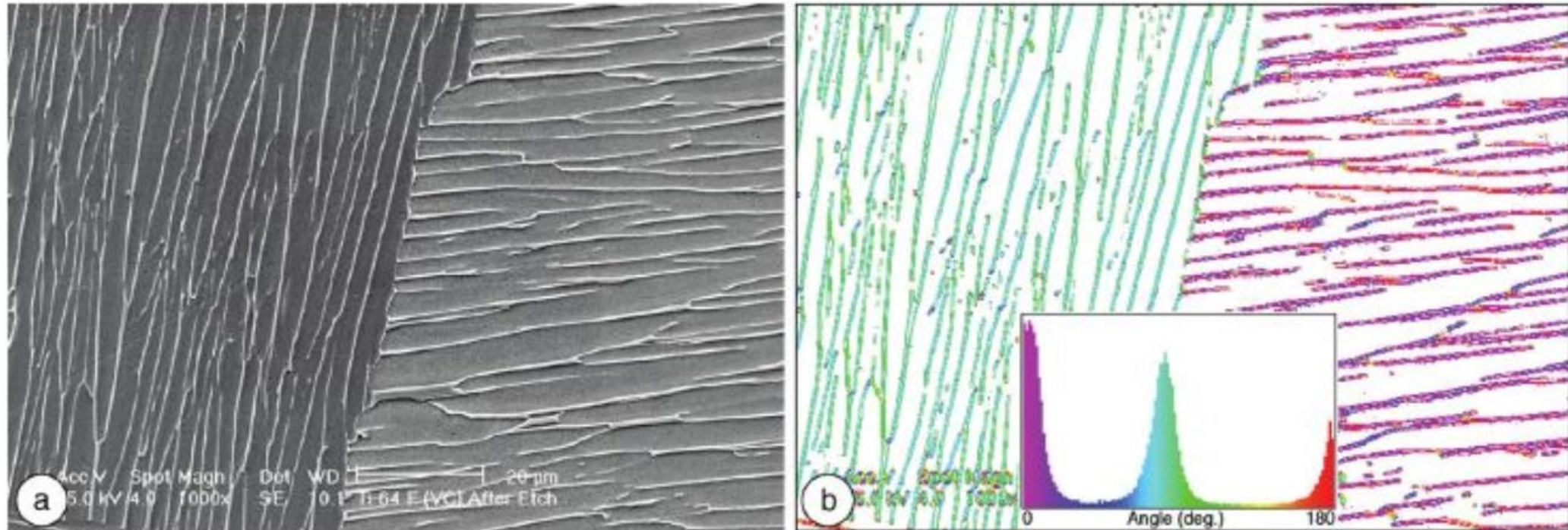
Gy=

|    |    |    |
|----|----|----|
| -1 | -2 | -1 |
| 0  | 0  | 0  |
| 1  | 2  | 1  |





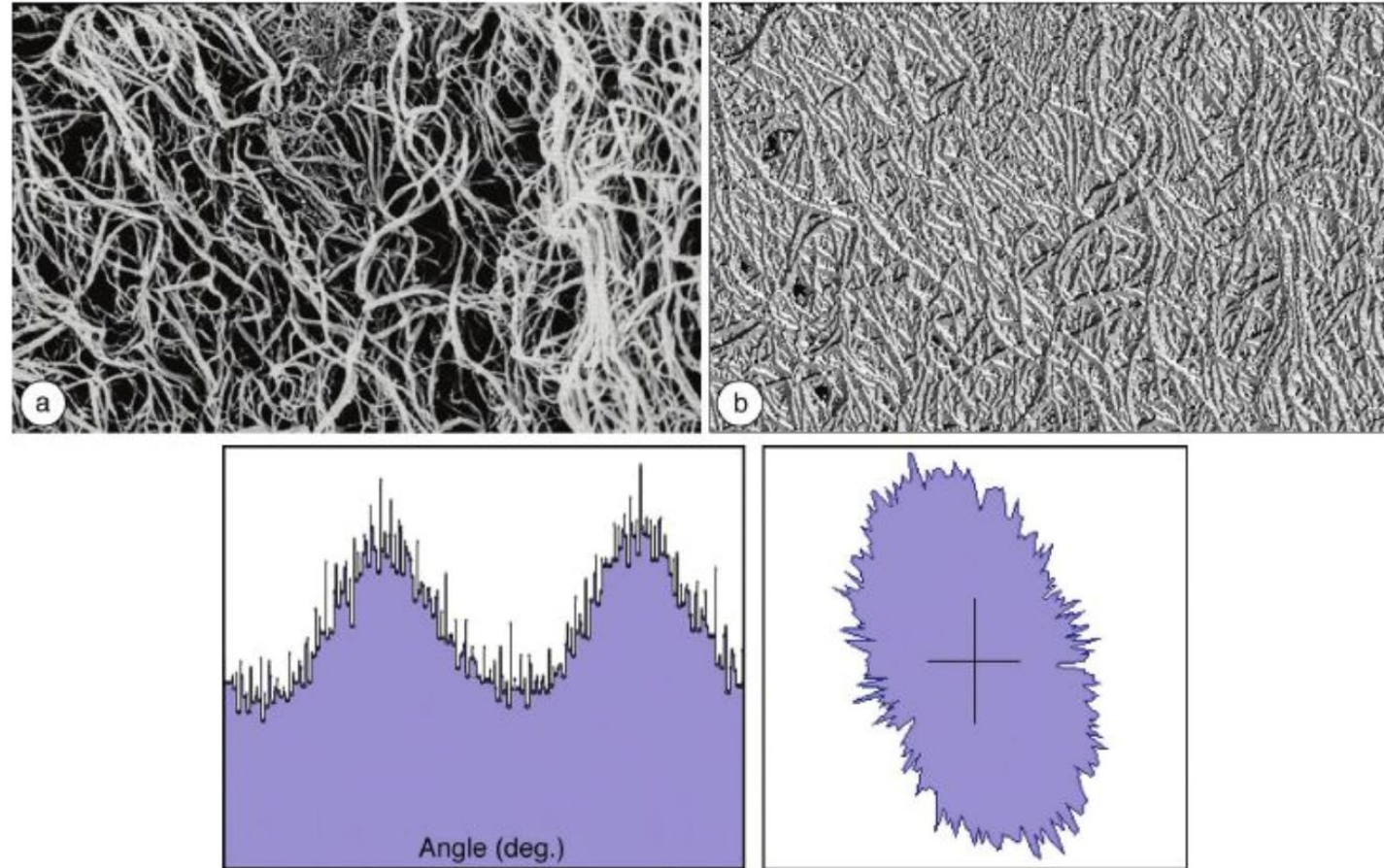
# Sobel Edge Detection – Example 1



**Figure 5.42** Lamellar structure in a titanium alloy: (a) SEM image; (b) edges selected by magnitude of the Sobel gradient and color-coded with orientation, with the histogram of edge orientations from 0 to 180 degrees. (SEM image courtesy of Hamish Fraser, Ohio State University.)

- Setting a threshold value at the magnitude of the vector  $|G| > G_0$  and keeping the **angle** information defined by  $(G_x, G_y)$  for those pixels above the threshold

# Sobel Edge Detection – Example 1



**Figure 5.43** Measurement of fiber orientation: (a) image of collagen fibers; (b) the Sobel orientation values assigned to pixel brightness. The histograms show the values in (b) from 0 to 360 degrees and as a rose plot.

- The efficiency with which the Sobel operator can be applied make it a very useful technique for characterizing the anisotropy of lines and edges



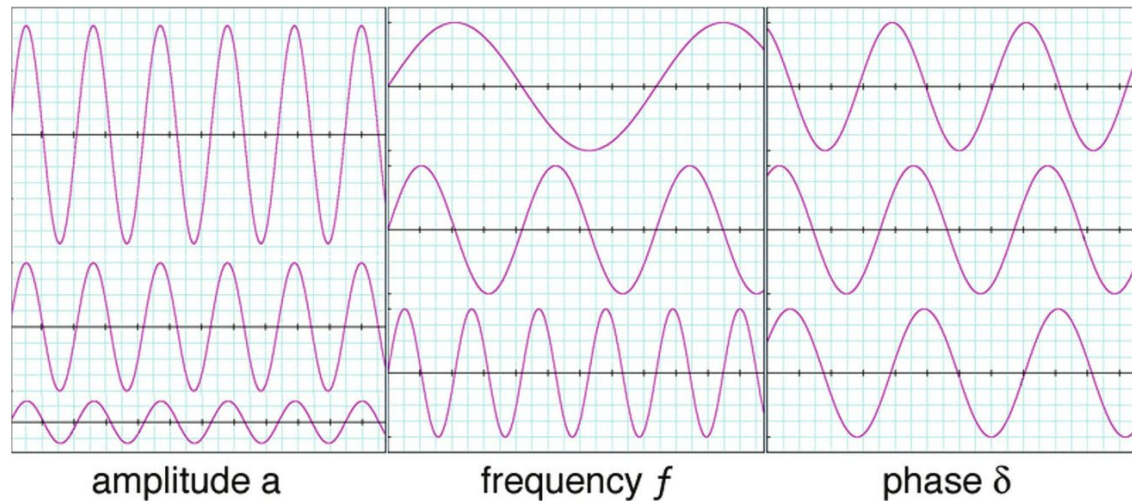
# Processing Images in Frequency Space

## Fourier Transform Basics

Usually, the mathematical background of a Fourier Transform begins with a one-dimensional waveform and then expands to two dimensions (image).

Fourier's theorem (1822) states that it is possible to construct any well-behaved one-dimensional function  $f(x)$  as a summation of a series of sine and cosine terms of increasing frequency,  $F(u)$

$$F(u) = \int_{-\infty}^{+\infty} f(x) \cdot e^{-2\pi i u x} dx$$



**Figure 6.1** Varying the parameters for the sine function  $y = a \cdot \sin(f \cdot 2\pi x + \delta)$ , where  $a$  is the amplitude,  $f$  is the frequency, and  $\delta$  is the phase.

**Origin: application in signal processing**



# Fourier Transform Basics

In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image

The image in the spatial and Fourier domain are of the same size

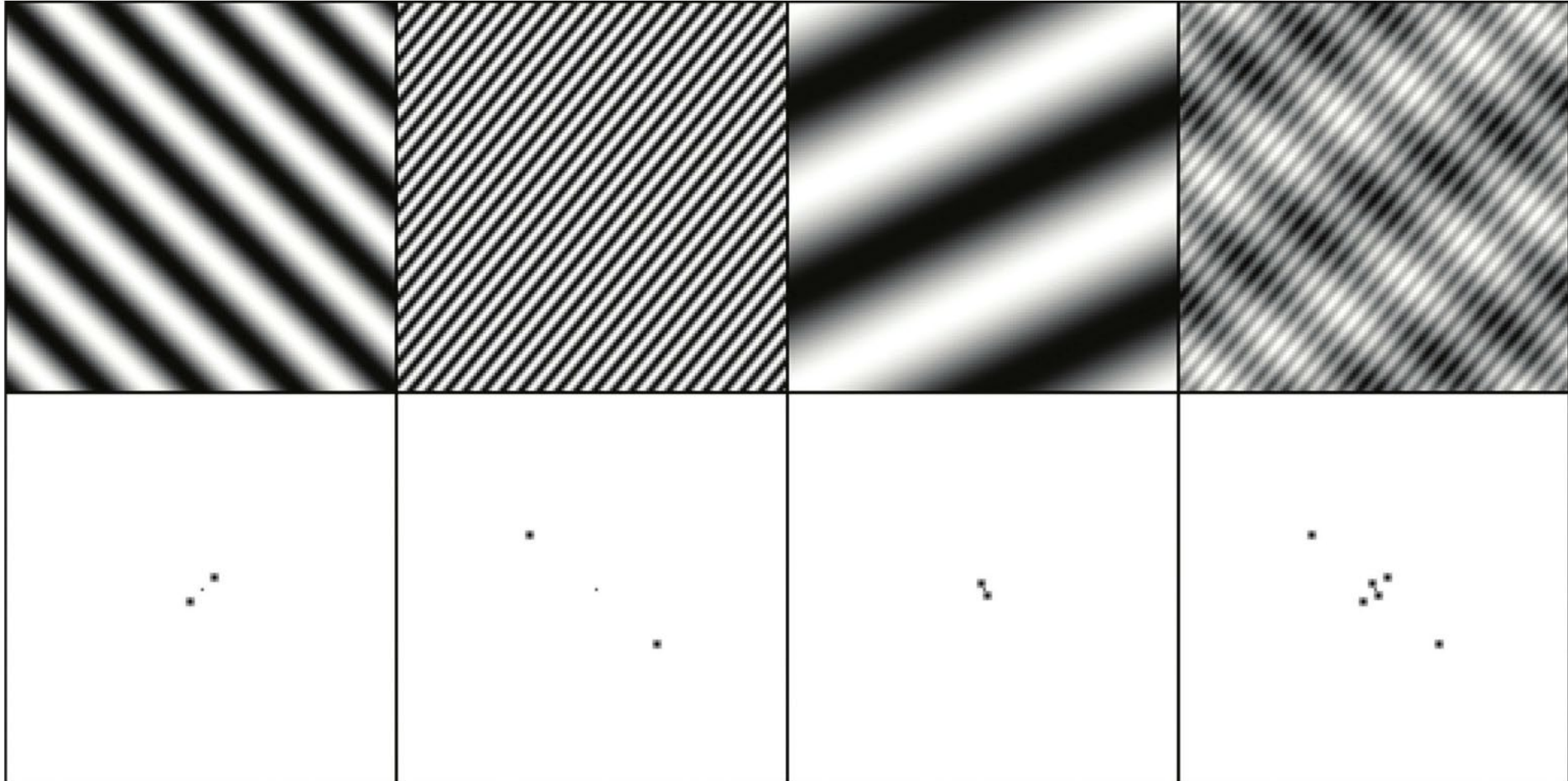
For a square image of size  $N \times N$ , the two dimensional FFT is:

$$F(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) e^{-i2\pi(\frac{ki}{N} + \frac{lj}{N})}$$

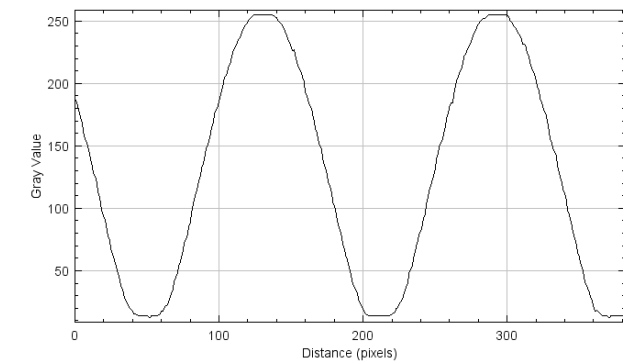
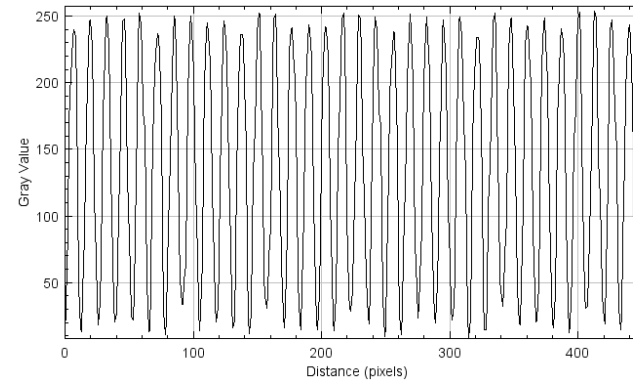
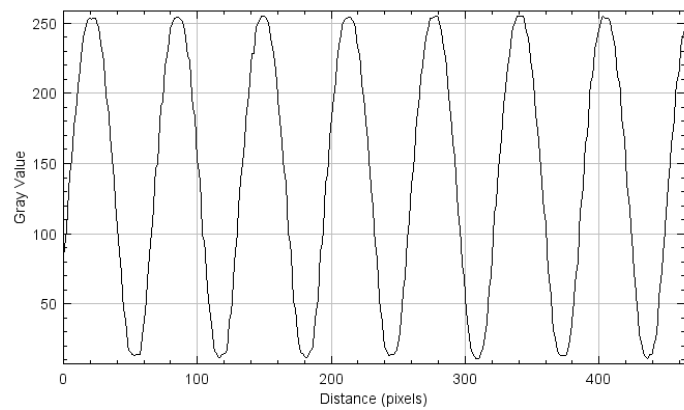
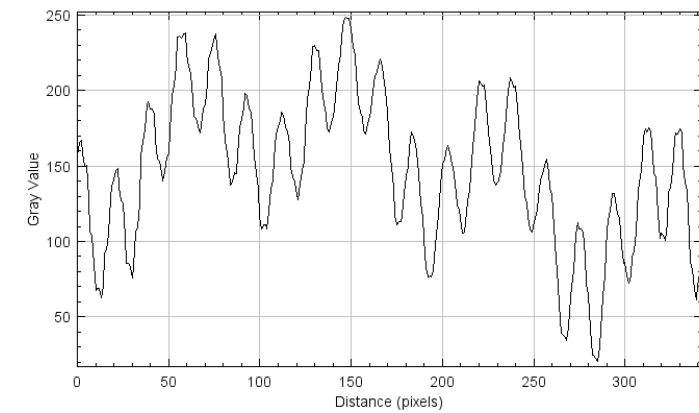
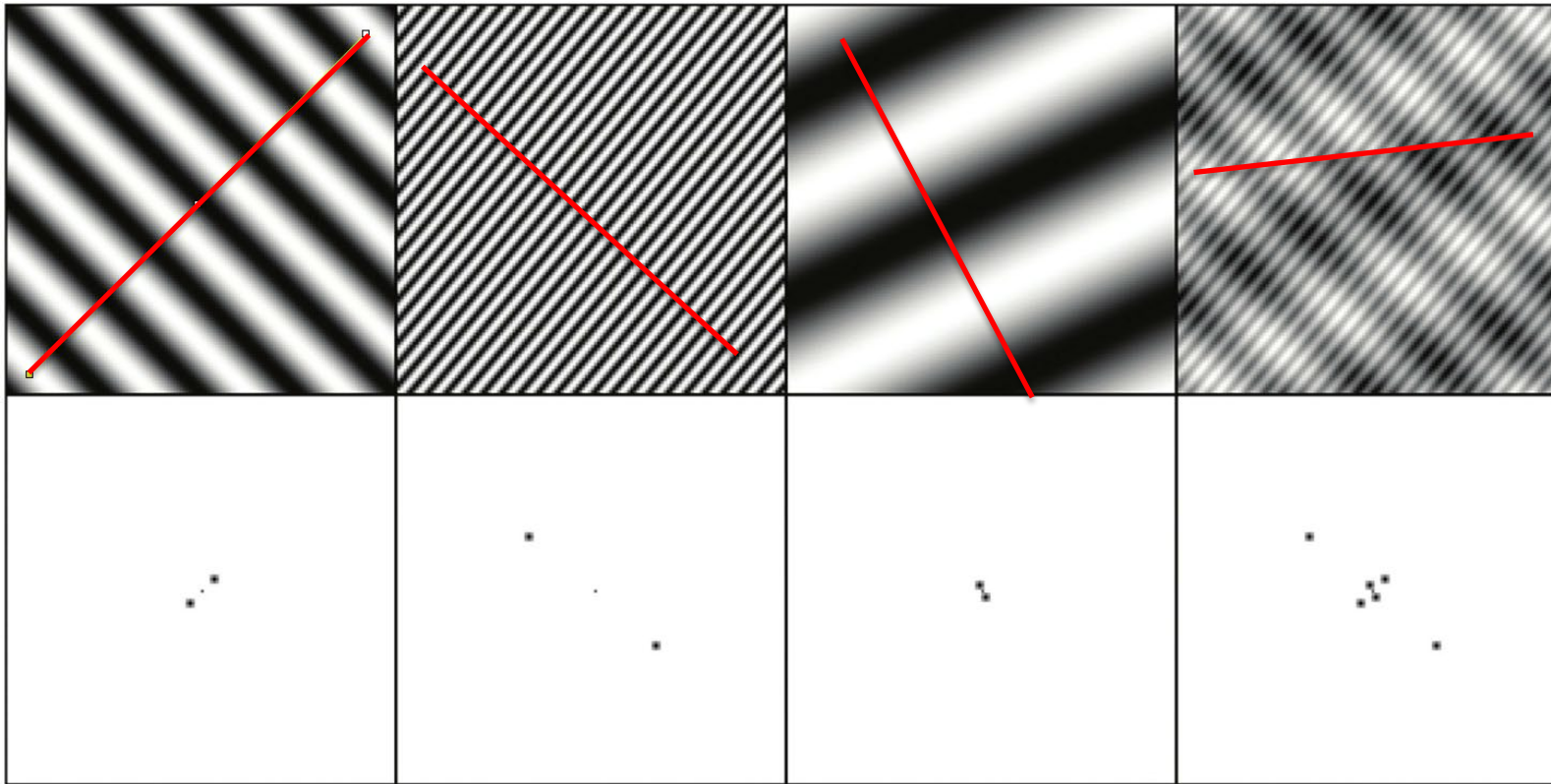
the basis functions are sine and cosine waves with increasing frequencies

# Two dimensional Fourier Transform

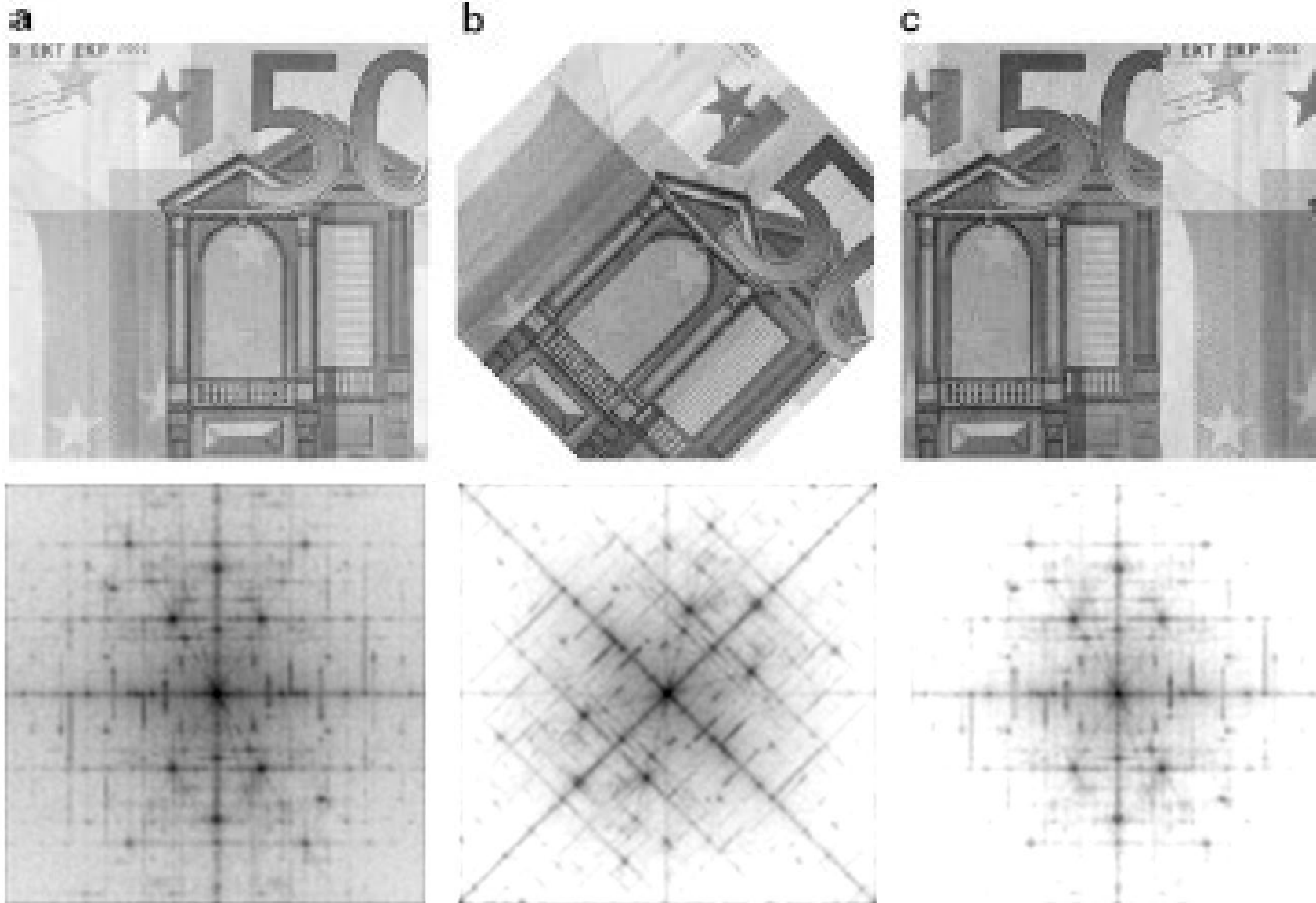
The two dimensional power spectra employ polar coordinates – the frequencies are plotted radially from the center







# Two dimensional Fourier Transform - Properties



- Image rotated

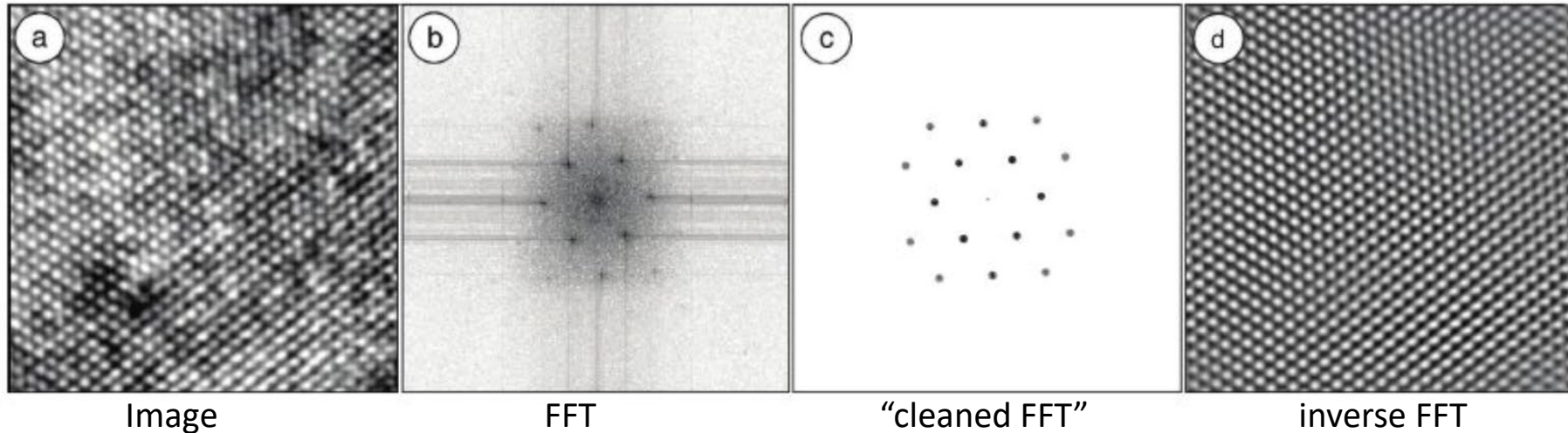
- Image shifted

# Inverse Fourier Transform, Fourier Filtering

The information content of a real image and its Fourier transform is the same – the one defines the other

*As you can calculate the Fourier transform of an image, you can also calculate an image from an FFT*

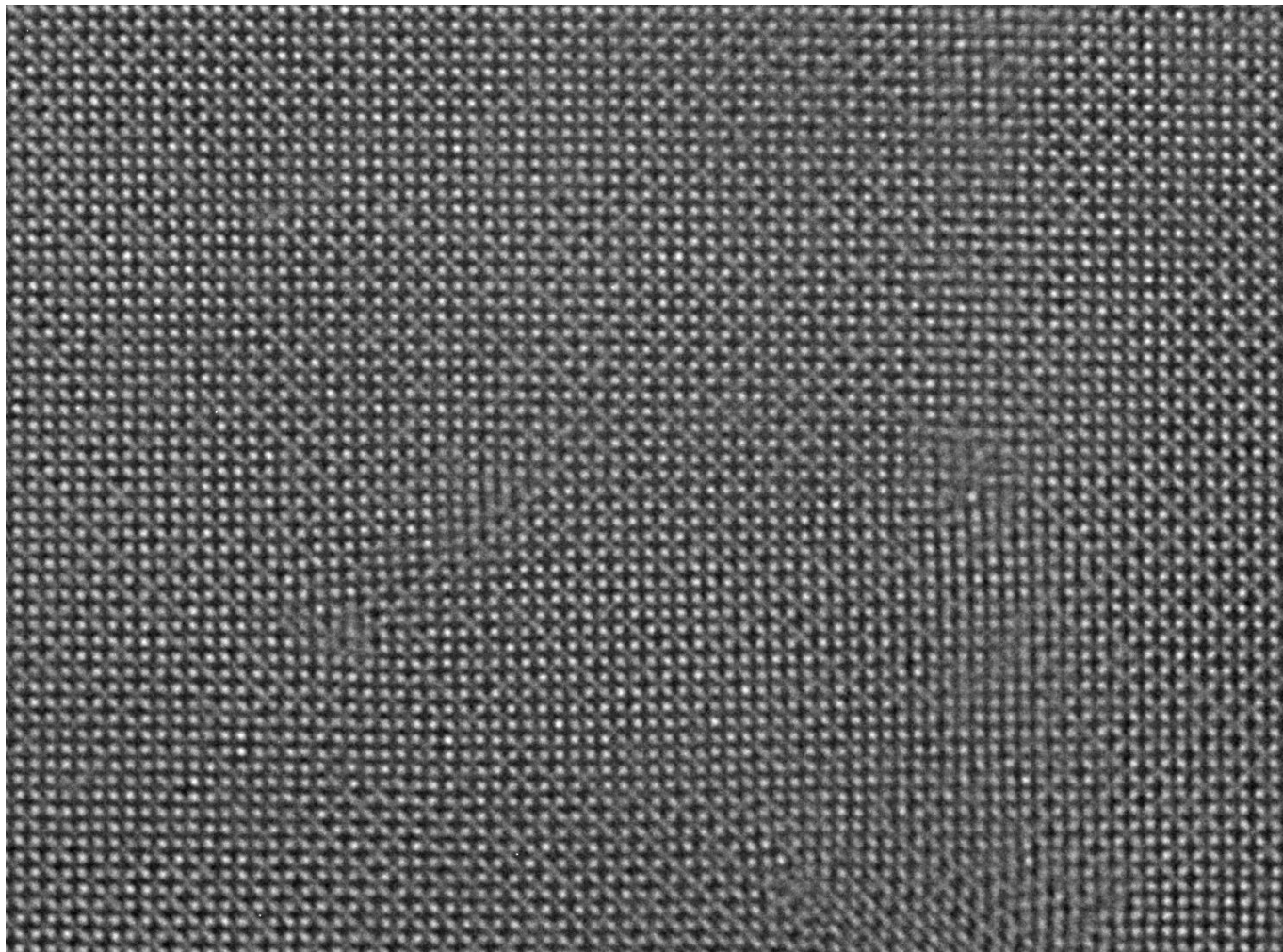
$$f(a, b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) e^{i2\pi(\frac{ka}{N} + \frac{lb}{N})}$$



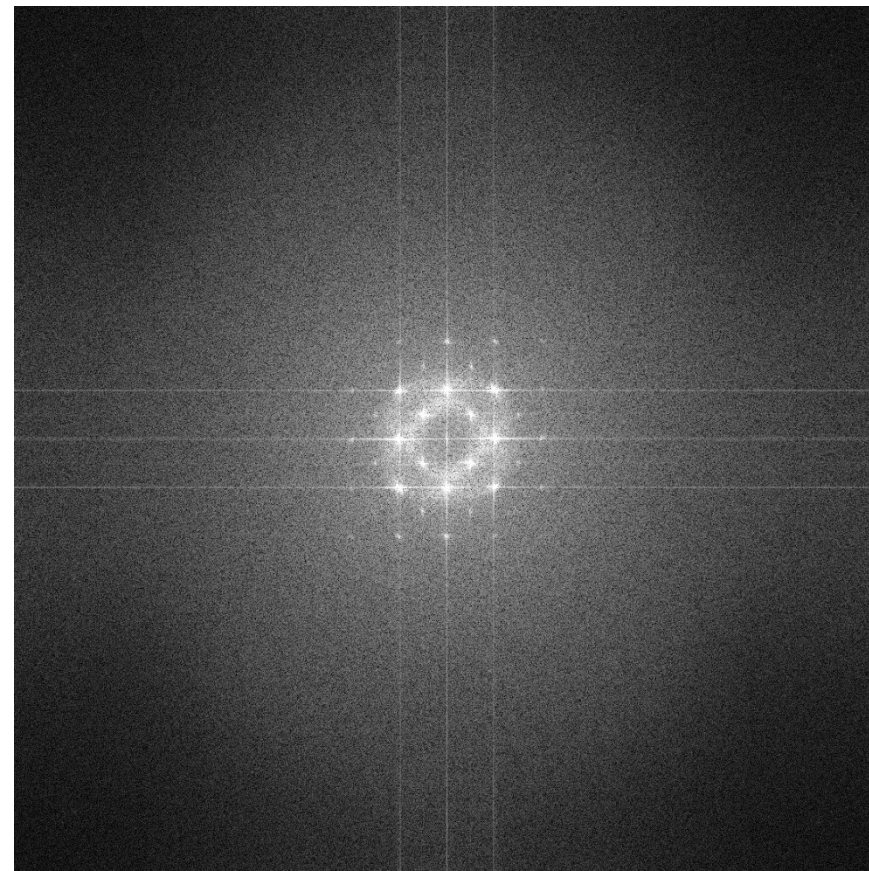
**Figure 6.12** (a) High-resolution TEM image of atomic lattice in silicon (a), with (b) the frequency transform. (c) Isolating just the periodic frequencies and (d) retransforming shows just the atom positions.

*The Fourier domain image has a much greater range (pixel values) than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values.*

# High-Resolution TEM image and its FFT

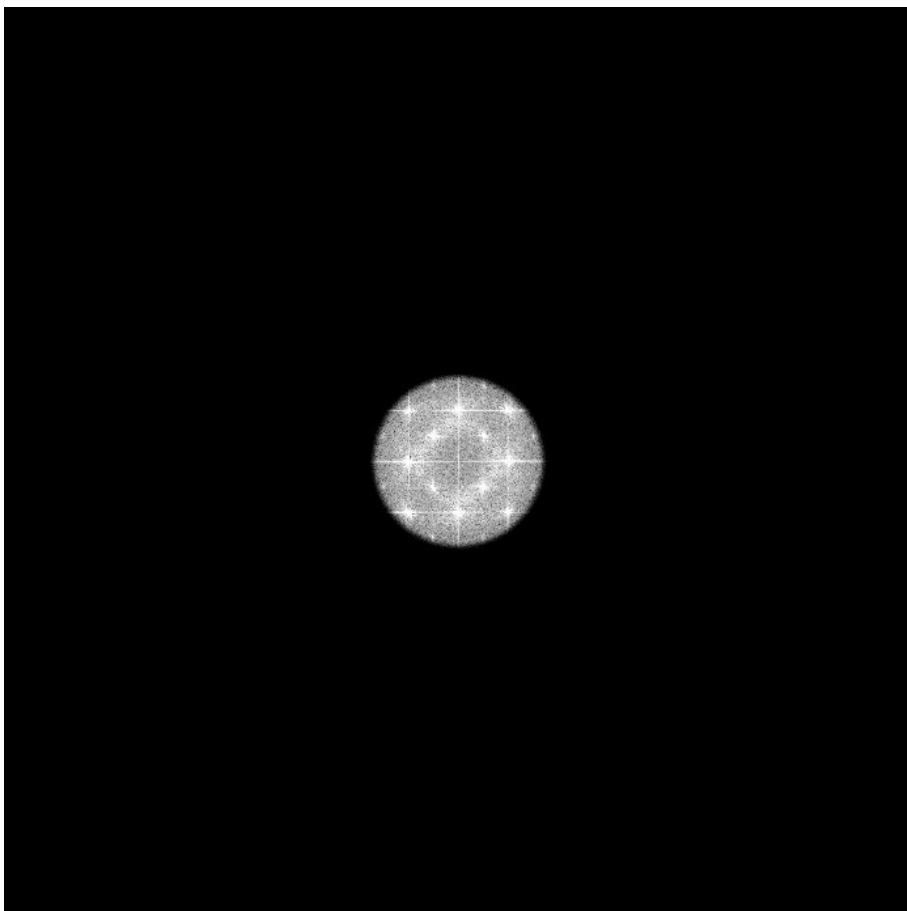


FFT

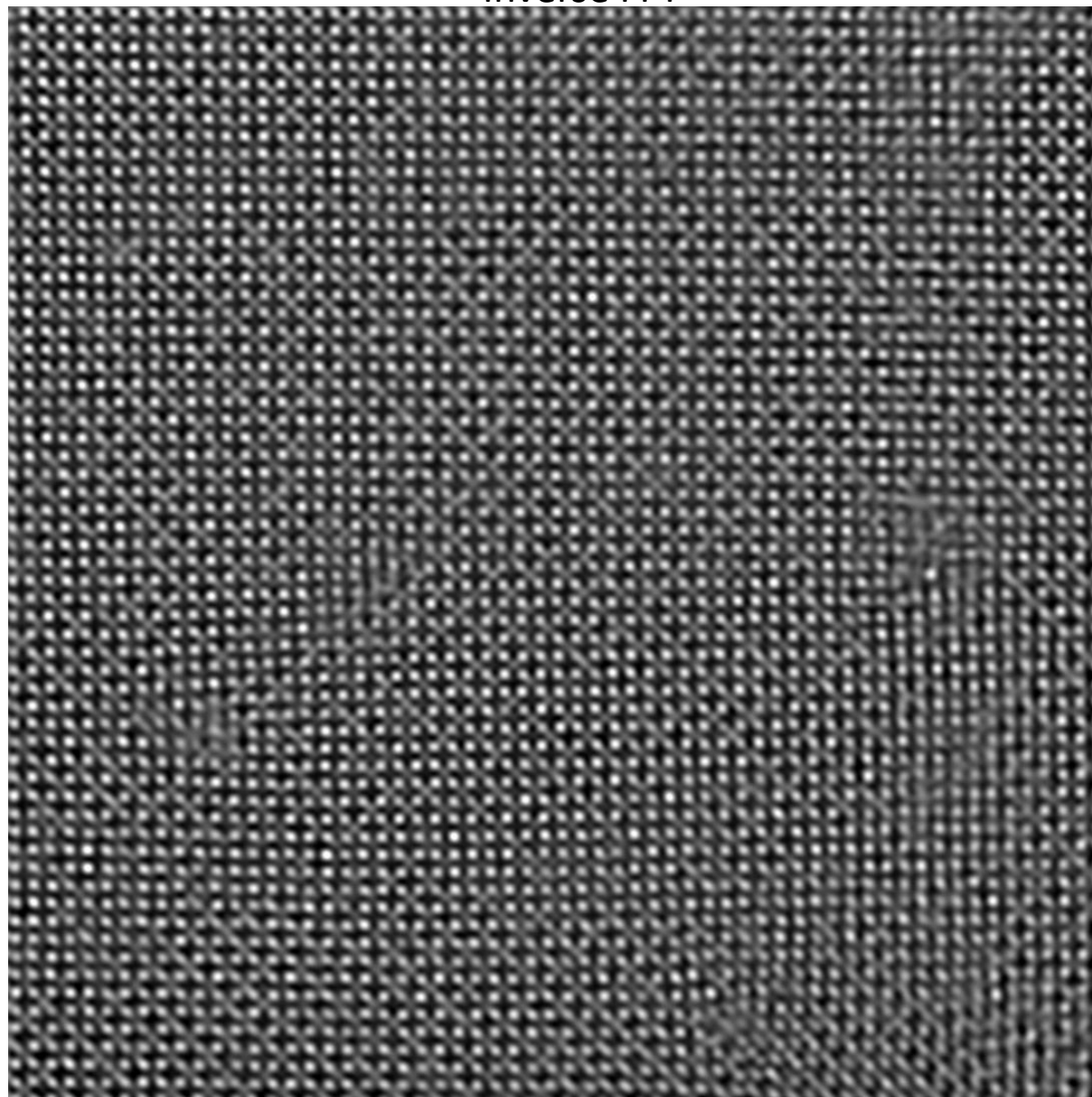




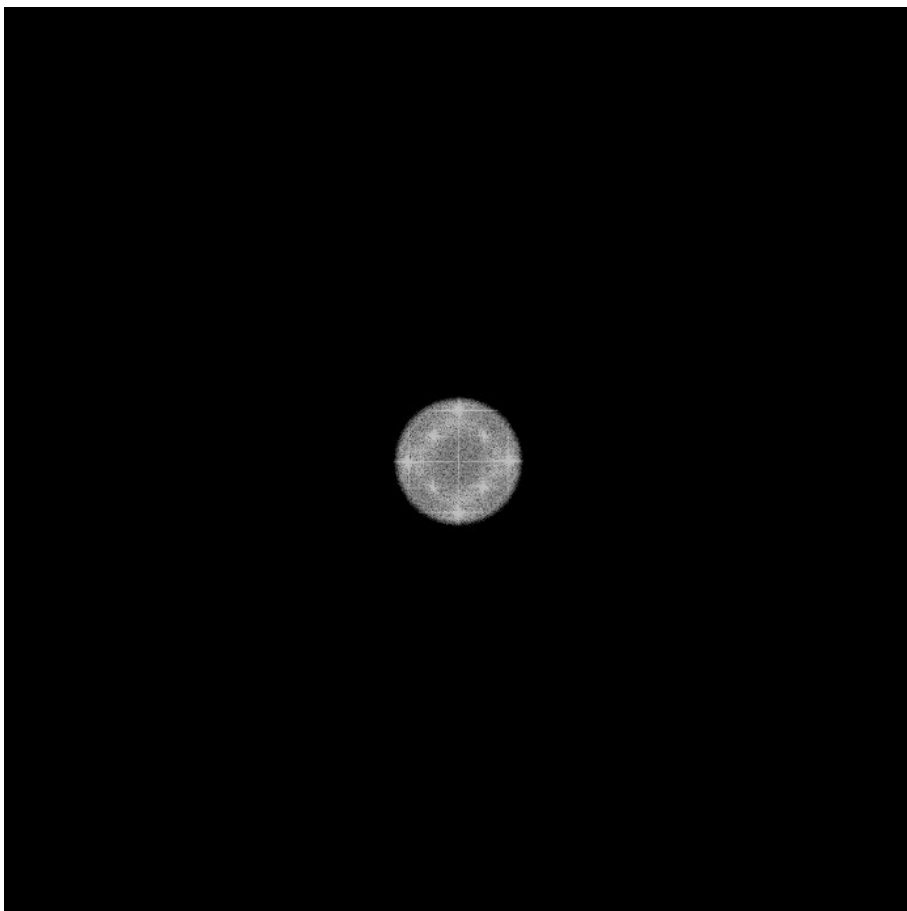
FFT “cleaned” up



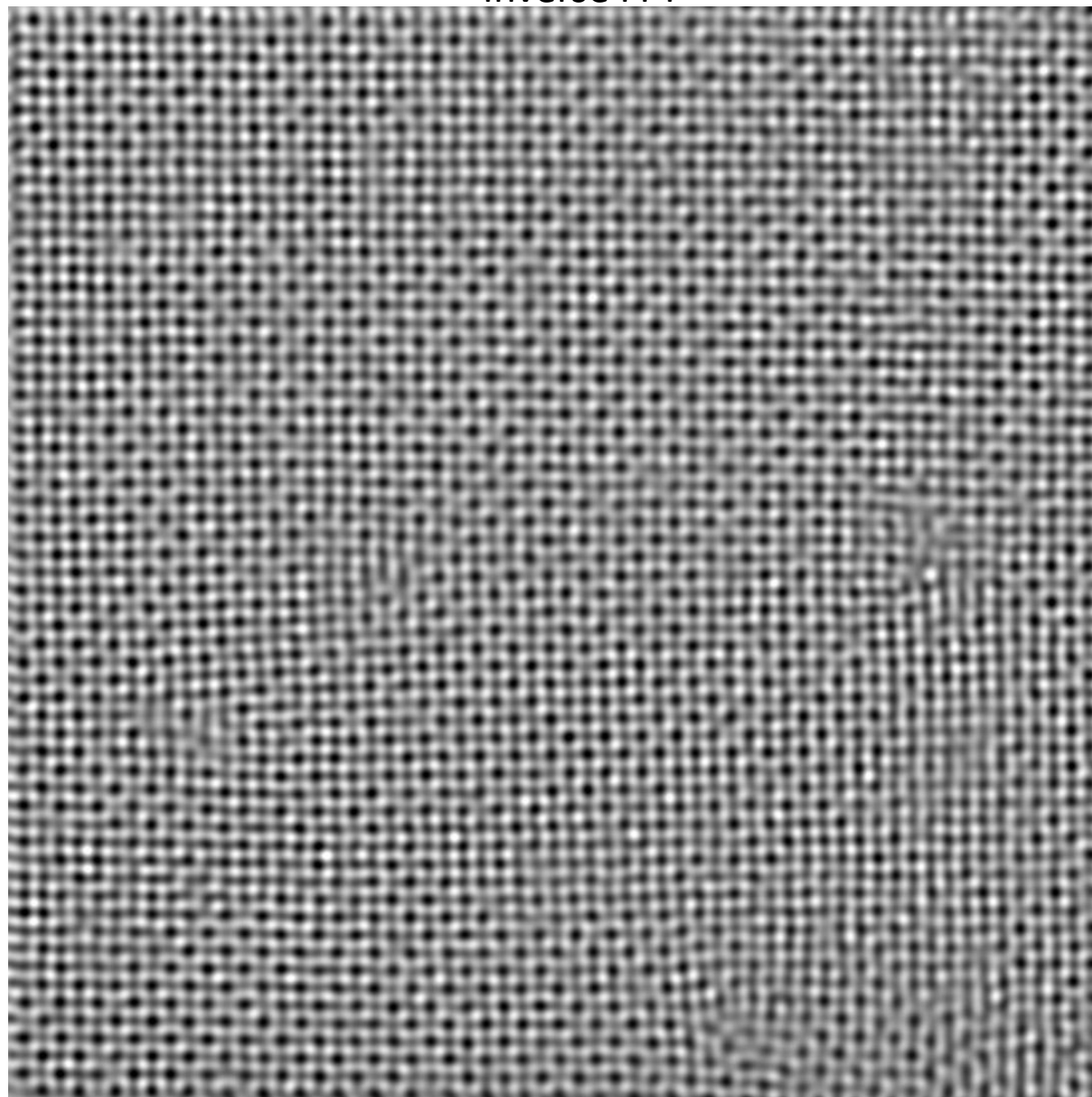
Inverse FFT

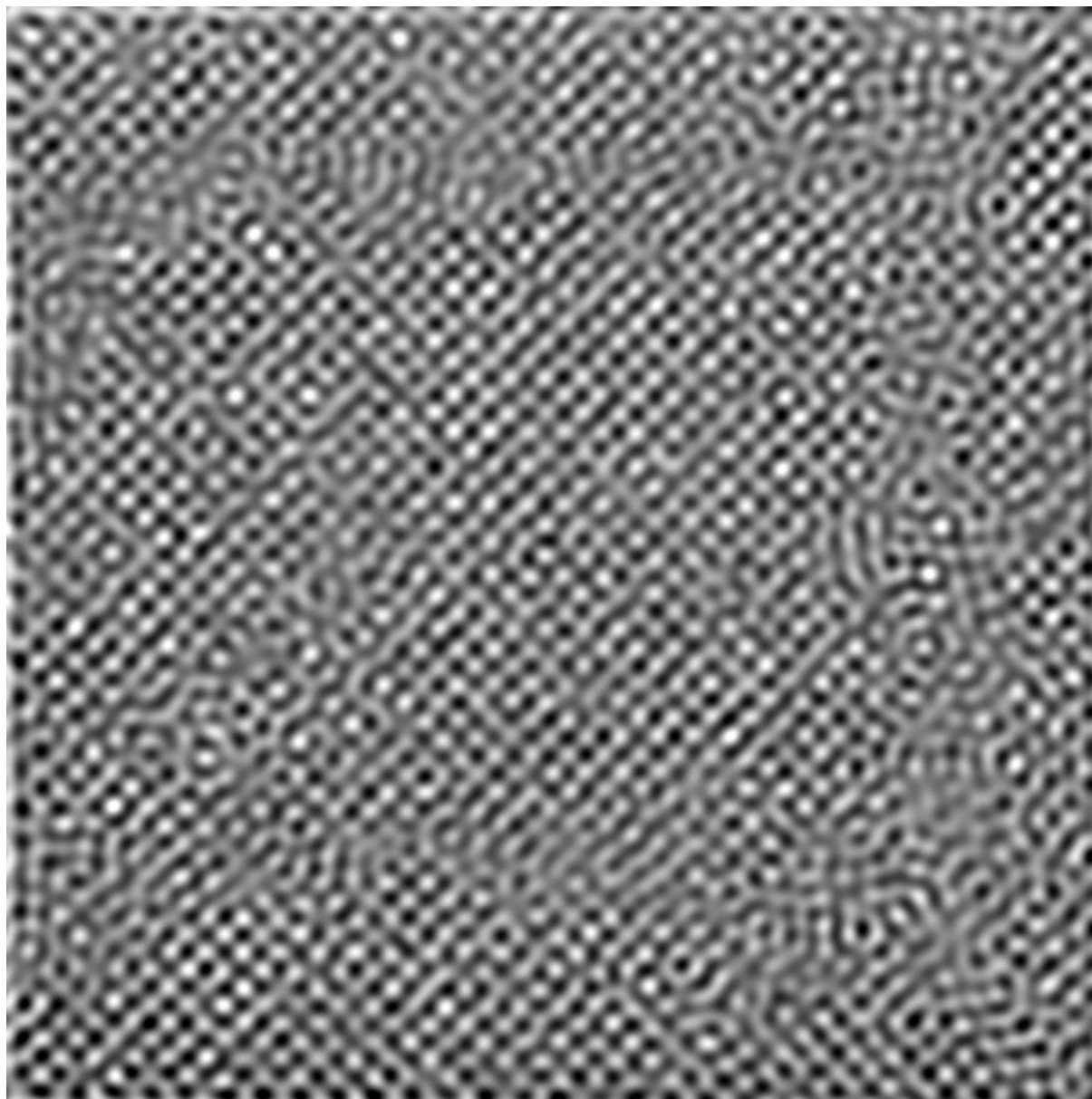
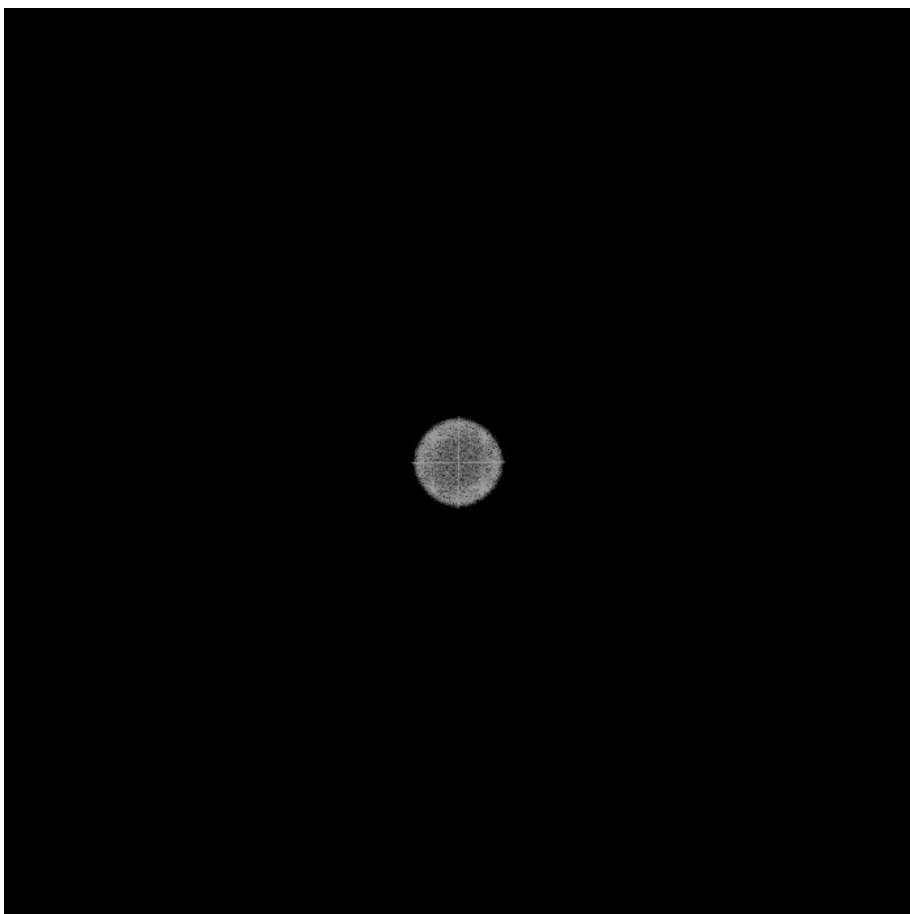


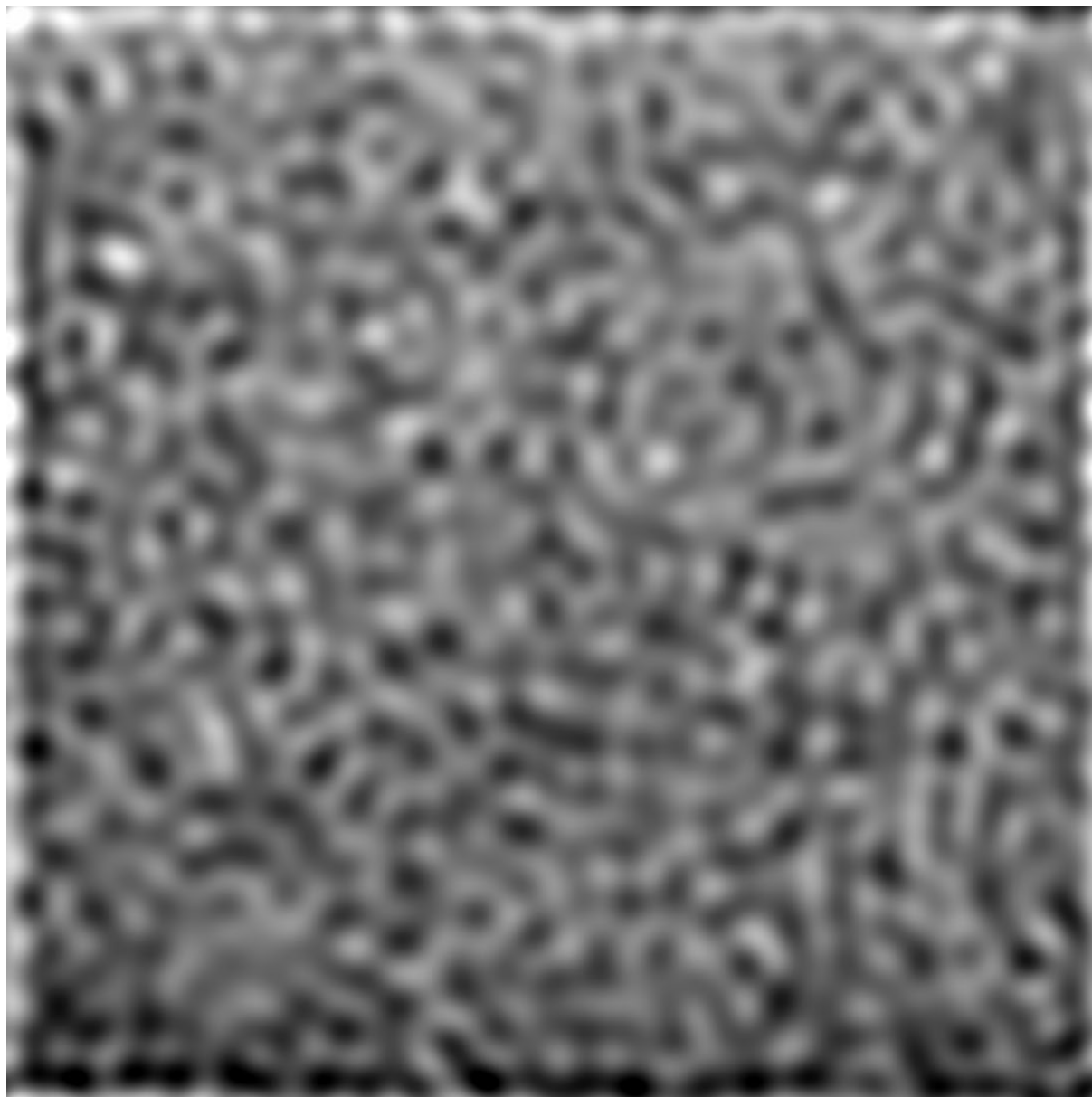
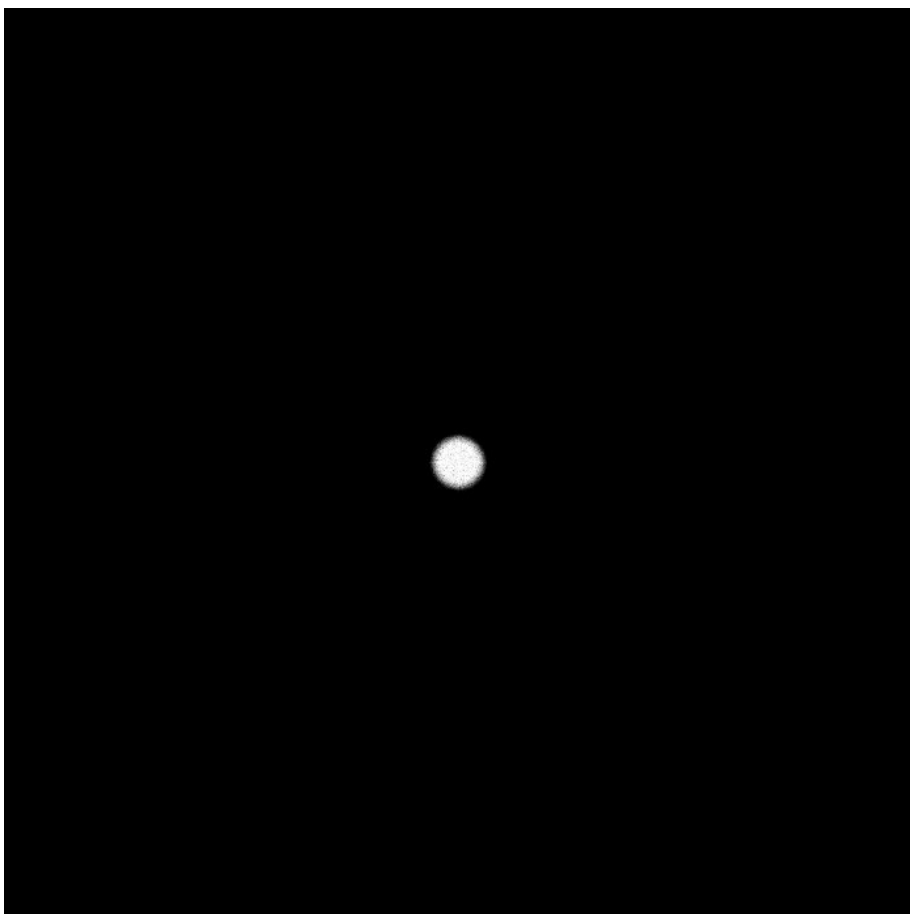
FFT “cleaned” up



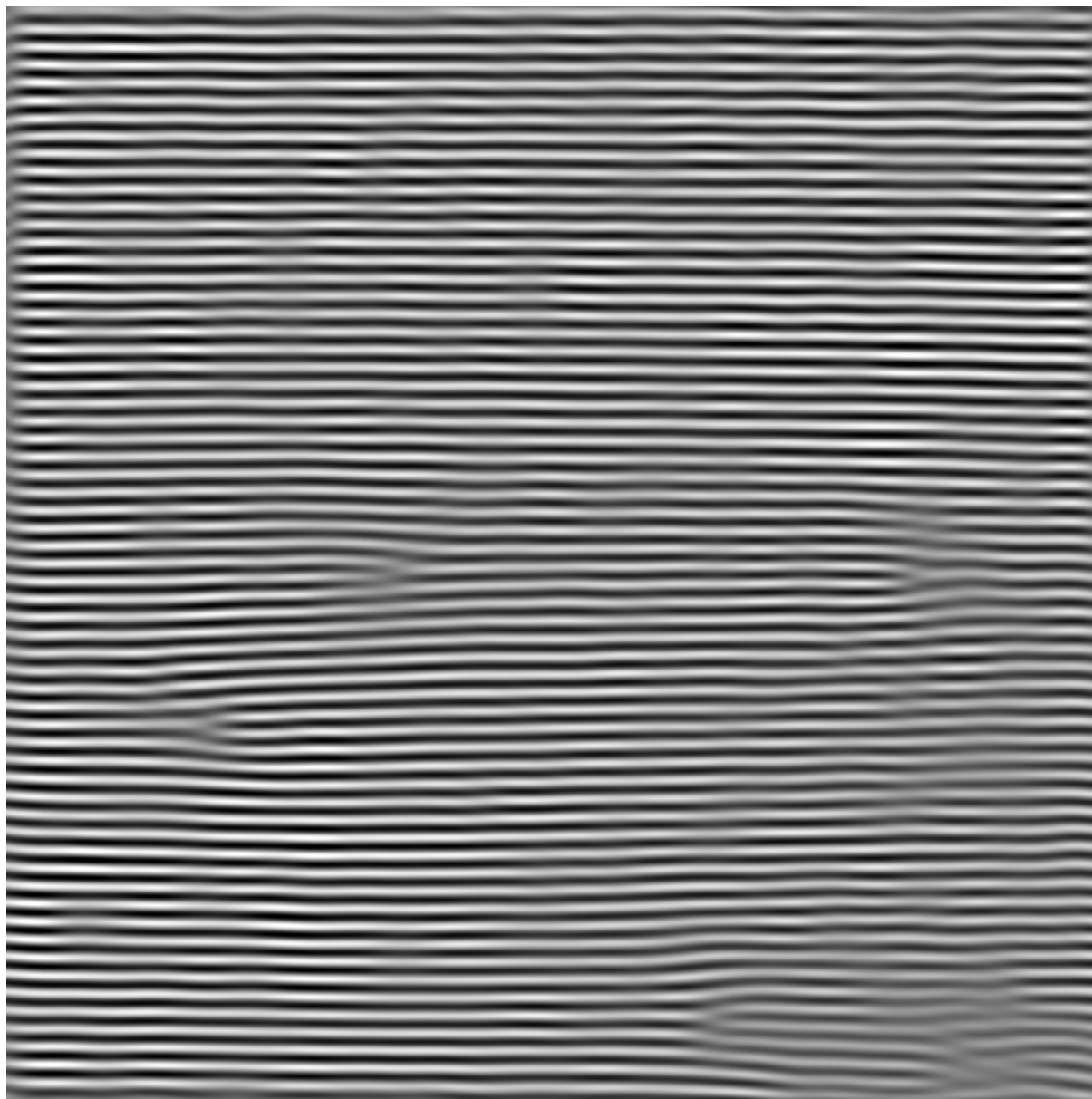
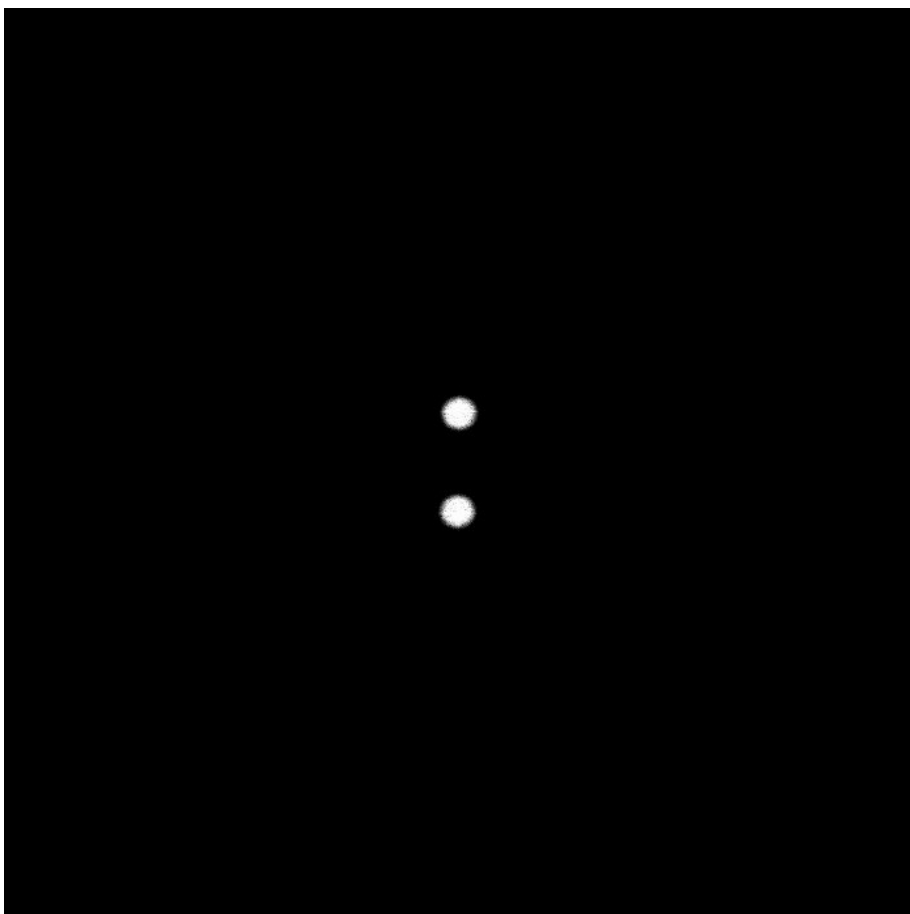
Inverse FFT

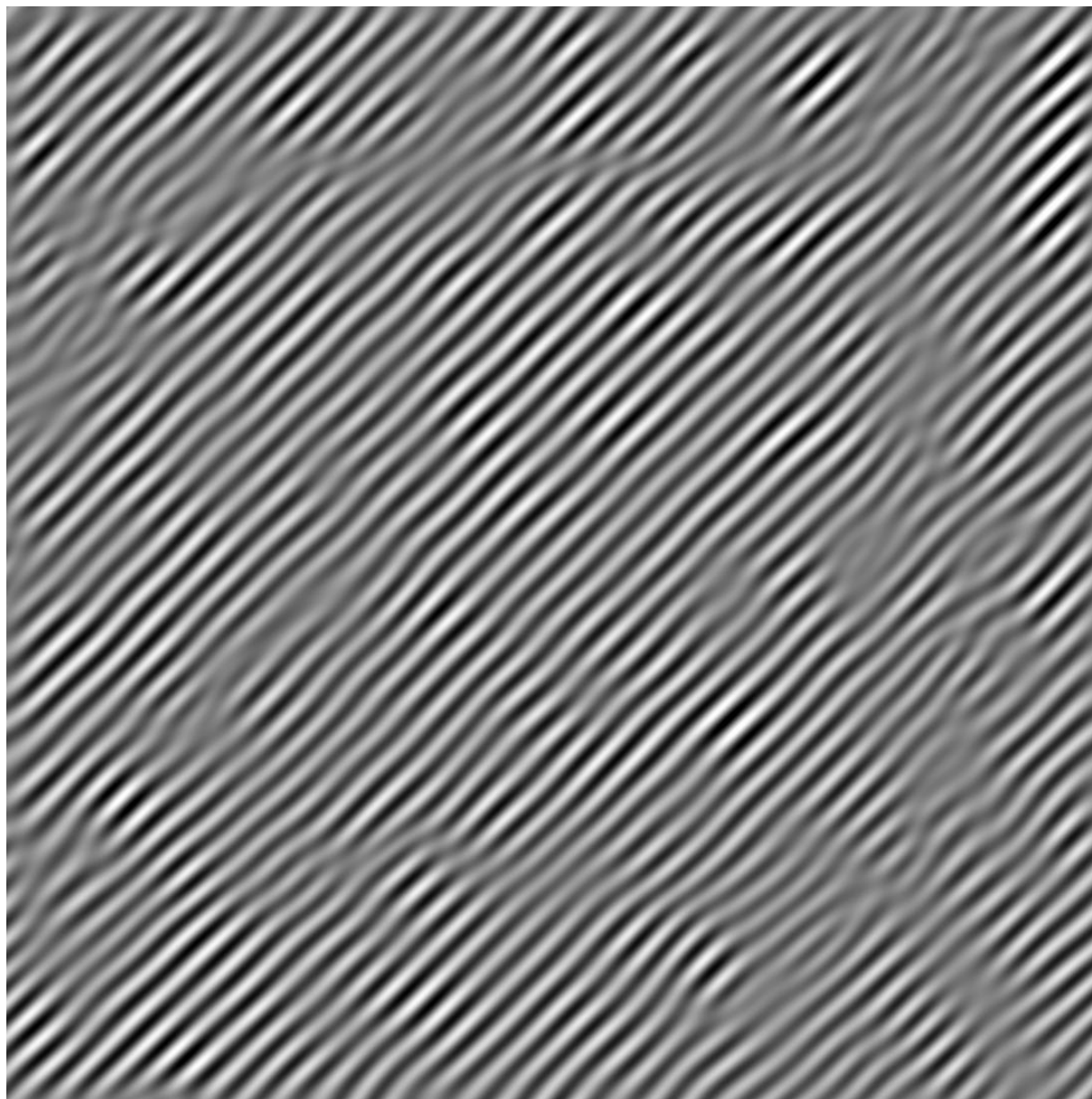
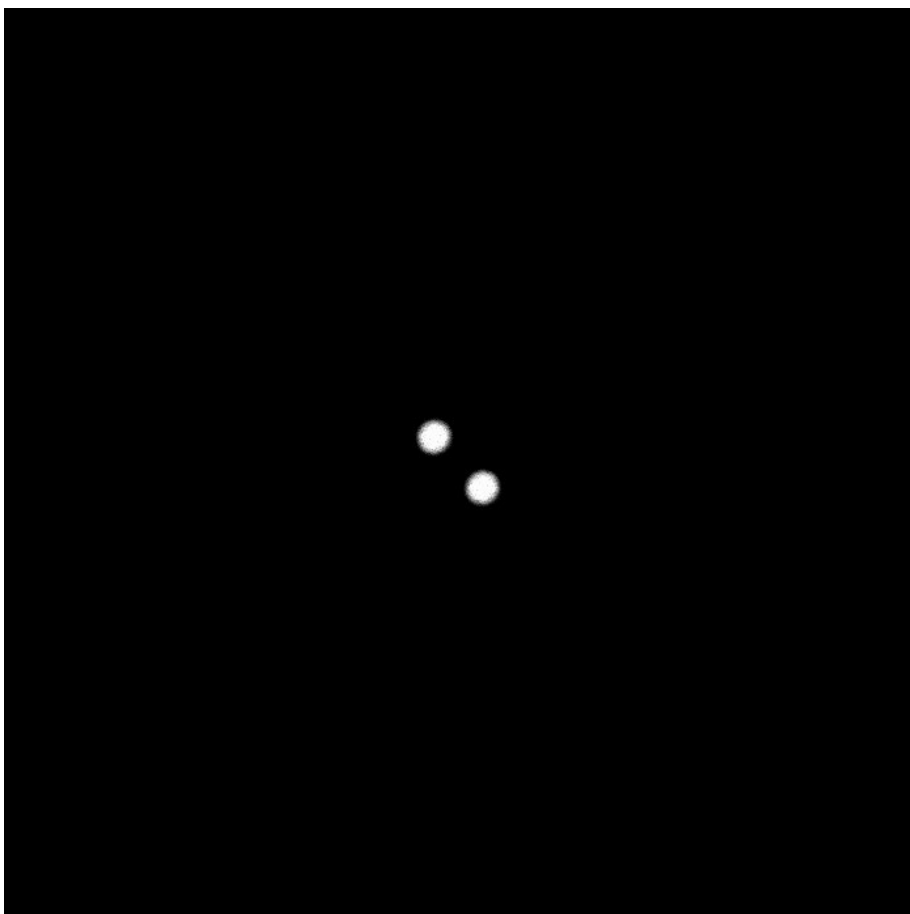












# Frequency Filtering

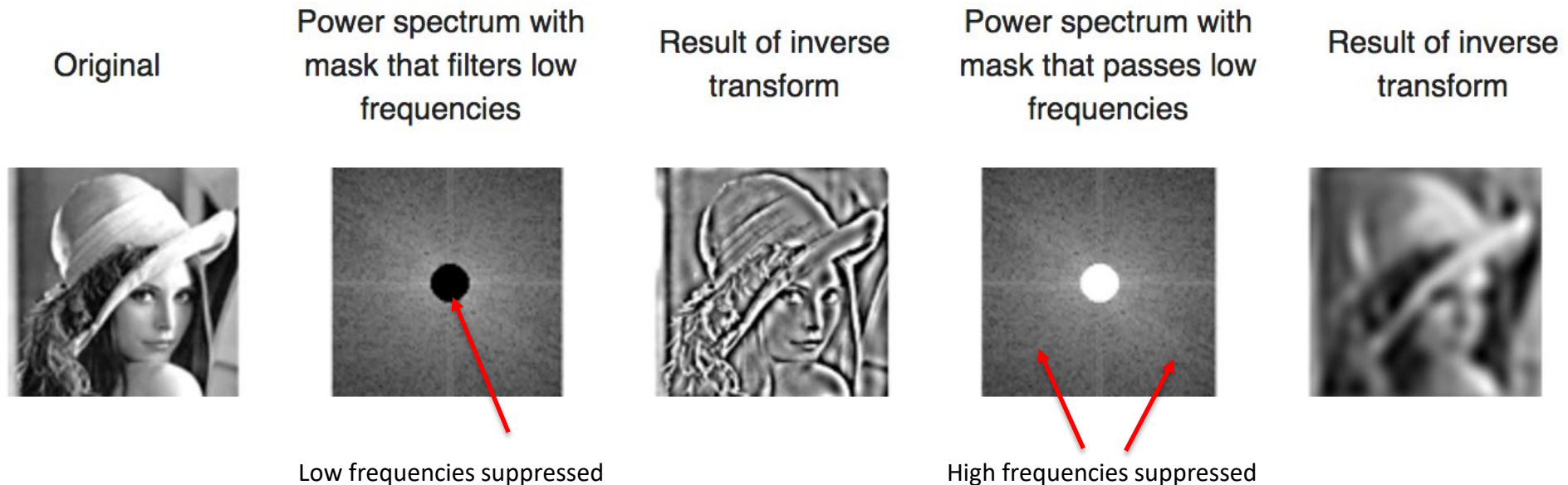
The image is Fourier transformed, multiplied by the filter function and then re-transformed to the spatial domain. Attenuating high frequencies results in a smoother image in the spatial domain, attenuating low frequencies enhances the edges

There are three kinds of filters: *lowpass*, *highpass*, *bandpass* filters

*Lowpass*: attenuates high frequencies and retains low frequencies -> [smoothing filter]

*Highpass*: yields edge enhancement of edge detections because edges contain mainly high frequencies

*Bandpass*: attenuates very low and very high frequencies but retains middle range



# Frequency Filtering

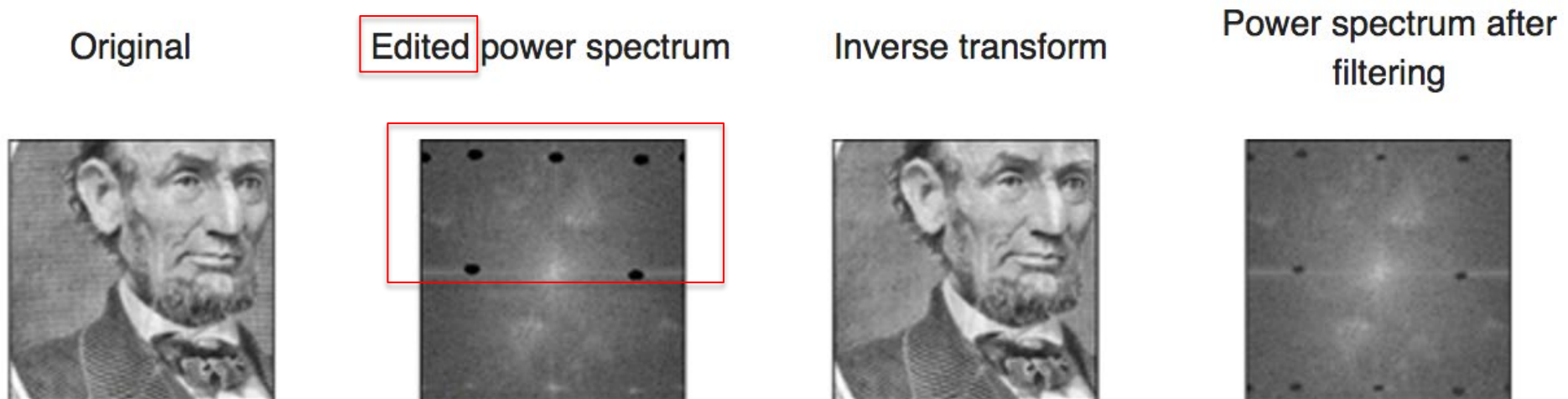
The image is Fourier transformed, multiplied by the filter function and then re-transformed to the spatial domain. Attenuating high frequencies results in a smoother image in the spatial domain, attenuating low frequencies enhances the edges

There are three kinds of filters: *lowpass*, *highpass*, *bandpass* filters

*Lowpass*: attenuates high frequencies and retains low frequencies -> [smoothing filter]

*Highpass*: yields edge enhancement of edge detections because edges contain mainly high frequencies

*Bandpass*: attenuates very low and very high frequencies but retains middle range





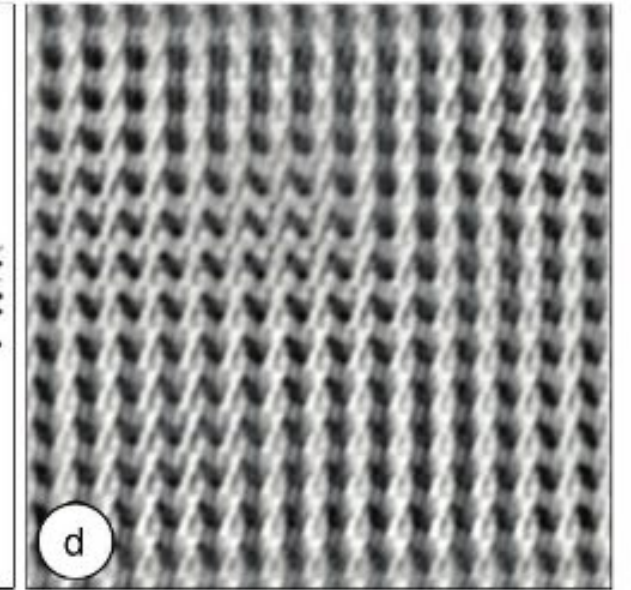
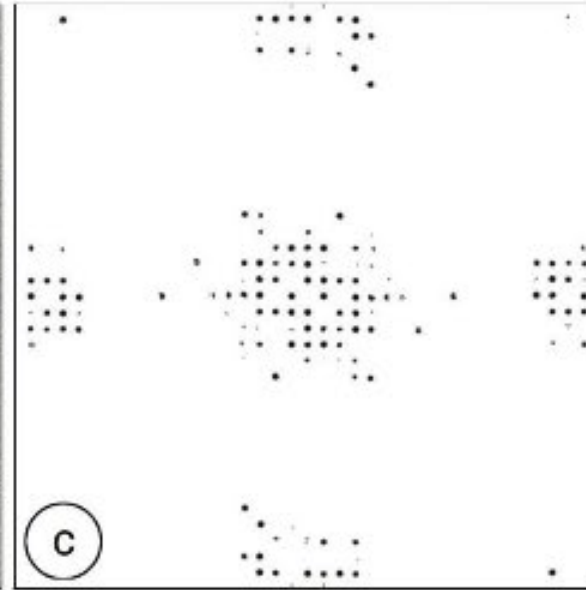
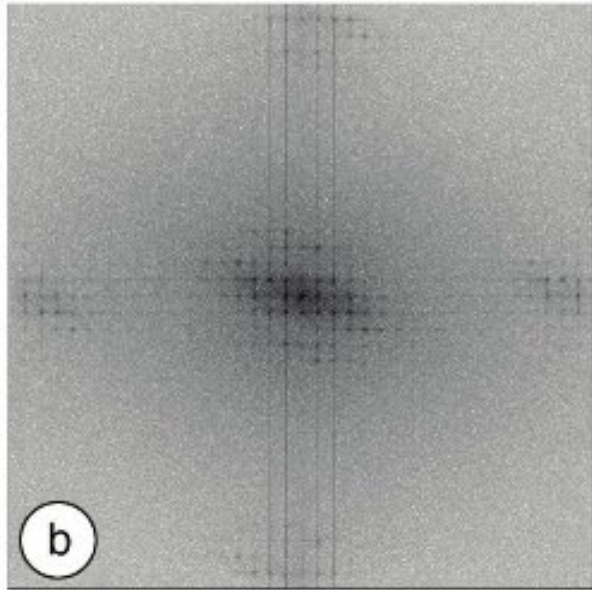
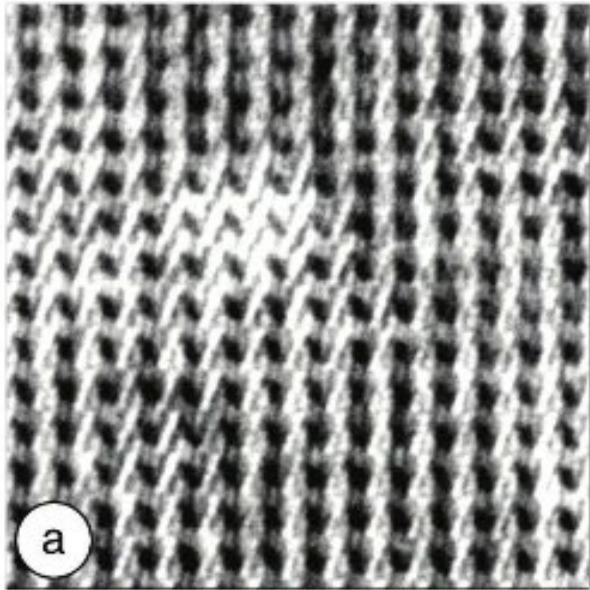
# Selection of Periodic Information – Applying a Mask

original

FFT

filtered to construct  
a mask

inverse FFT with mask

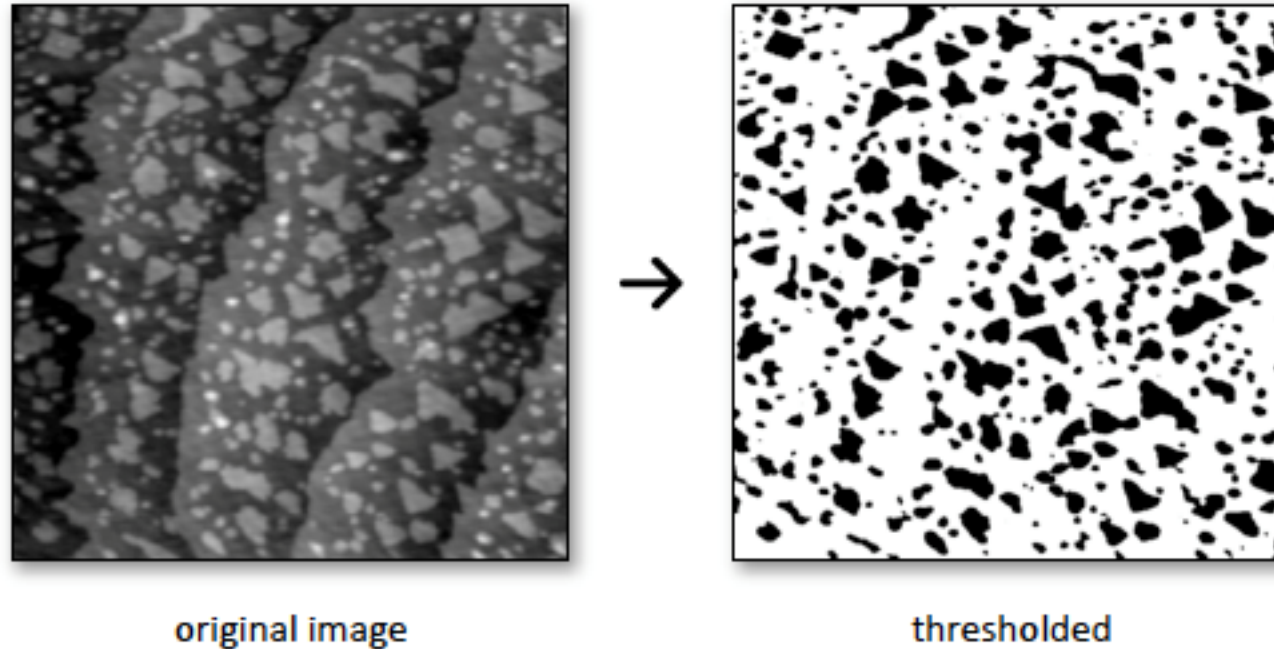


- The Fourier transform has peaks or spikes that represent the periodic structure
- Using a top hat filter with size just large enough to cover the spikes, a mask is constructed to select a small region around each of the periodic spots

# **Segmentation & Thresholding**

# Thresholding concept

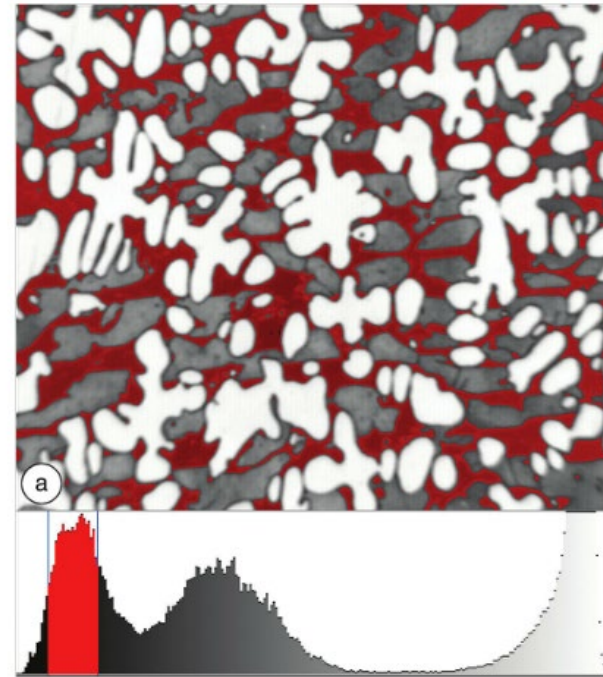
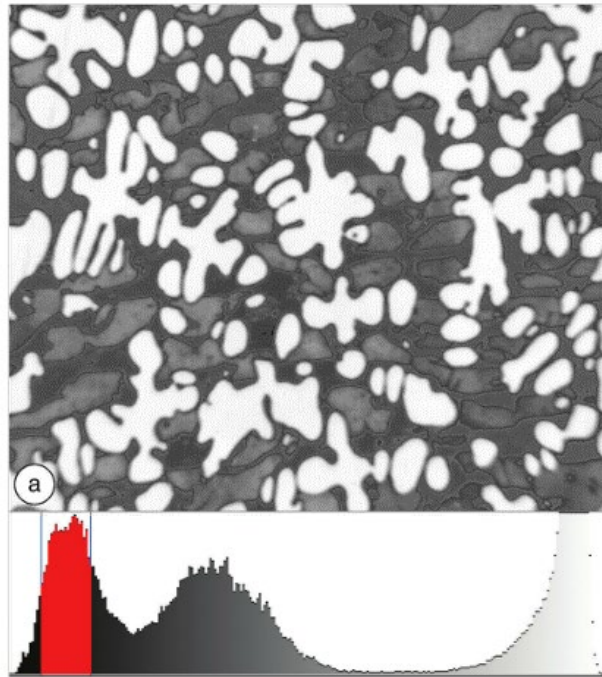
Selecting objects or features within an image by defining a range of brightness values in the original image – this selection process is called thresholding



It may require *pre-processing* for optimal results  
The output is a **binary image** representing the segmentation

# Thresholding basics

Thresholding with the histogram means that a peak indicates that many pixels have nearly the same brightness, and that consequently they may represent the same structure or type of object

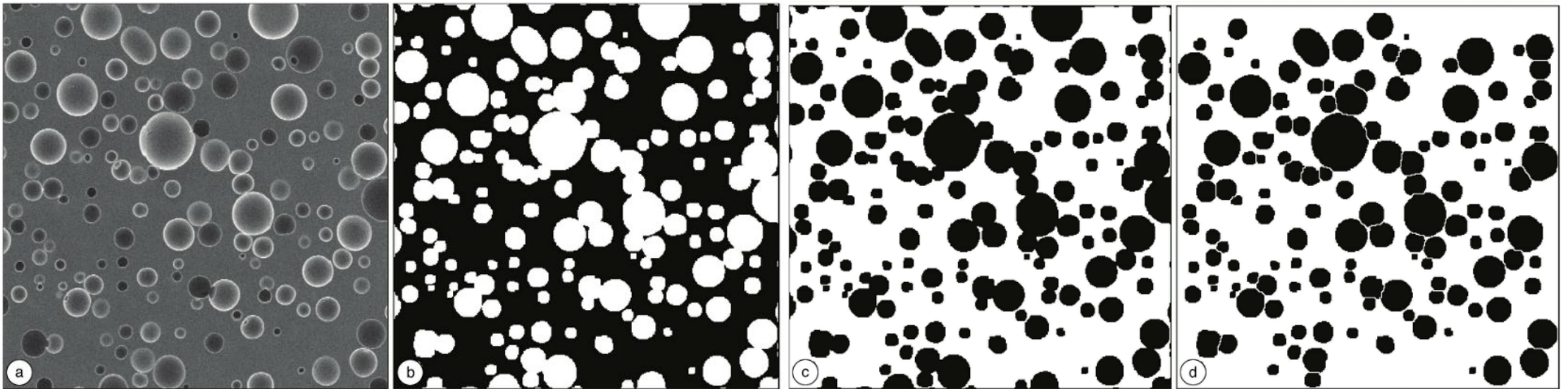


It proceeds with the assumption that a peak in the histogram corresponds to a structure in the image



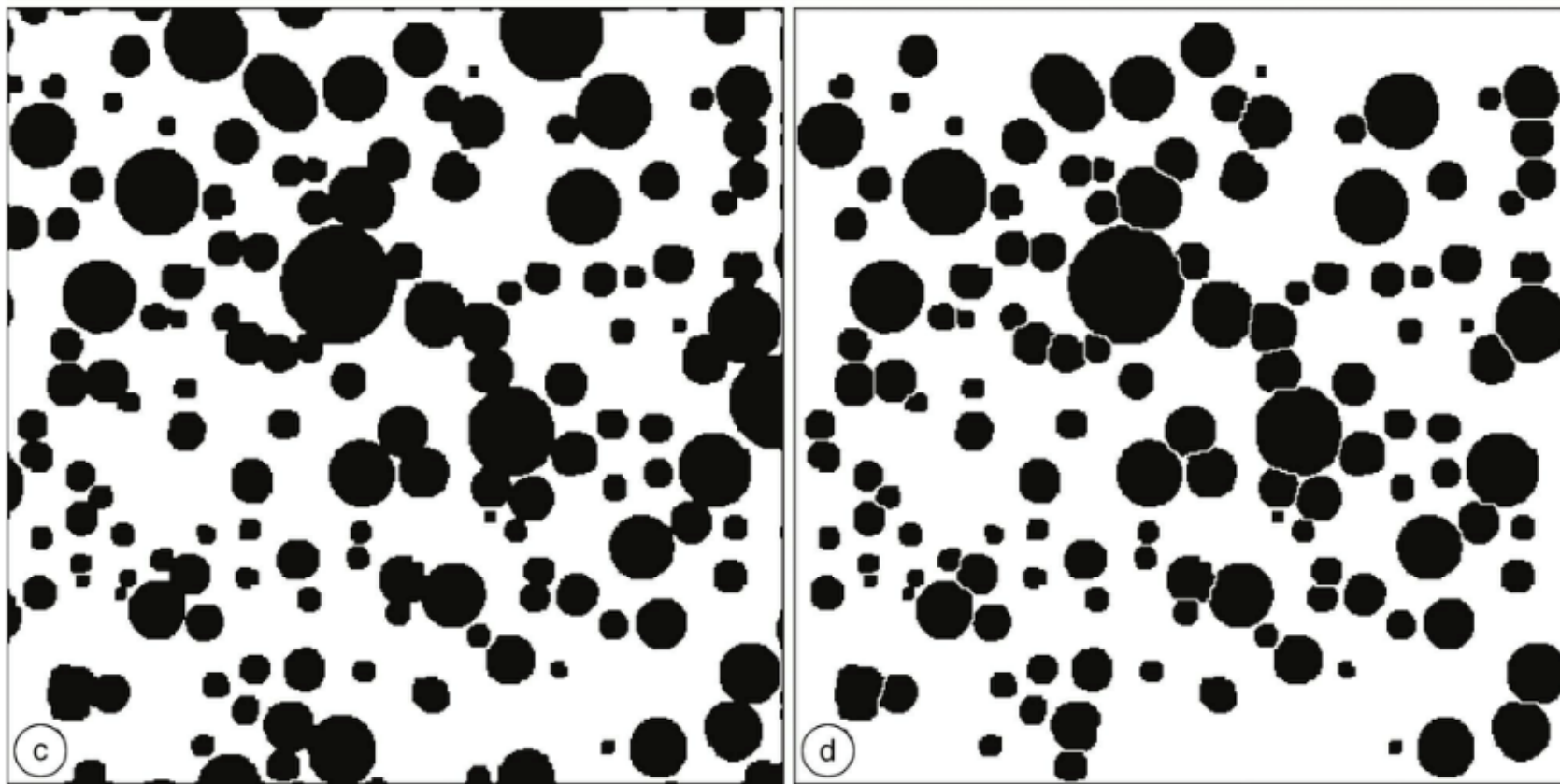
# Thresholding basics

Most of the times, the objects of interest do not share a single brightness or color value, nor a common range of those values. But if the background around the features, or the surface on which the objects lie, is fairly uniform in color or brightness, it may be practical to select that by thresholding.



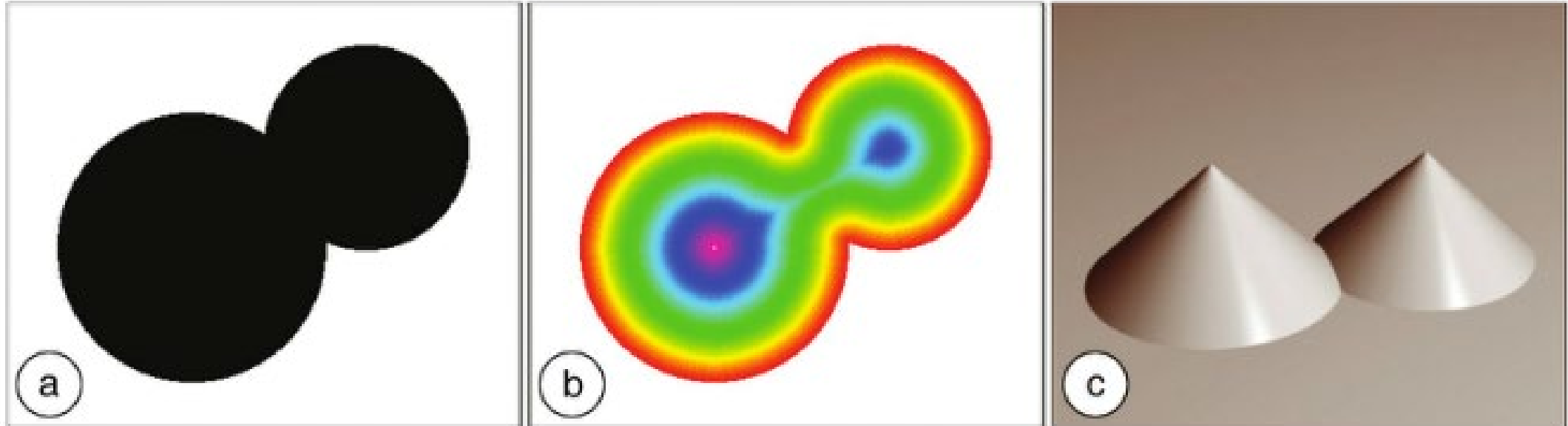
**Figure 7.7** Thresholding the background: (a) image of pores in epoxy resin; (b) thresholded intermediate gray background pixels and eliminating isolated regions that do not connect to the image boundaries; (c) inverted image delineating the pores; (d) a watershed applied to separate the measurable pores.

## Separation of touching particles by “watersheding”



# Watershedding

Watershed segmentation is a way of automatically separating or cutting apart particles that touch. It works best for smoothed objects that don't overlap too much and it applies to binary images only



It proceeds with a complicated algorithm that involves

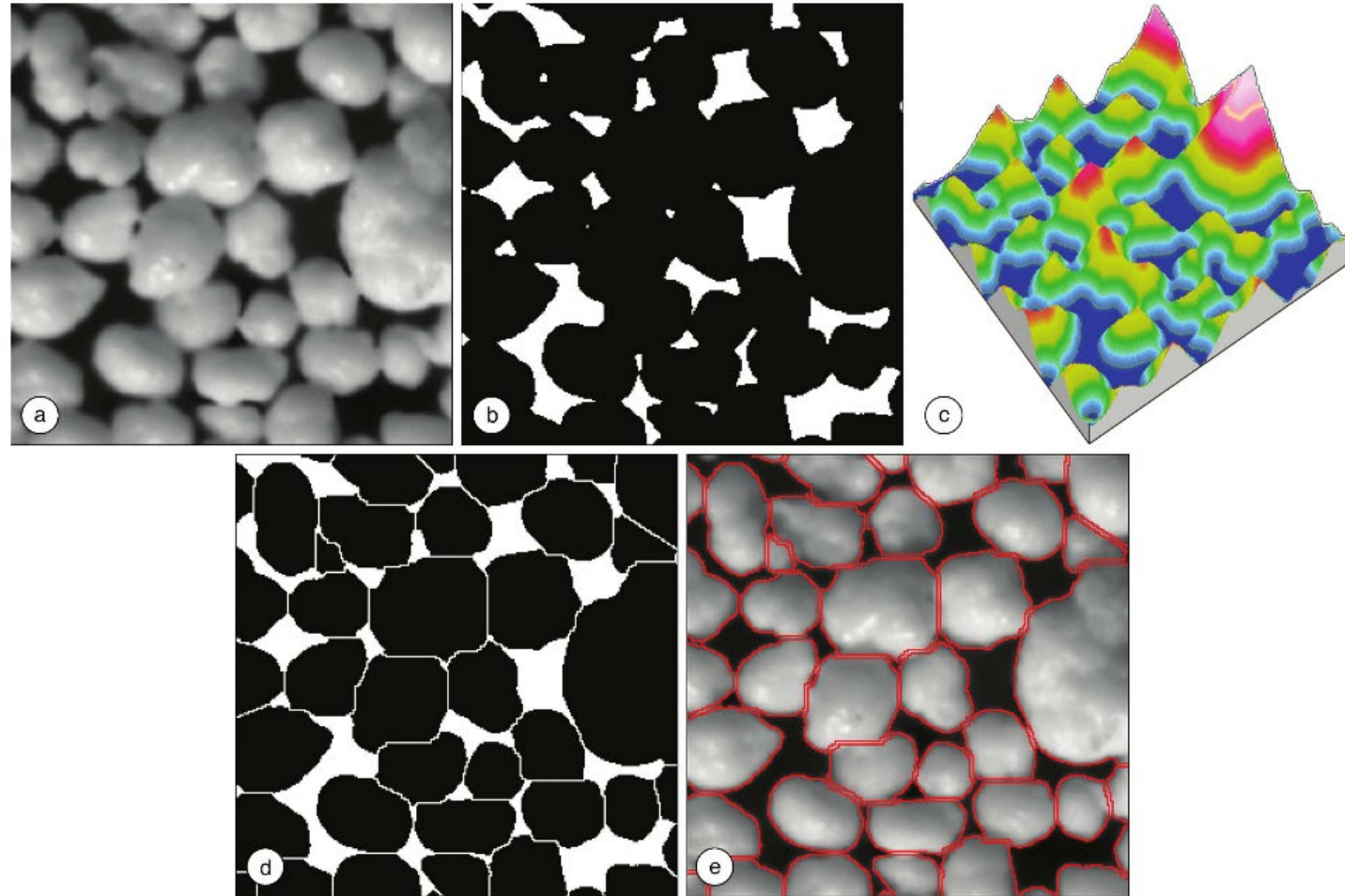
erosion: removes pixels from features in an image by turning pixels OFF -> the ones that are touching

dilation: adds pixels -> can fill holes

**euclidean distance map:** every pixel is assigned a value that is its distance from the nearest background pixel

# Watershed

The assumption is that the features are sufficiently convex, i.e. the euclidean distance map has only one peak in each feature

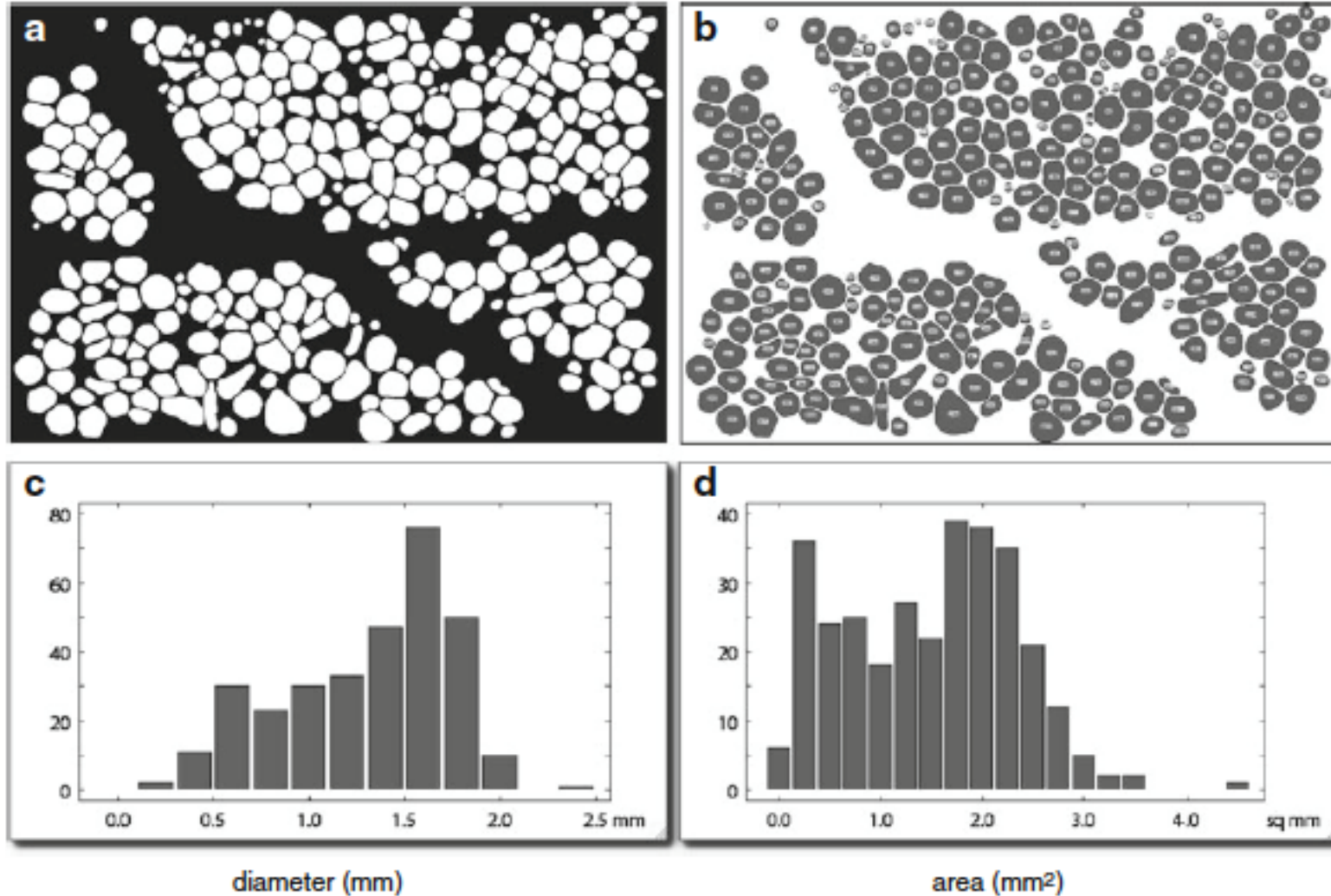




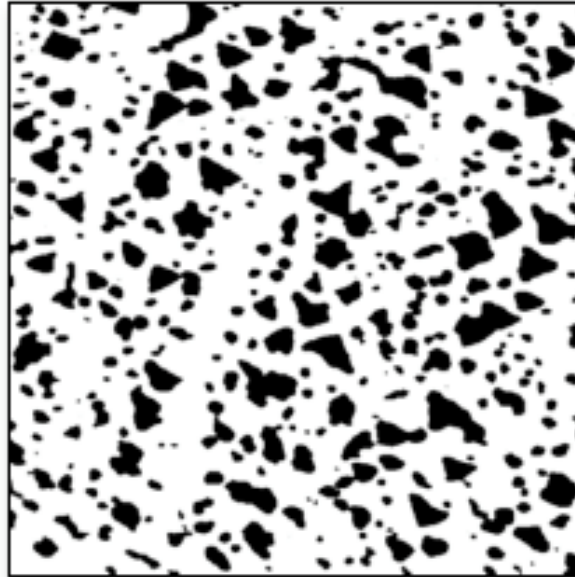
# **Image Measurements**

# Grain Size -Example

Oolitic limestone sample from the Jura mountains in Switzerland –  
ImageJ -> Analyse particles/Distribution



# Particle Analysis (example in Fiji/ImageJ)

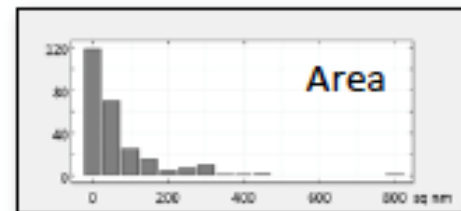
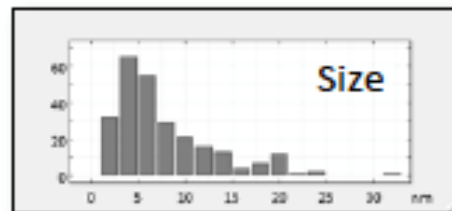


|     | Area   | Mean | StDv | X      | Y      | Len    | Majr  | Minr  | Angle | Min  | Max  |
|-----|--------|------|------|--------|--------|--------|-------|-------|-------|------|------|
| 1.  | 328.38 | 0.78 | 0.07 | 109.15 | 329.47 | 105.67 | 34.59 | 11.94 | 43.0  | 0.59 | 0.91 |
| 2.  | 6.86   | 0.63 | 0.02 | 135.40 | 341.15 | 11.54  | 4.65  | 1.88  | 9.0   | 0.59 | 0.65 |
| 3.  | 5.39   | 0.59 | 0.01 | 294.51 | 341.03 | 8.74   | 3.33  | 2.06  | 177.5 | 0.59 | 0.60 |
| 4.  | 8.90   | 0.61 | 0.02 | 340.26 | 339.57 | 12.36  | 5.79  | 1.08  | 94.9  | 0.59 | 0.65 |
| 5.  | 1.47   | 0.59 | 0.00 | 43.40  | 340.90 | 4.37   | 2.37  | 0.79  | 0.0   | 0.59 | 0.59 |
| 6.  | 10.29  | 0.62 | 0.03 | 129.07 | 339.20 | 11.13  | 4.40  | 2.88  | 172.8 | 0.59 | 0.67 |
| 7.  | 24.50  | 0.67 | 0.05 | 298.94 | 335.79 | 17.89  | 6.72  | 4.64  | 170.9 | 0.59 | 0.77 |
| 8.  | 5.88   | 0.65 | 0.04 | 172.26 | 336.17 | 8.74   | 3.63  | 2.06  | 175.2 | 0.59 | 0.70 |
| 9.  | 48.02  | 0.70 | 0.08 | 312.58 | 332.72 | 27.79  | 9.18  | 6.66  | 54.0  | 0.59 | 0.87 |
| 10. | 40.67  | 0.64 | 0.03 | 28.80  | 332.96 | 24.07  | 8.05  | 6.43  | 5.1   | 0.59 | 0.68 |
| 11. | 34.30  | 0.73 | 0.09 | 279.41 | 332.26 | 21.44  | 7.83  | 5.57  | 20.0  | 0.59 | 0.89 |
| 12. | 6.37   | 0.61 | 0.01 | 252.02 | 333.85 | 9.73   | 4.17  | 1.95  | 11.0  | 0.59 | 0.63 |
| 13. | 1.96   | 0.60 | 0.01 | 14.70  | 333.02 | 4.95   | 1.91  | 1.30  | 0.0   | 0.59 | 0.61 |
| 14. | 5.39   | 0.63 | 0.03 | 264.09 | 331.23 | 8.33   | 3.33  | 2.06  | 2.5   | 0.59 | 0.67 |
| 15. | 21.56  | 0.65 | 0.04 | 298.71 | 329.05 | 18.08  | 5.95  | 4.61  | 149.8 | 0.59 | 0.73 |
| 16. | 5.39   | 0.60 | 0.01 | 235.04 | 329.57 | 8.33   | 3.49  | 1.97  | 11.5  | 0.59 | 0.62 |
| 17. | 56.04  | 0.73 | 0.12 | 335.38 | 321.78 | 46.50  | 15.97 | 7.66  | 138.2 | 0.59 | 1.00 |
| 18. | 44.59  | 0.65 | 0.03 | 223.54 | 326.75 | 24.82  | 9.36  | 6.06  | 175.6 | 0.59 | 0.70 |
| 19. | 19.60  | 0.61 | 0.02 | 144.71 | 325.83 | 18.88  | 7.72  | 3.23  | 169.3 | 0.59 | 0.64 |
| --- | ---    | ---  | ---  | ---    | ---    | ---    | ---   | ---   | ---   | ---  | ---  |
| --- | ---    | ---  | ---  | ---    | ---    | ---    | ---   | ---   | ---   | ---  | ---  |
| --- | ---    | ---  | ---  | ---    | ---    | ---    | ---   | ---   | ---   | ---  | ---  |



Mean nearest neighbour distance =  $13 \pm 5$  nm

Nearest neighbour lies in azimuthal direction  $83^\circ$  (anisotropy = 0.19)



# Estimating Size

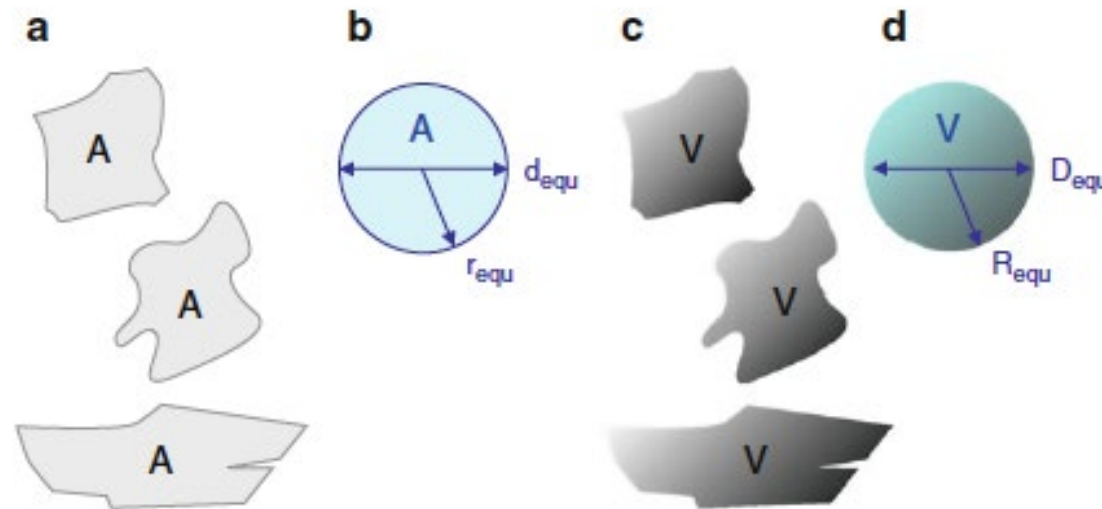
## Estimating area

Counting pixels: The area of a particle A is given by the number of pixels in that phase and the area density is obtained by relating the number of pixels of the particles to the total number of pixels -> Measure command in ImageJ

**Estimating volume  $V_\alpha$  of phase  $\alpha$**   
It corresponds to the area fraction

$$V_\alpha = \int A_\alpha(z) dz$$

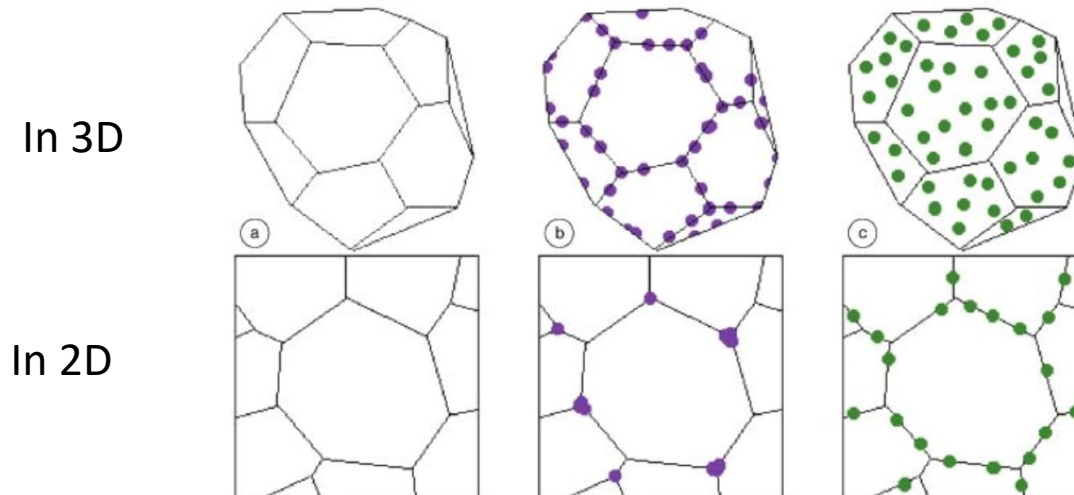
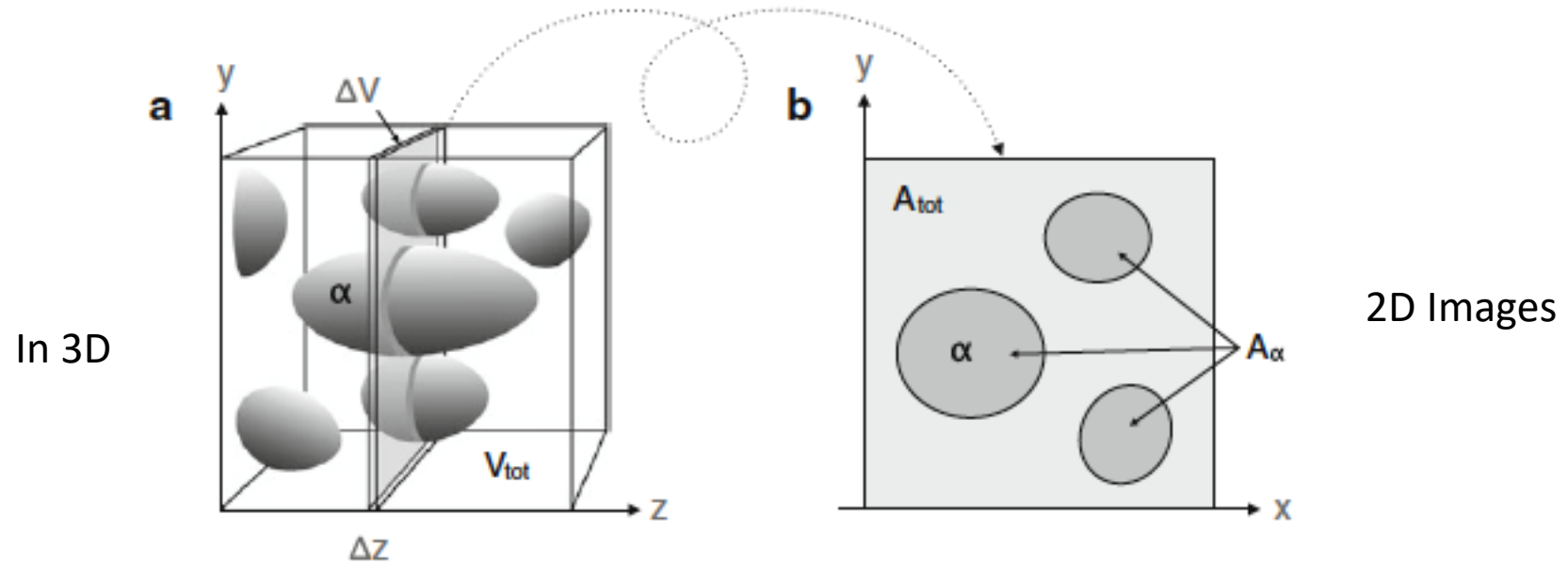
$$\frac{\bar{A}_\alpha}{A_{\text{tot}}} = \frac{V_\alpha}{V_{\text{tot}}}$$



The best estimate is by using the cross-sectional area and the concept of area-equivalent circle/eclipse



# Particle Analysis – How it works



# Software Practice: ImageJ [or Fiji]

# IMAGEJ

An open platform for scientific image analysis

<http://imagej.net/ImageJ2>



<https://imagej.net/Fiji/Downloads>