Training School # 3

Restoration of BioImage
by
Digital Filters

February 12th-16th, 2017

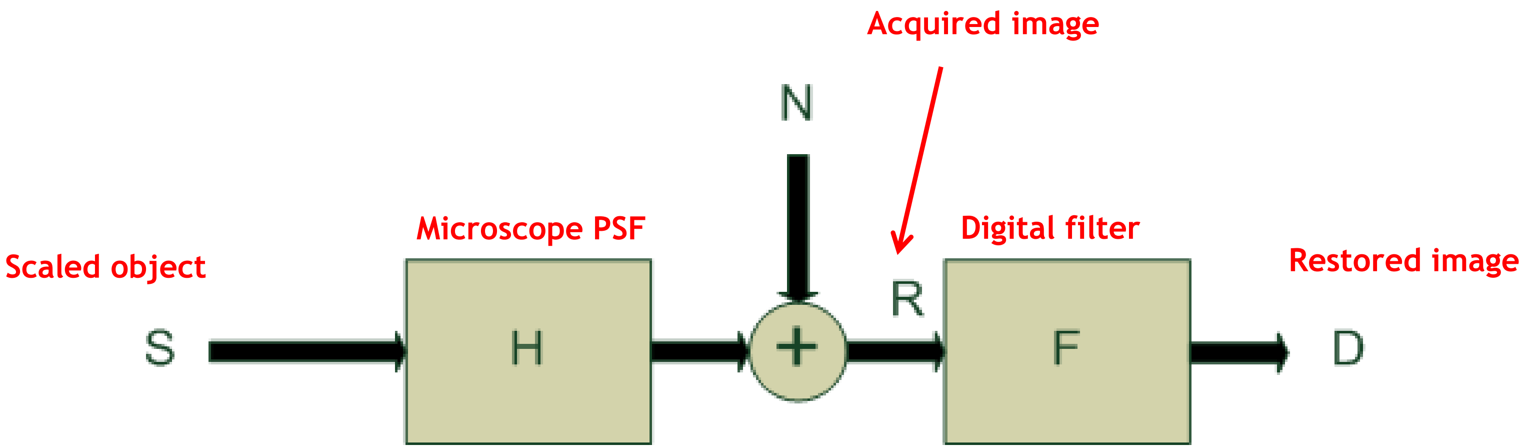
Gulbenkian Institute

Oeiras, Portugal

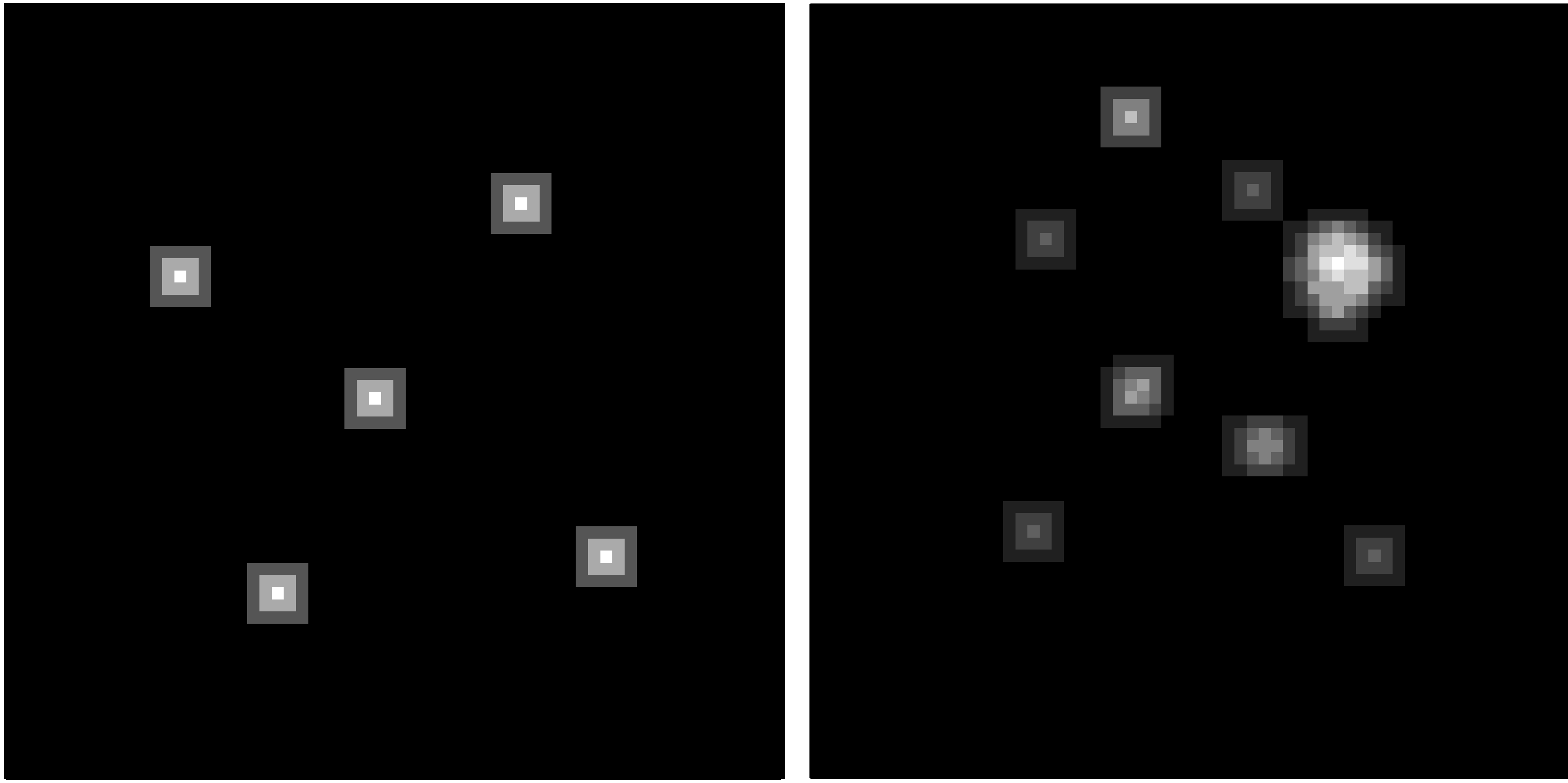
Content

- Linear systems
- Convolution
- Noise
- Denoising by linear filters
- Linear filters adapted to geometry
- Denoising by non-linear filters
- Spatially variant filters
- Fourier Transform
- Convolution Theorem
- Linear Deconvolution

Linear acquisition pipeline and image restoration



2D Convolution (stamping kernel)

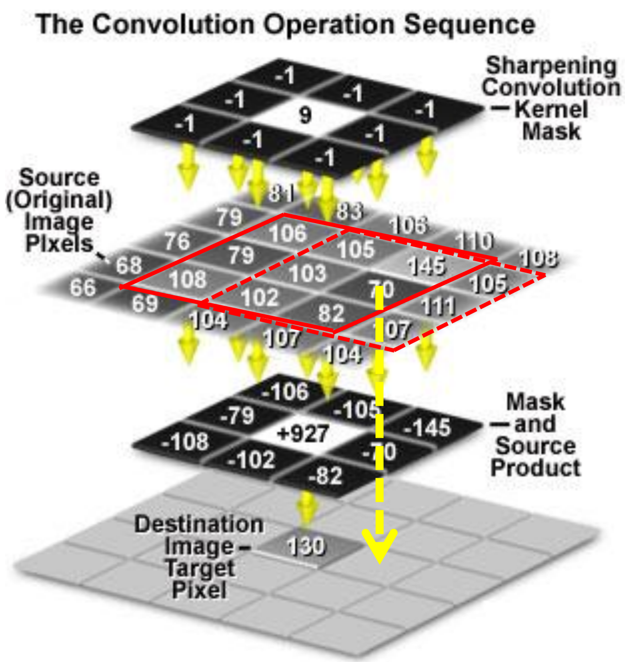


Filter kernel

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \leftarrow \rightarrow$$

Convolution: Accumulate a **filter kernel** stamped around every pixel and scaled by its intensity

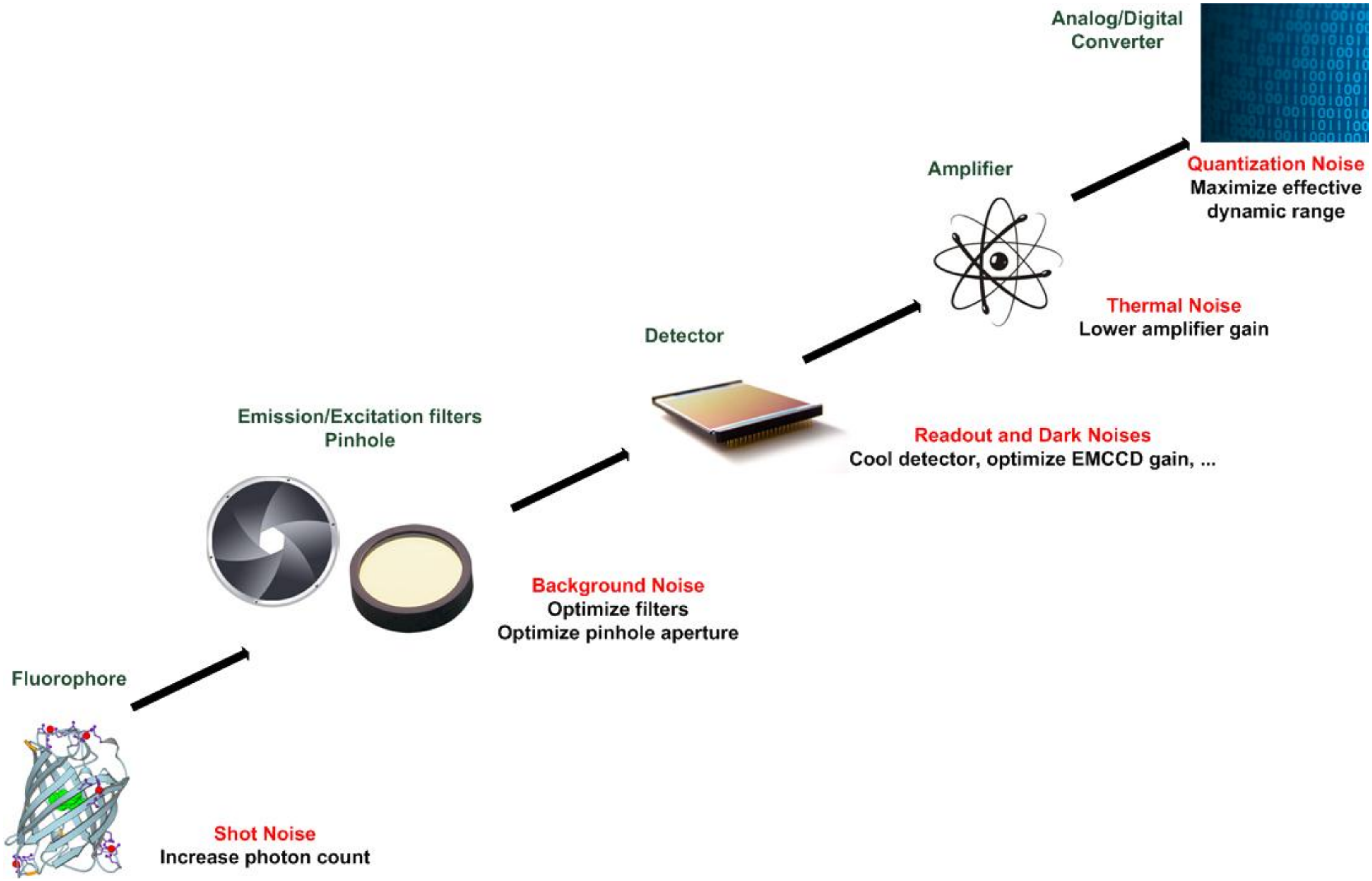
2D Convolution (sliding window)



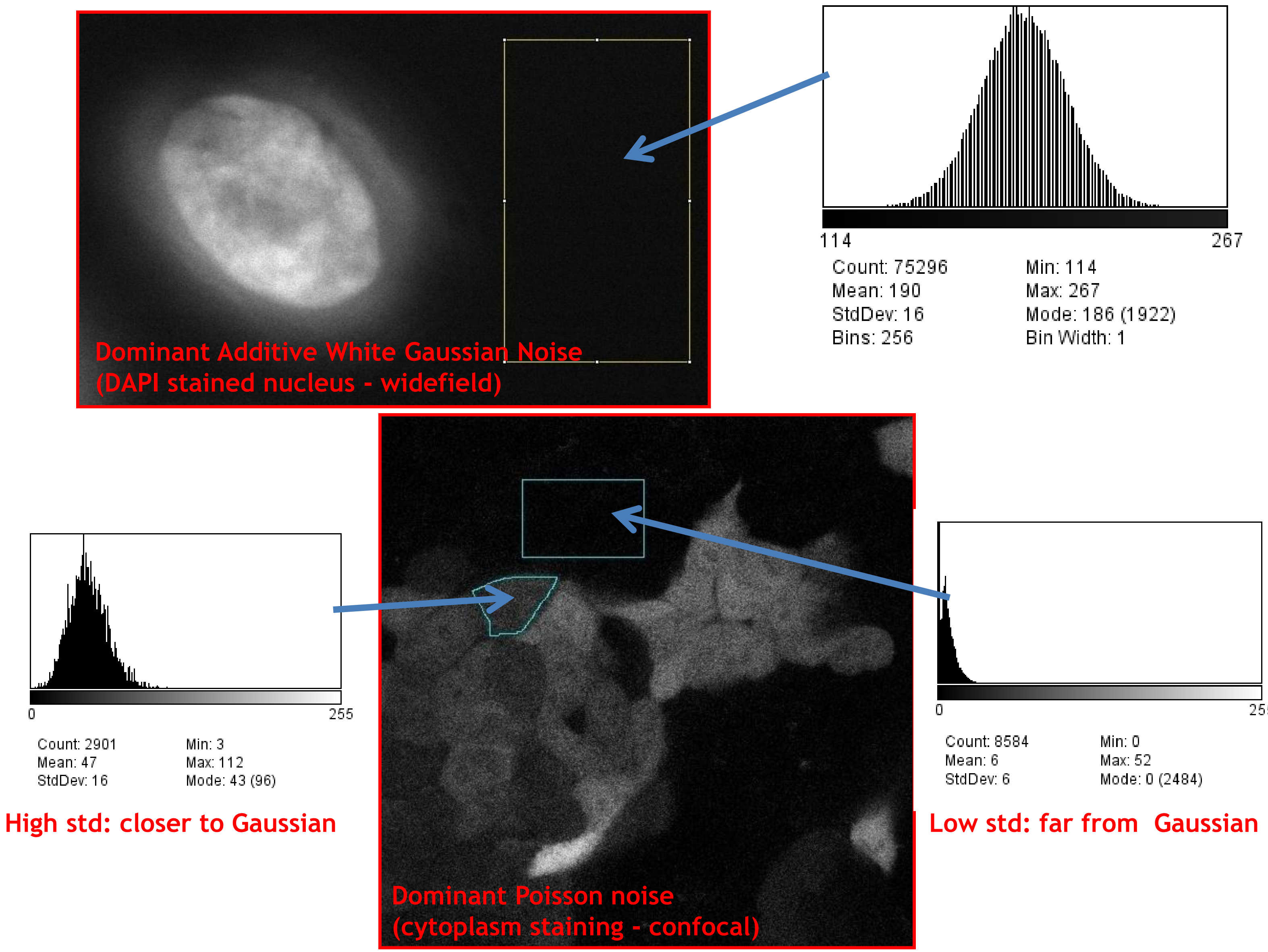
Sliding window: Central pixel = **weighted sum** of input image pixels in window

Weights: Filter kernel coefficients

Noise Sources in Digital Microscopy

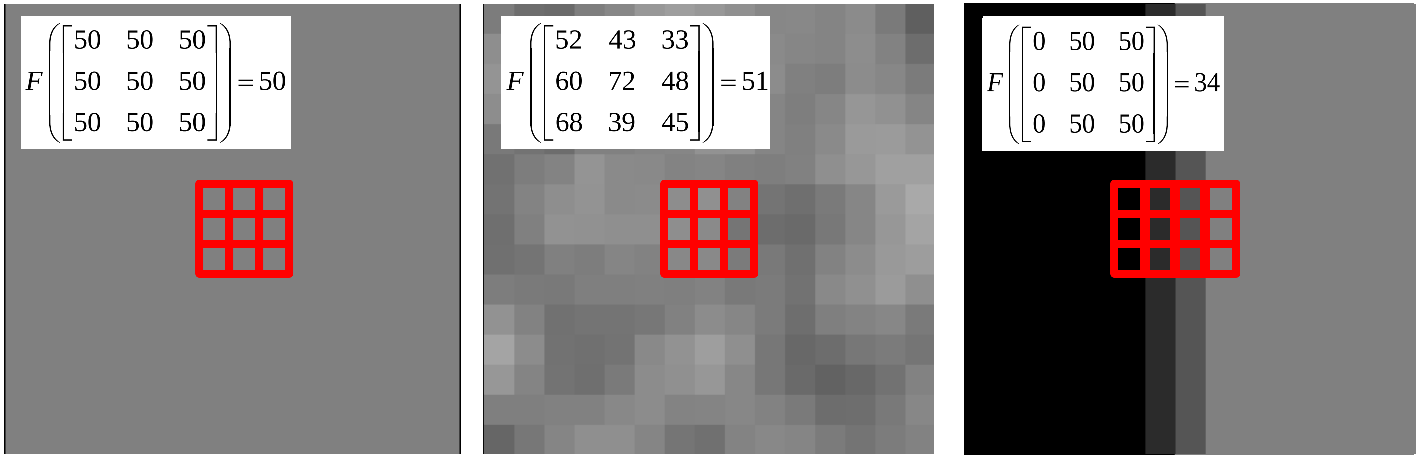


Noise Statistics



Simplest denoising filter: Smooth filter

$$F = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$



$$F\left(\begin{bmatrix} 50 & 50 & 50 \\ 50 & 50 & 50 \\ 50 & 50 & 50 \end{bmatrix}\right) = 50$$

$$F\left(\begin{bmatrix} 52 & 43 & 33 \\ 60 & 72 & 48 \\ 68 & 39 & 45 \end{bmatrix}\right) = 51$$

$$F\left(\begin{bmatrix} 0 & 50 & 50 \\ 0 & 50 & 50 \\ 0 & 50 & 50 \end{bmatrix}\right) = 34$$

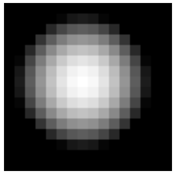
Uniform region (intensity = 50)

Noisy region (intensity = 50) + noise

Border between uniform regions

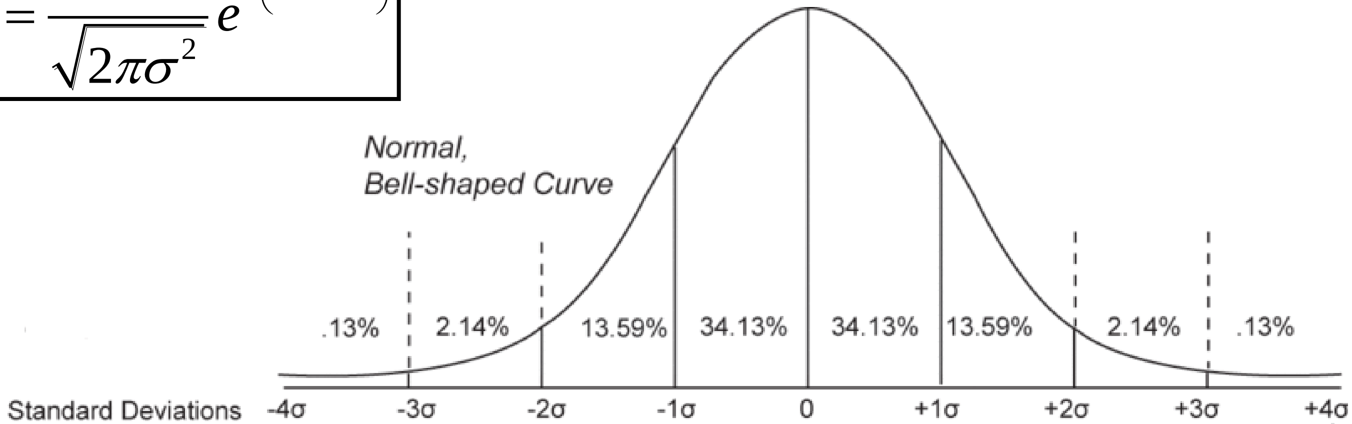
- Smooth filter (3x3 pixel kernel): **local averaging**
- Uniform regions: output = input
- Additive noise: negative/positive random variations around uniform value
- Average: reduces mean variation
- Edge: slow response, less steep (blurred)

Gaussian Function



2D Gaussian kernel

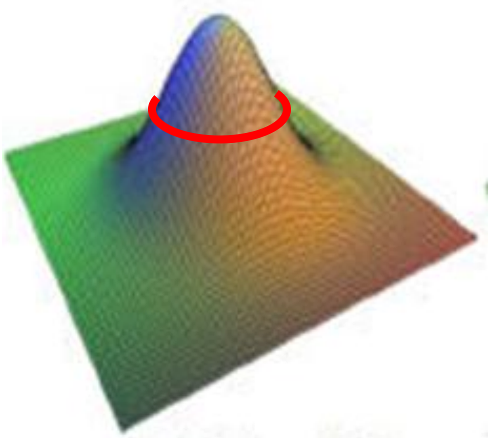
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$$



Most of the area below a Gaussian lies in [***-3σ***, ***3σ***]

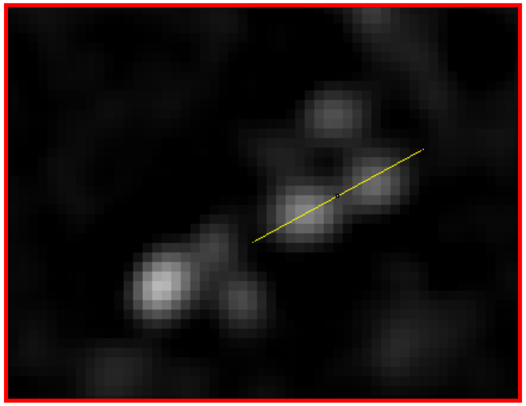
σ: Gaussian radius since radius of FWHM ring ≈ 1.18 ***σ***

f(x,y) = *f(x)*f(y)*

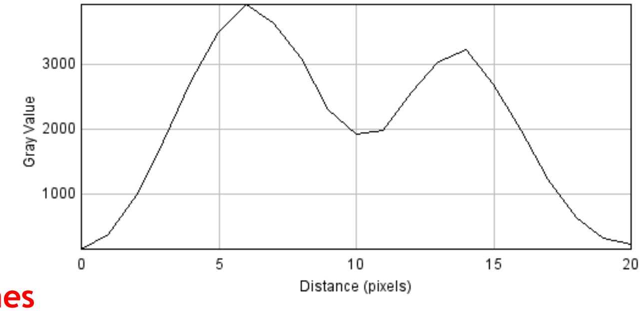


- Minimum time-bandwidth product: "slightest edge blur" for given noise rejection
- Smooth frequency cut-off
- Radially symmetric
- Separable

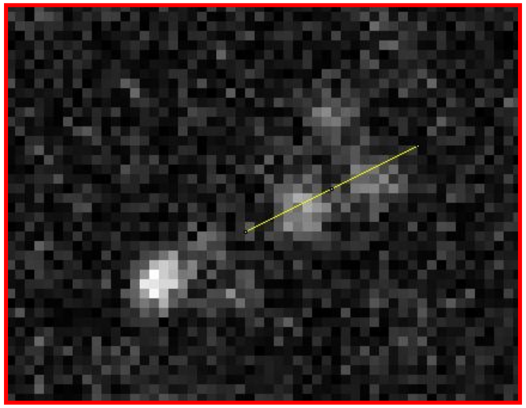
Gaussian Blur and Spatial Resolution



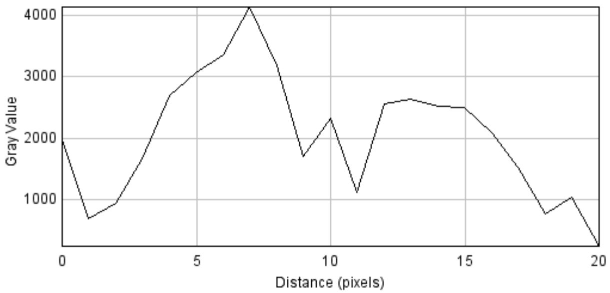
Autophagosomes widefield



Two objects can be resolved

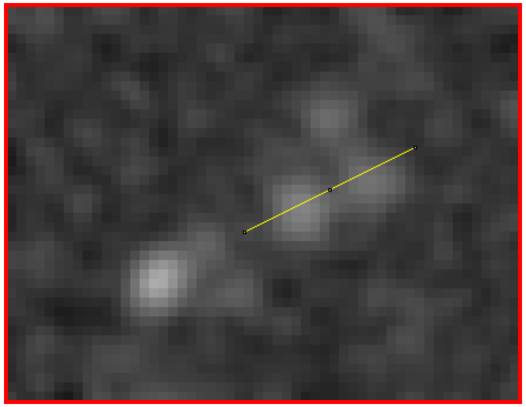


Same FOV, shorter exposure



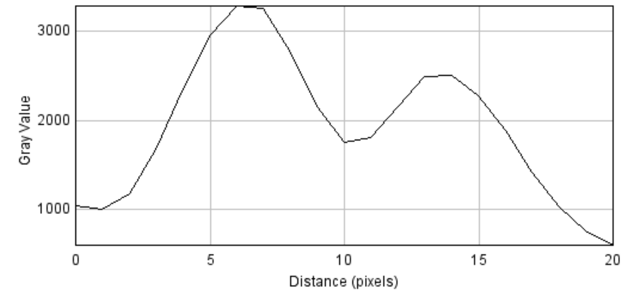
Objects cannot be unambiguously resolved

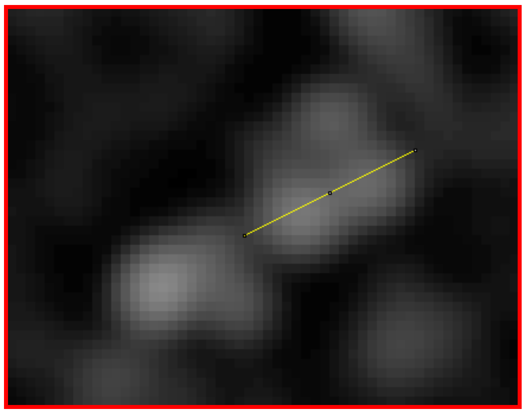
Denoising and Spatial Resolution



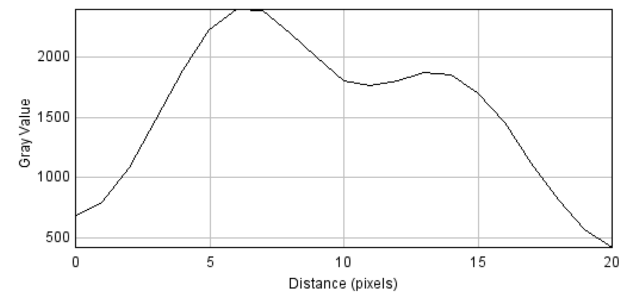
Short exposure:
Gaussian filtered
(radius = 1 pix)

Noise is reduced,
objects can be resolved

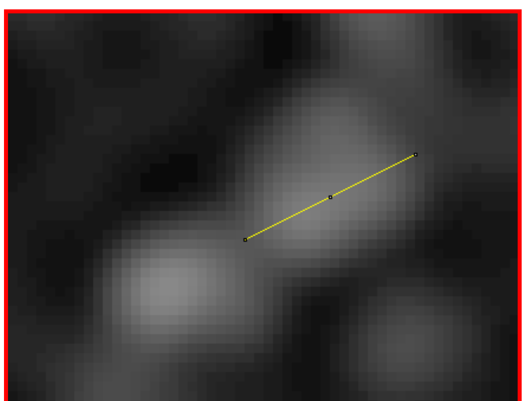




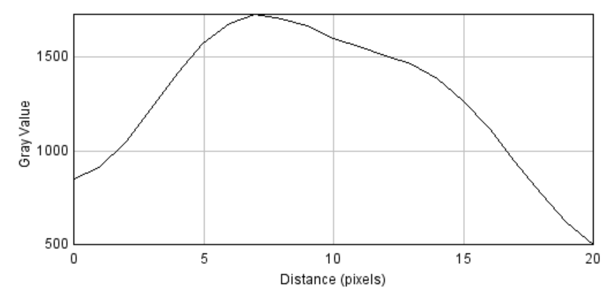
Short exposure:
Gaussian filtered
(radius = 2 pix)



The contrast between objects is reduced: loss of spatial resolution

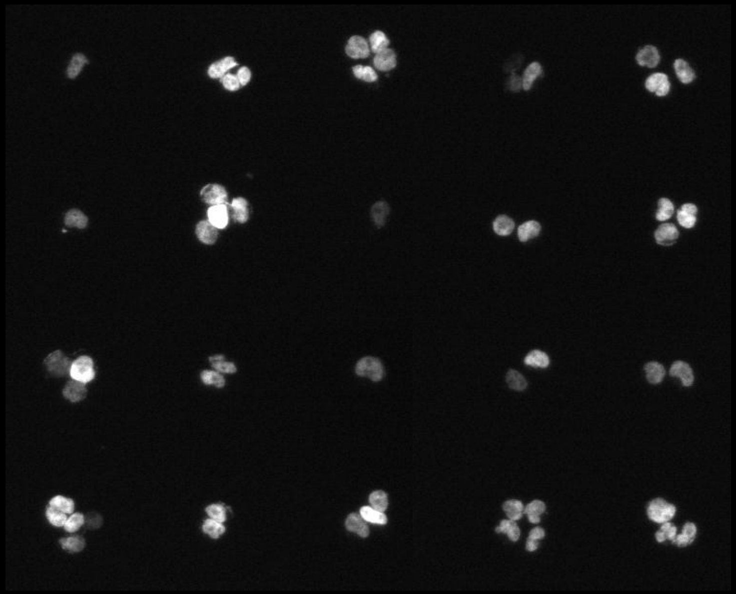


Short exposure:
Gaussian filtered
(radius = 3 pix)

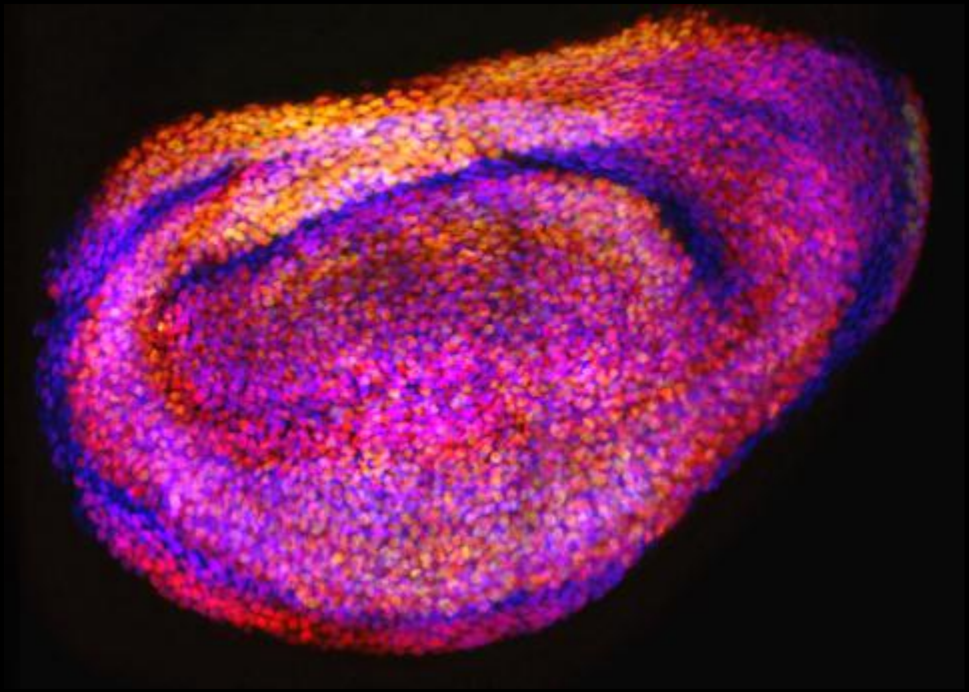


Roundish convex objects

Applications: Nuclei, round cells, round embryos,…



Cells seeded on Cytoo chip
DAPI stained nuclei
(confocal 20x)
MIP



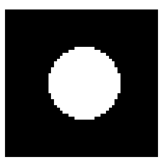
Drosophila imaginal disk
DAPI stained nuclei
(confocal 40x)
Color coded MIP

Laplacian of Gaussian (LoG) Filter

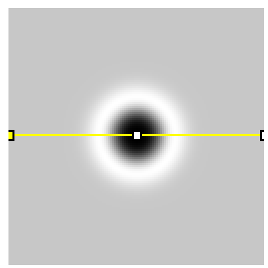
→ Plugins/Feature Extraction/FeatureJ/Laplacian

Laplacian filter kernel (2D): $$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

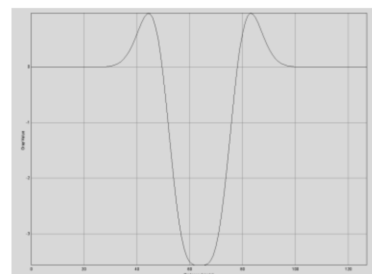
Laplacian of Gaussian filter(LoG):
Gaussian blur kernel (σ) followed by Laplacian filter



Original image



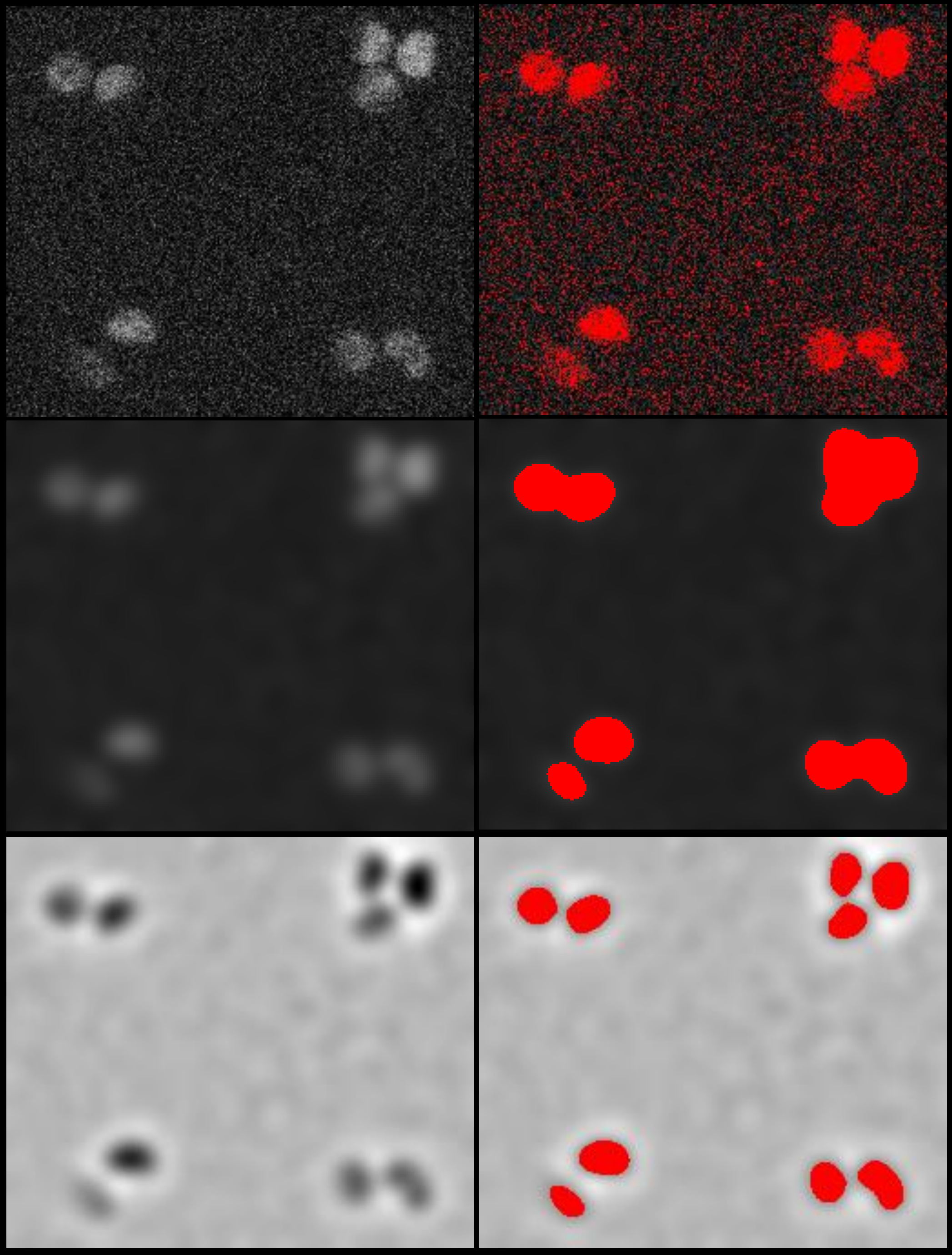
LoG filtered



Intensity **profile**

Maximum contrast:

$$\sigma = \frac{r}{2}$$



Original

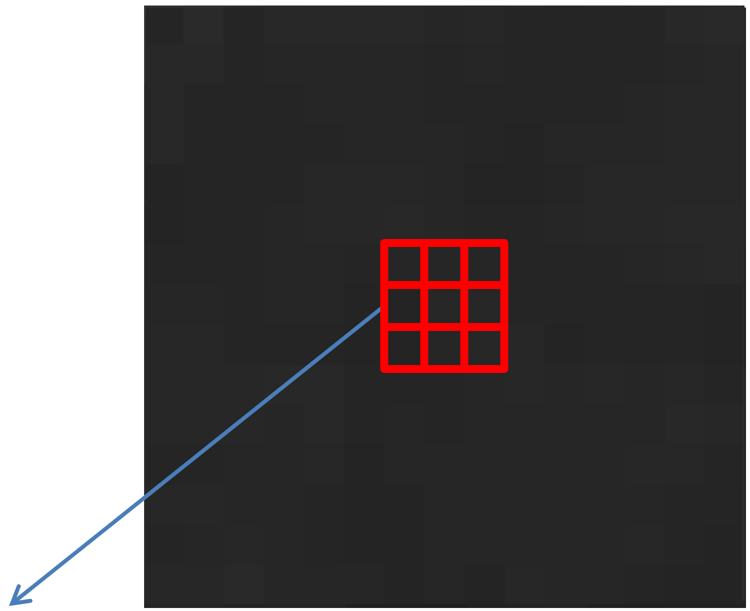
Gaussian
blurred
(r= 5 pix)

LoG
(r= 7 pix)

Non-linear Filters

Also implemented as sliding window but operation inside window is **non linear**

<u>Order statistics filters</u>: minimum, maximum and median



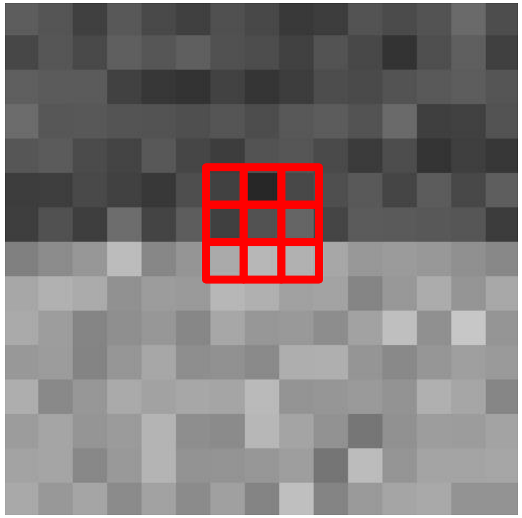
$$\begin{bmatrix} 36 & 40 & 36 \\ 34 & 39 & 246 \\ 7 & 45 & 39 \end{bmatrix}$$

Intensity values sorting

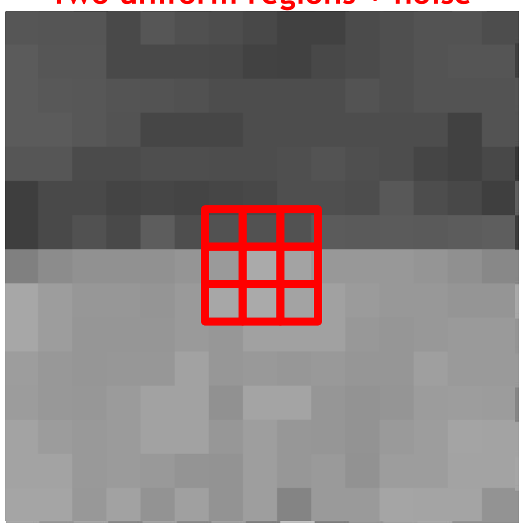
$$\rightarrow [7 \quad 34 \quad 36 \quad 36 \quad 39 \quad 39 \quad 40 \quad 45 \quad 246]$$

min median max

Median Filter



Two uniform regions + noise



Median filter attenuates noise while preserving edge transitions

$$M\left(\begin{bmatrix} 65 & 39 & 72 \\ 64 & 82 & 101 \\ 170 & 183 & 177 \end{bmatrix}\right) = 82$$

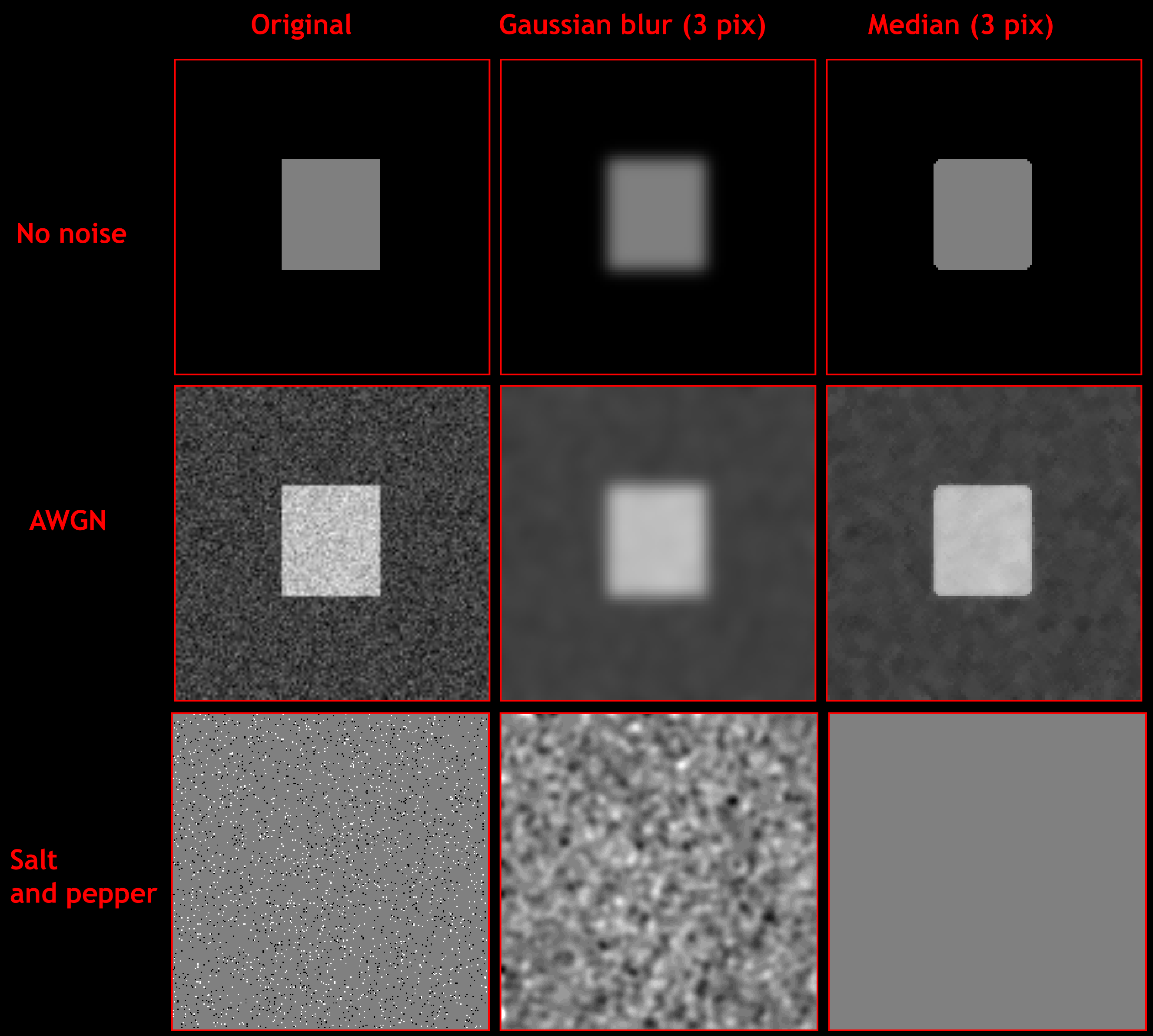
median

$$[\underbrace{39 \quad 64 \quad 65 \quad 72 \quad 82 \quad 101}_{\text{darkest pixels}} \quad 170 \quad 177 \quad 183]$$

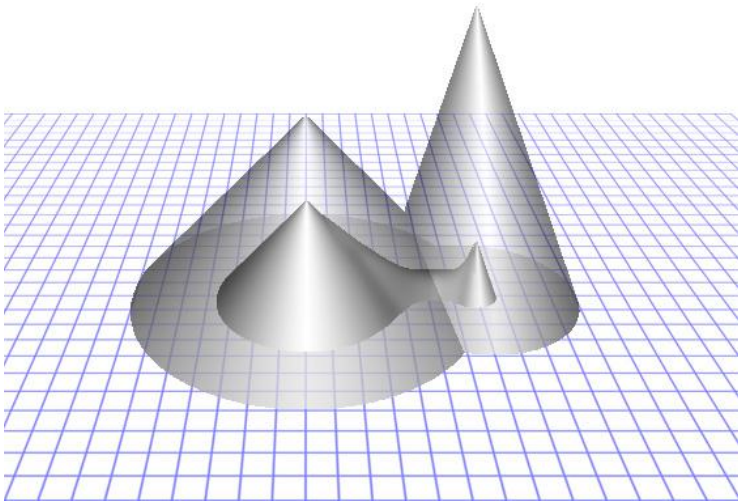
$$M\left(\begin{bmatrix} 64 & 82 & 101 \\ 170 & 183 & 177 \\ 183 & 176 & 161 \end{bmatrix}\right) = 170$$

median

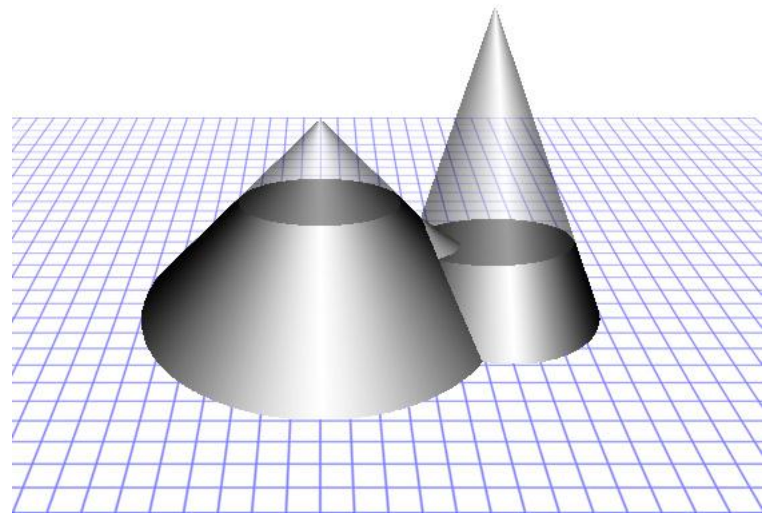
$$[64 \quad 82 \quad 101 \quad \underbrace{161 \quad 170 \quad 176 \quad 177 \quad 183 \quad 183}_{\text{brightest pixels}}]$$


Original
Gaussian blur (3 pix)
Median (3 pix)
No noise
AWGN
Salt
and pepper

Grayscale Morphology: Erosion



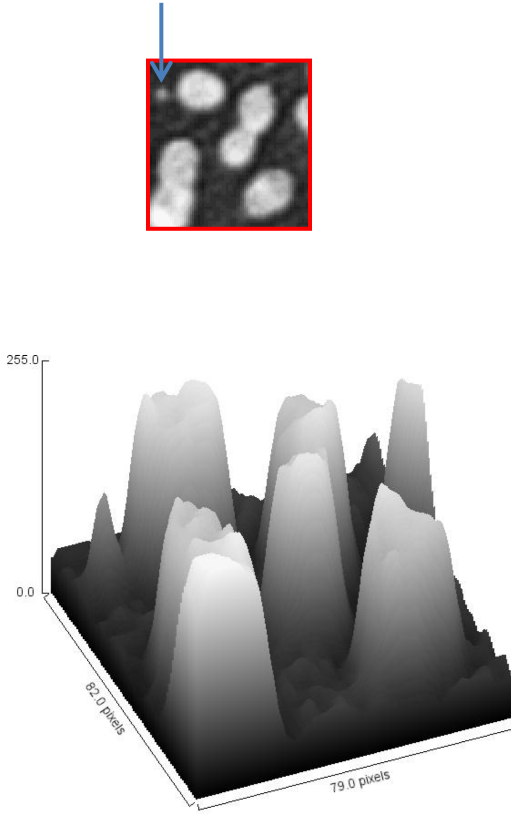
Erosion (plain)



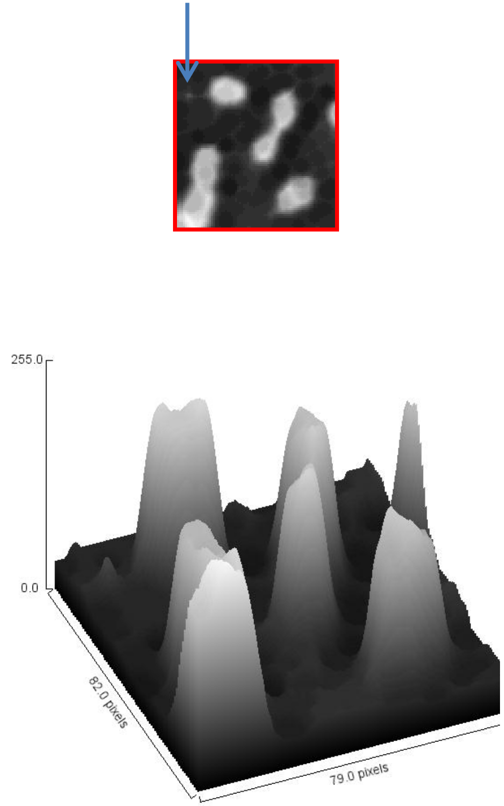
Opening (plain) and original (transparent)

Opening = Erosion (min) + Dilation (max)
Better preserves objects than erosion

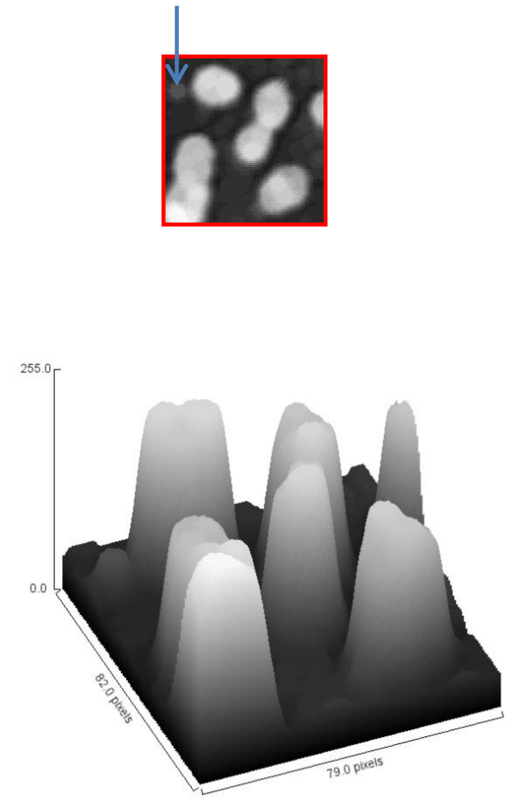
Grayscale Morphology: Opening



Original



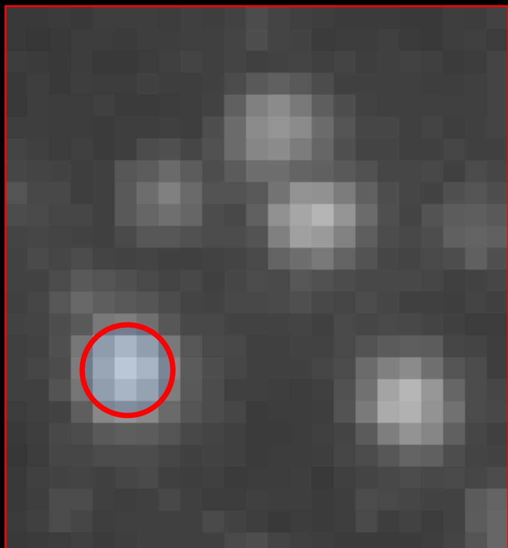
Erosion



Opening

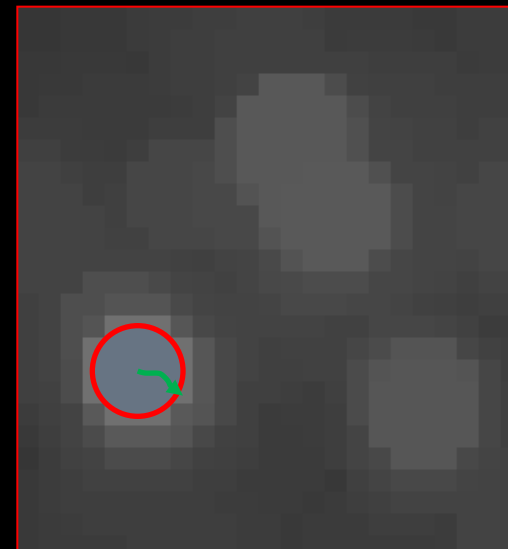
Opening Filter

→ Morphology/Gray morphology

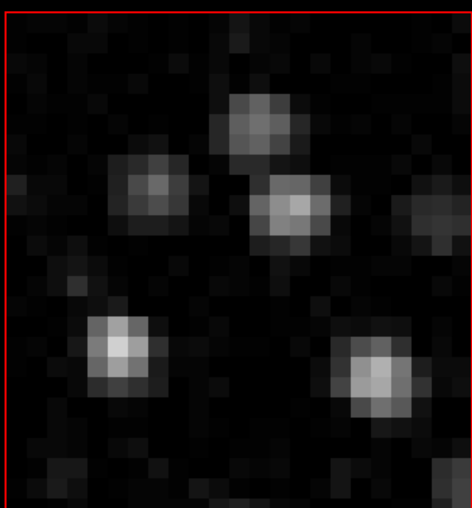


1: Original

2: Min filter (erosion)



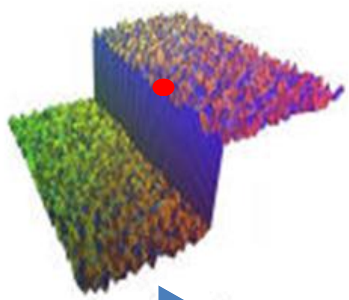
3: Max filter of 2 (dilation)



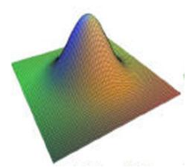
4: Top-hat opening (1-3)

Top-hat opening is useful for background subtraction

Bilateral Filter

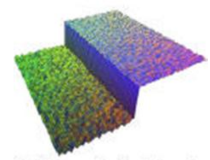


Initial kernel



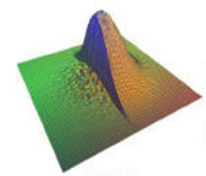
x

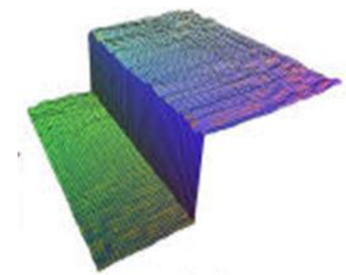
Modulation function



=

Filter kernel





$$g(x,y) = e^{-\left(\frac{(A(x,y)-A(x_0,y_0))^2}{\sigma_R{}^2}\right)}$$

Original image (topographic view): the central pixel of the filter window is right on the edge.

Bilateral filter result ($\boldsymbol{\sigma_S}$ = 5 pixels, $\boldsymbol{\sigma_R}$ = 50 gray level)

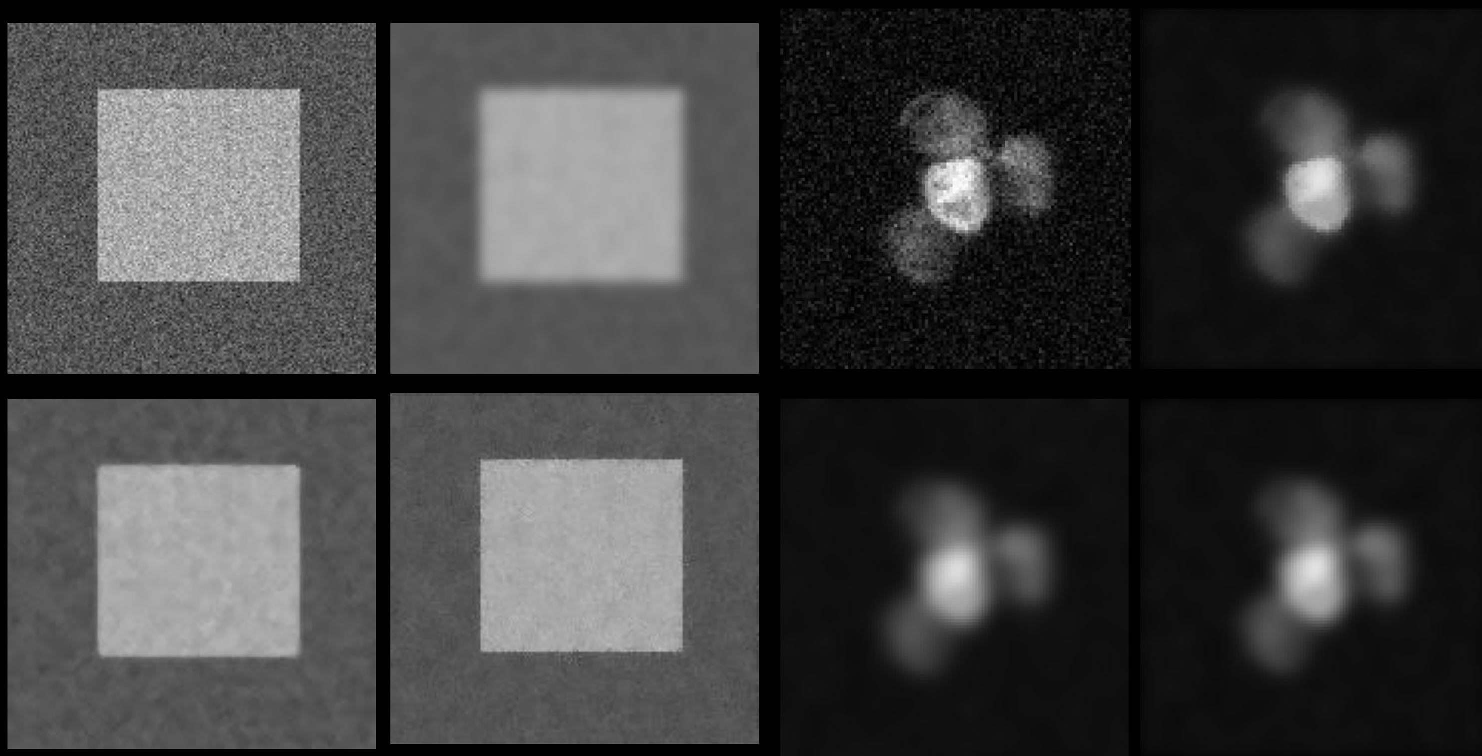
Initial kernel ***F*** Gaussian (standard deviation $\boldsymbol{\sigma_S}$).

Kernel multiplied by modulation function of intensity difference to central pixel (Intensity range parameter $\boldsymbol{\sigma_R}$).

--> Resulting kernel is a function of image data (**spatially varying**).

Bilateral Filter

Original | Gaussian blur (3 pix rad) | Original (DAPI stained nuclei) | Bilateral filter (3 pix rad, adjusted range)

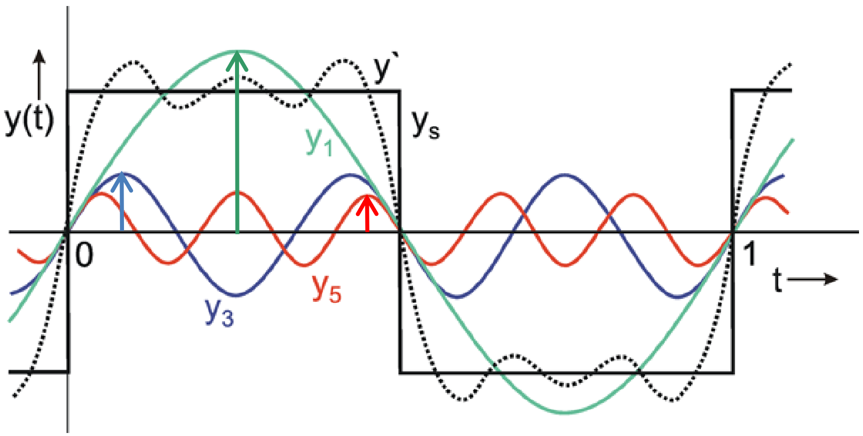


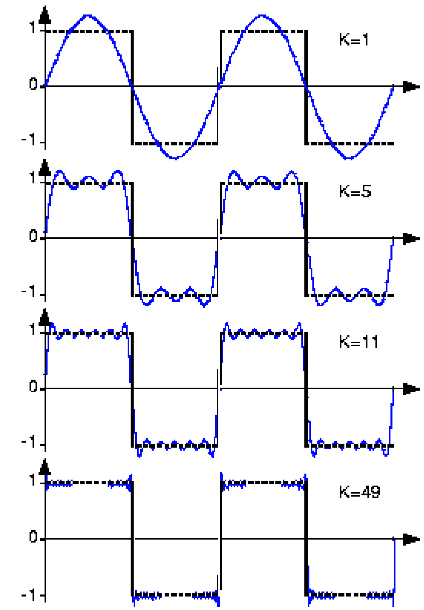
Median (3 pix) | Bilateral (6 pix, 100 levels) | Gaussian blur (3 pix) | Bilateral filter (3 pix, inf. range)

1D Fourier Transform

A periodic function can be approximated by a sum of sine waves with proper shifts and scalings.

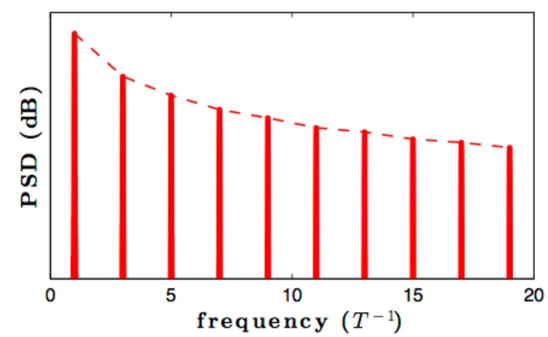
The Fourier coefficients are complex numbers coding the scaling (magnitude) and shift (phase).



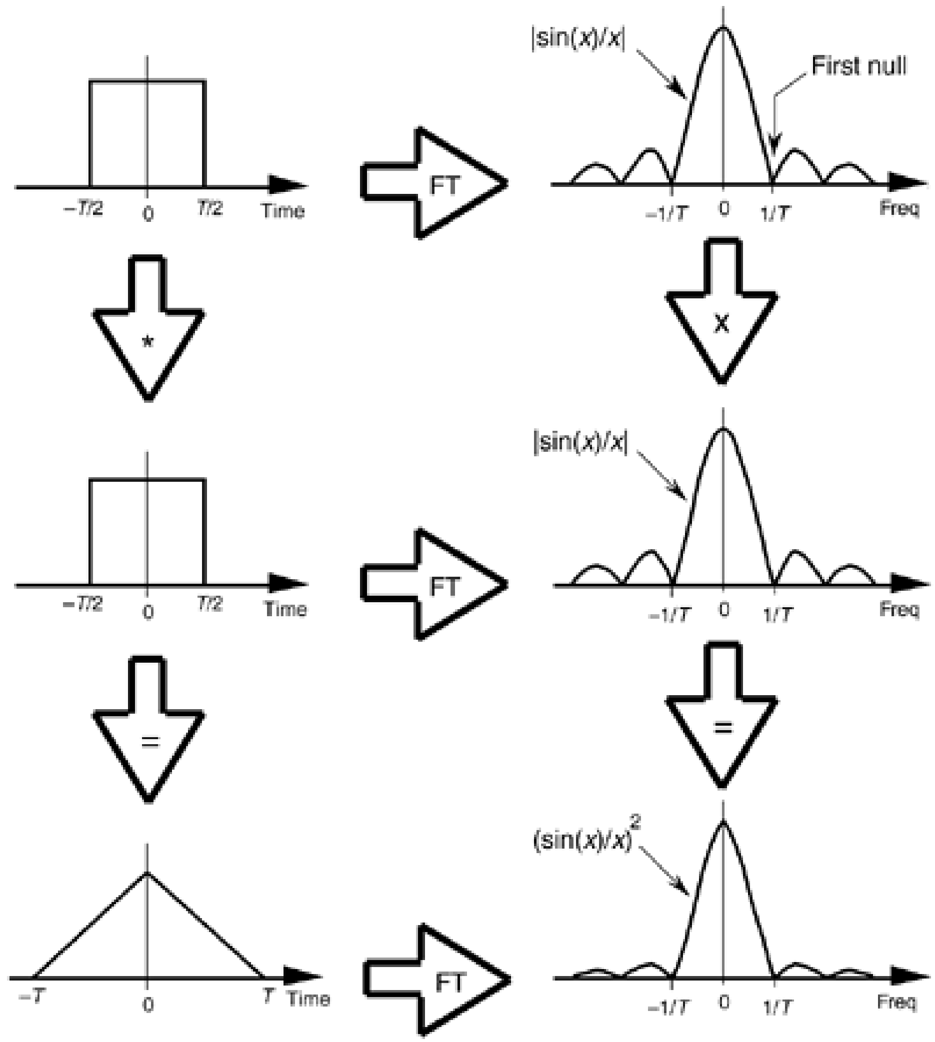


- The approximation accuracy depends on the number of sines involved and the smoothness of the function.

- The square modulus of the Fourier coefficients are the so-called power spectrum of a signal.



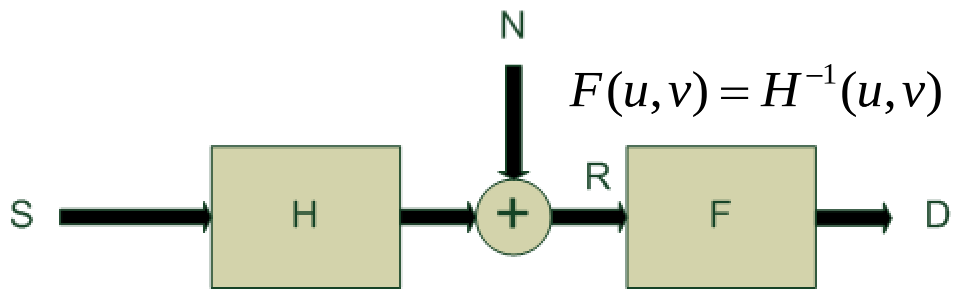
Convolution Theorem

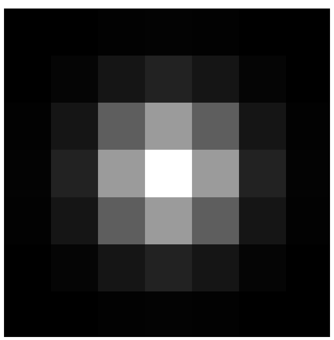


Linear Deconvolution: Inverse Filter Deconvolution

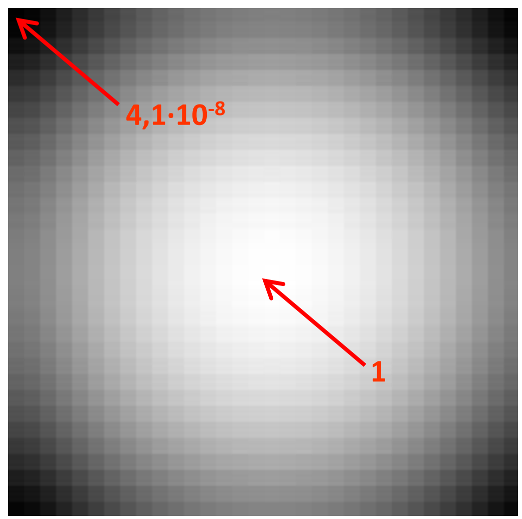
As convolution in the spatial domain can be performed as a multiplication in the frequency domain, inverse filtering can be performed as a division in the frequency domain!

$$F(u,v) = H^{-1}(u,v)$$

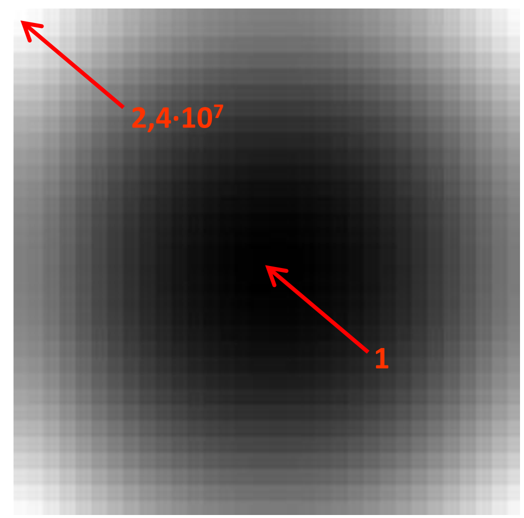




A very simple model for the PSF *H* (Gaussian std = 1 pixel)



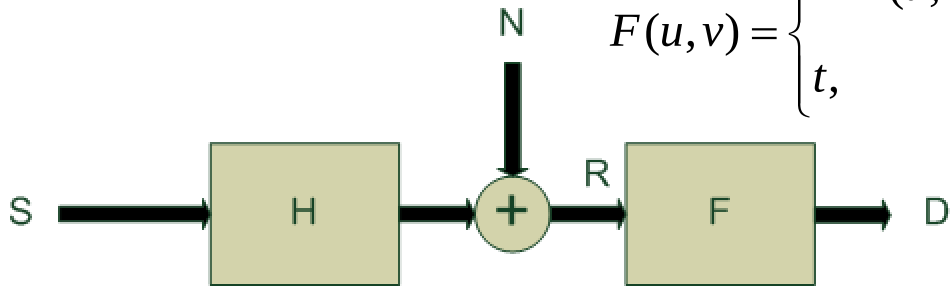
H power spectrum (log display) overlaid with raw values

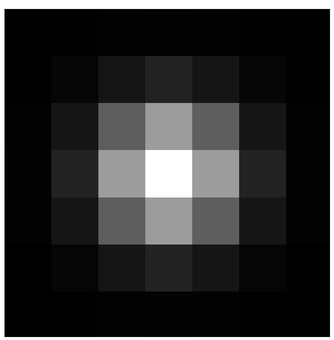


H^{-1} power spectrum (log display) overlaid with raw values

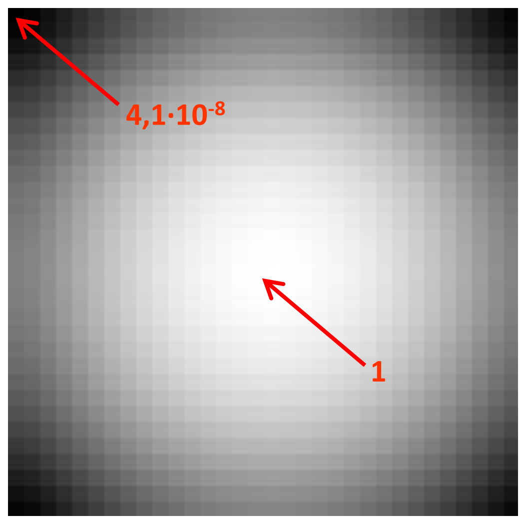
Second try: Regularized Inverse

$$F(u,v)=\begin{cases} H^{-1}(u,v), & \left|H^{-1}(u,v)\right|\le t \\ t, & \left|H^{-1}(u,v)\right|> t \end{cases}$$

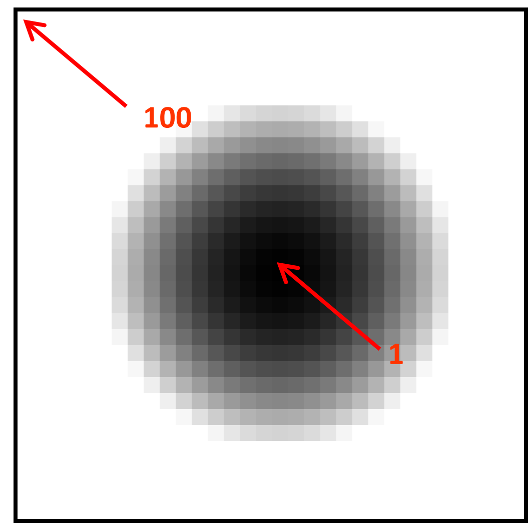




A very simple model
for the PSF *H*
(Gaussian std = 1 pixel)



H power spectrum (log display)
overlaid with raw values



$(H^{-1})_{reg}$ (1% clipping)
power spectrum (log display)
overlaid with raw values

Optimum linear restoration: Wiener filter

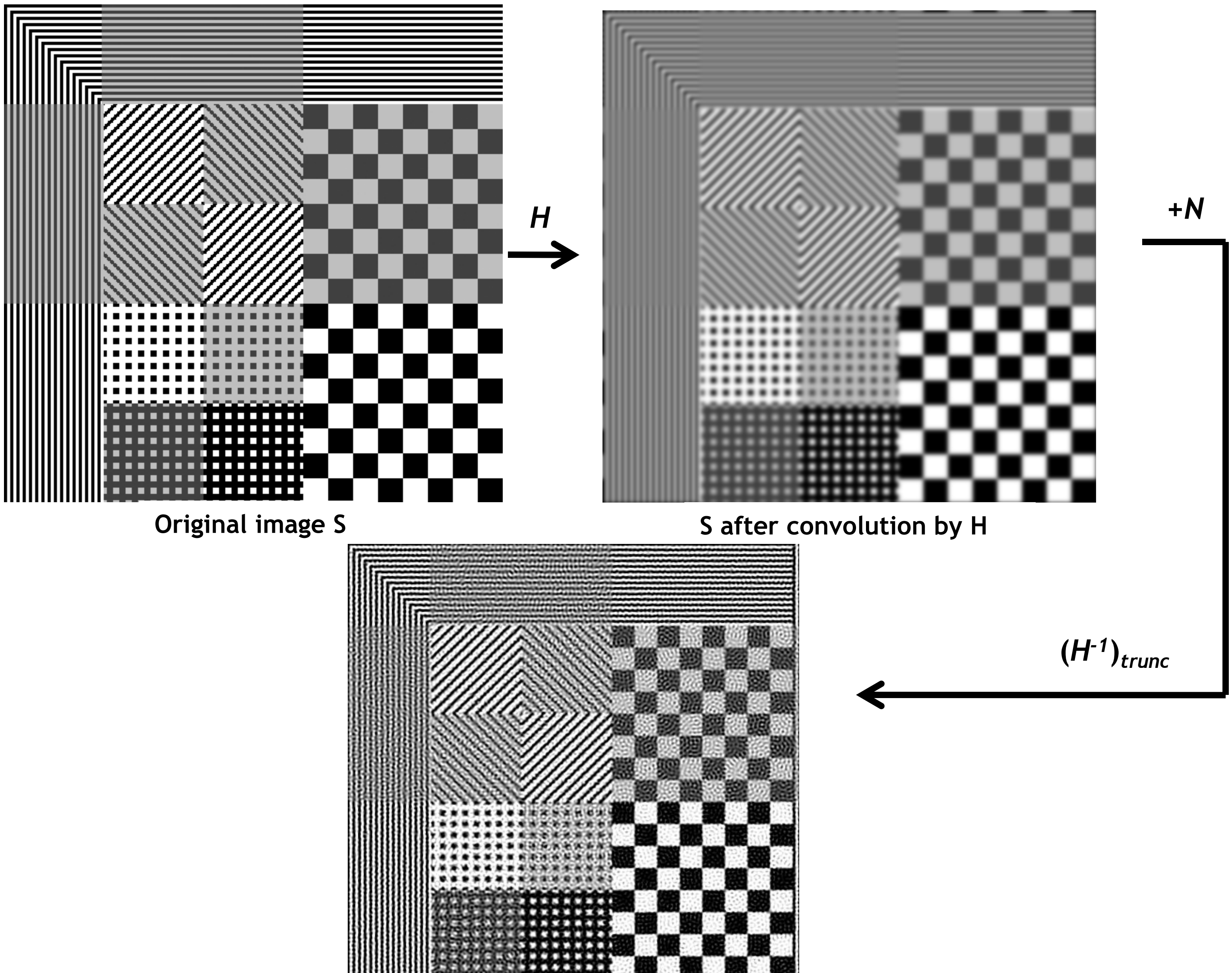
$$\boxed{F(u,v) = \frac{H^*(u,v)\left|S(u,v)\right|^2}{\left|H(u,v)\right|^2 \cdot \left|S(u,v)\right|^2 + \left|N(u,v)\right|^2}}$$

Wiener filter

<u>Bands free of noise</u>:	$\lvert N(u,v)\rvert = 0$	$\rightarrow F(u,v) = H(u,v)^{-1}$	(inverse filter)
<u>Strong noise bands</u>:	$\lvert N(u,v)\rvert \rightarrow \infty$	$\rightarrow F(u,v) \rightarrow 0$	(cut-off)
<u>Intermediate bands:</u>			best trade-off

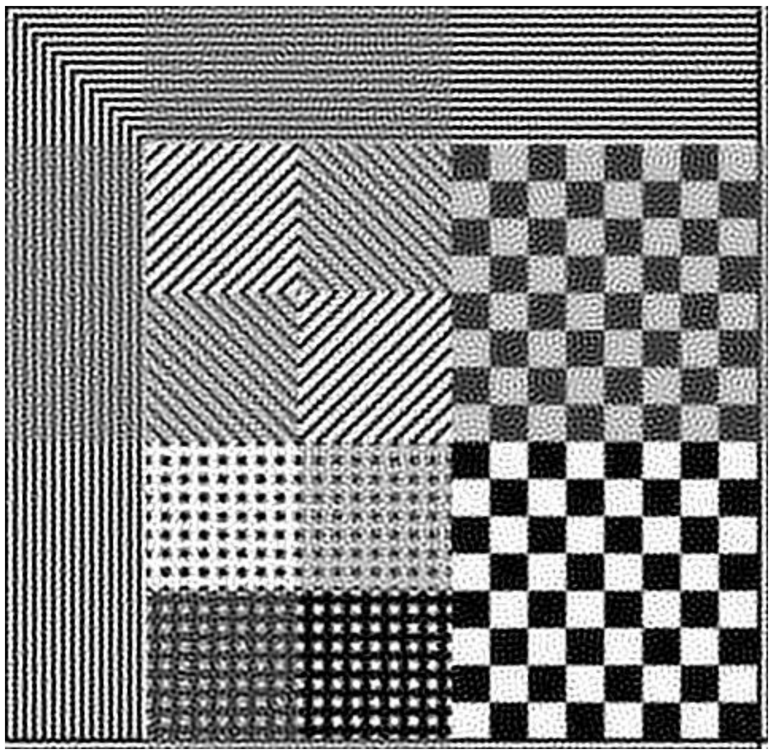
The solution depends on both N and S (unknown)

Regularized Inverse Filter Deconvolution

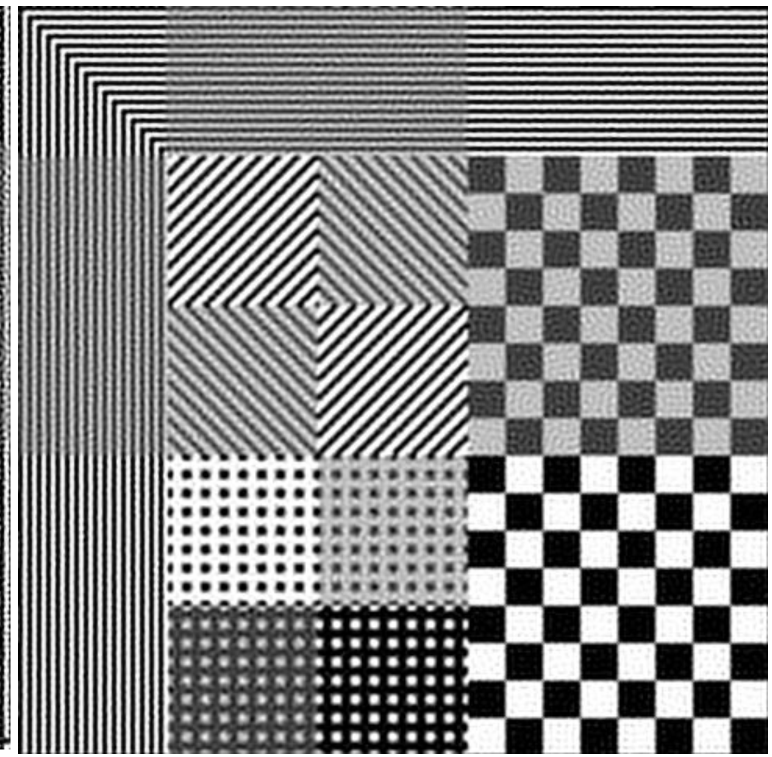


Noise std = 10^{-4}

Wiener Deconvolution



Regularized inverse filter result
Noise std = 10^{-4}



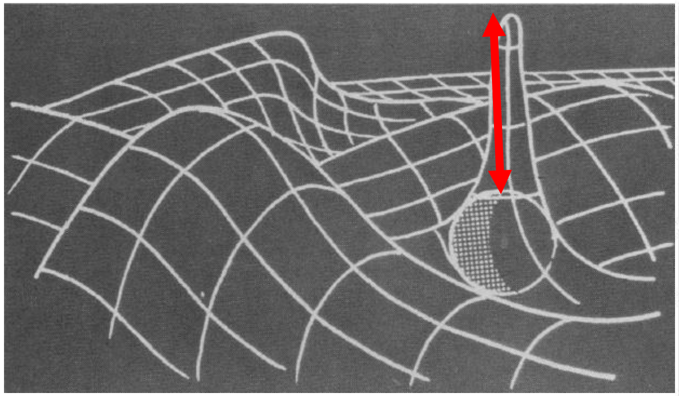
Wiener filter result
Noise std = 10^{-4}

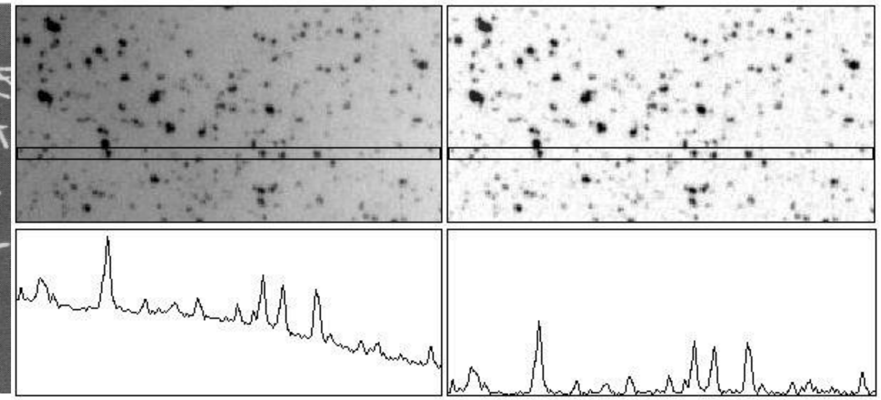
EXTRA SLIDES

Rolling Ball Algorithm ≈ Top-hat opening

→ Process/Subtract Background...

video inverted images

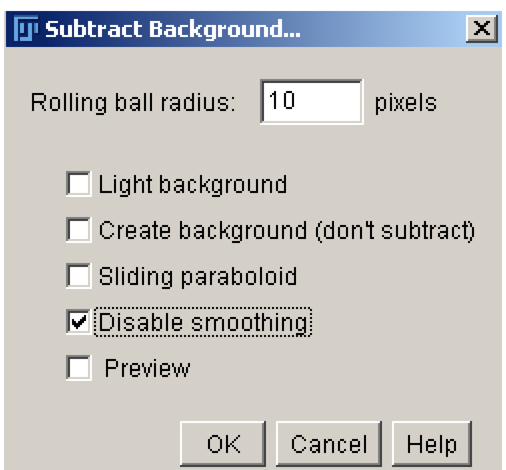




Before

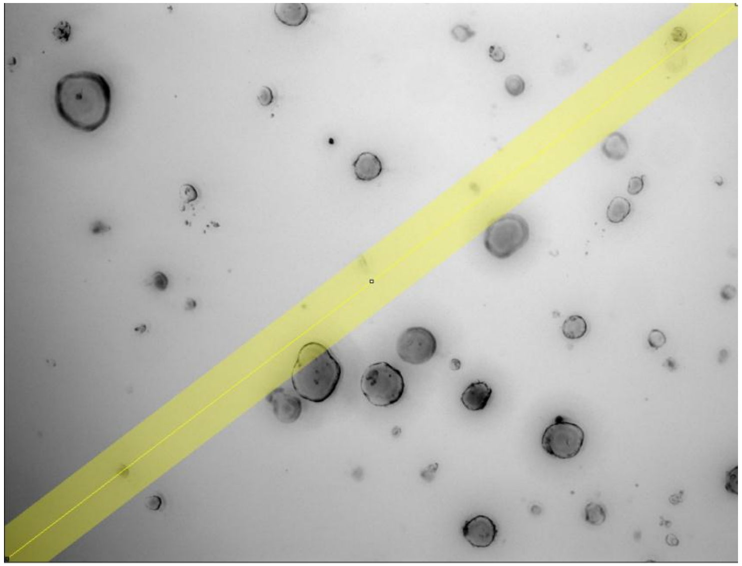
After

For each pixel the level of the tip of a rolling sphere is subtracted to the image

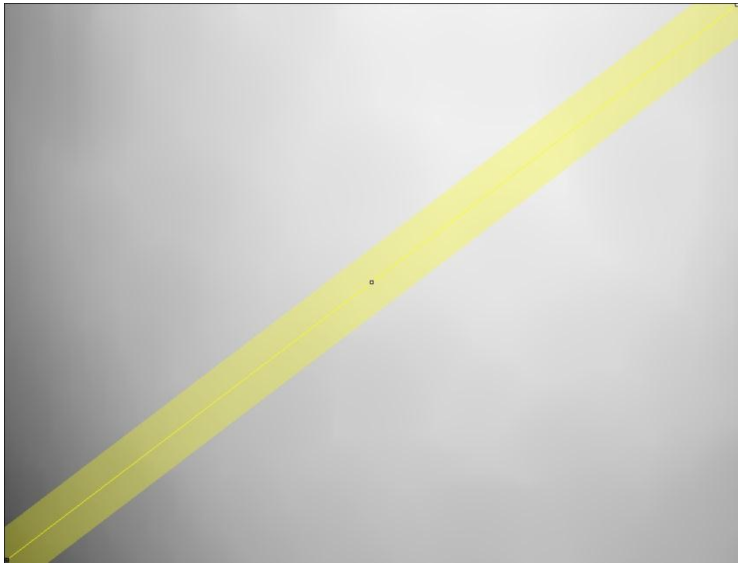


- Light background: dark objects over bright background
- Create background: returns background
- Sliding paraboloid: approximation (faster)
- Disable smoothing: ensures that results >0

Illumination Correction

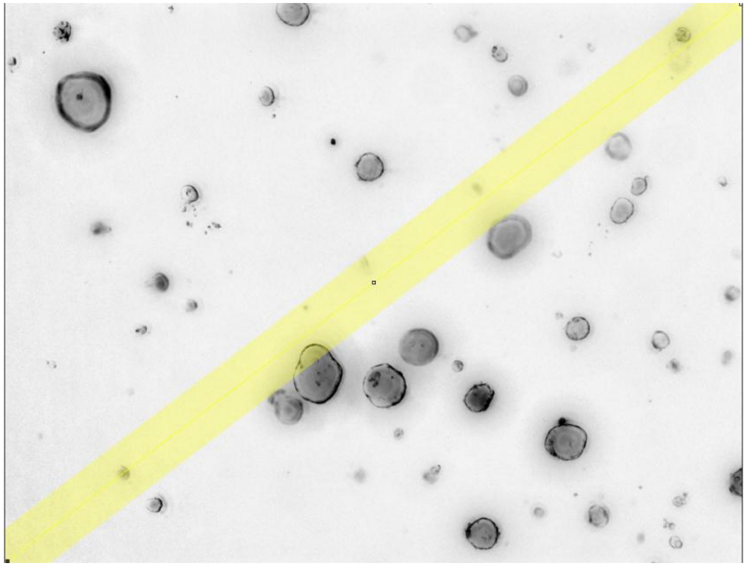


1. Original (Z min. proj)

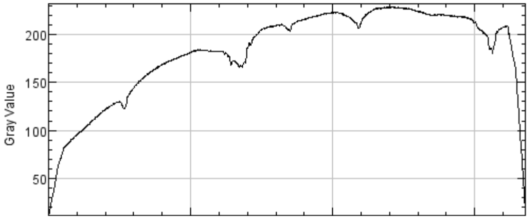


2. After subtract background (r=256 pix) with:

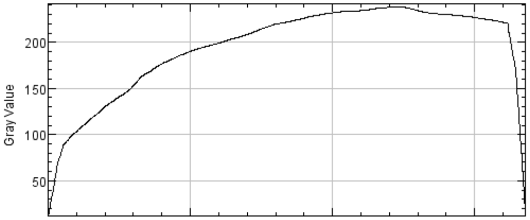
-Light background
-Create background
-Disable smoothing



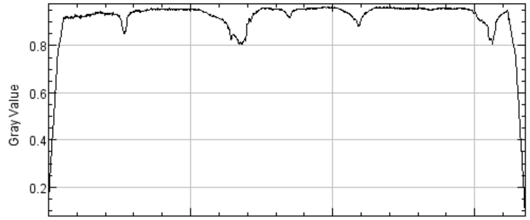
1. divided by 2.



1.



2.



3.