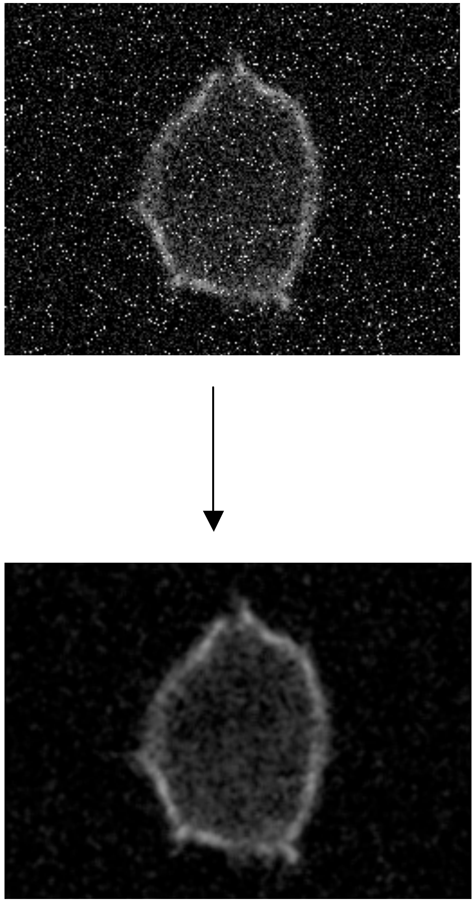
Basic techniques in IP - part 2:

Image processing in the spatial / frequency / time domain.

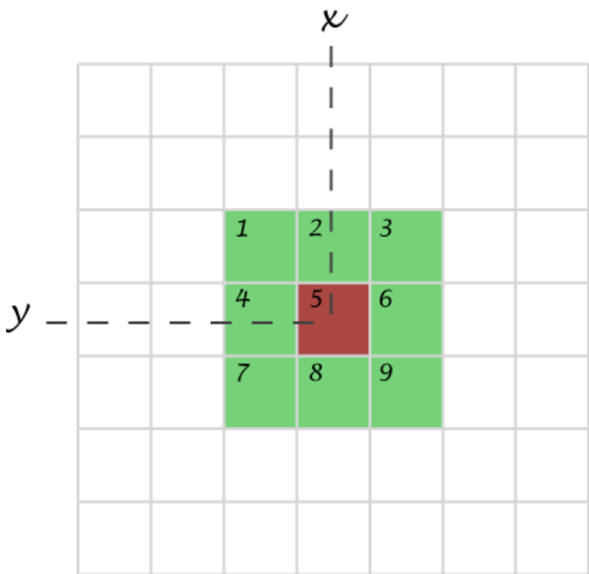
Filtering - Neighborhood -
Fourier space - Time series

I. Image processing in the spatial domain



- A. Introduction
  - Neighborhood
  - Operation on neighbors
- B. Spatial filters
  - Mean filter
  - Median filter
  - Edge detection

A. Introduction



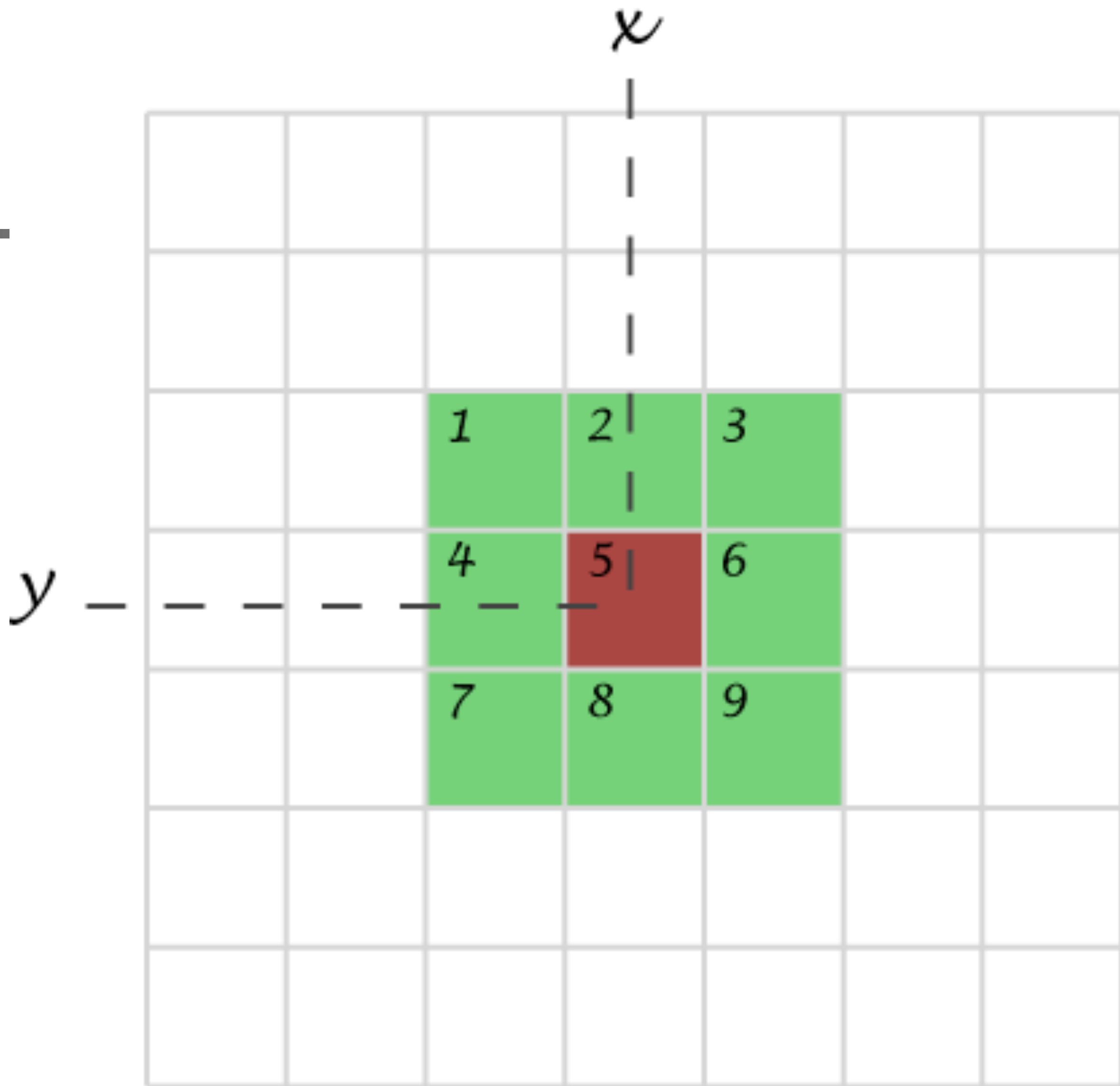
Neighborhood (or kernel):
pixels that matters

- <u>Definition</u>

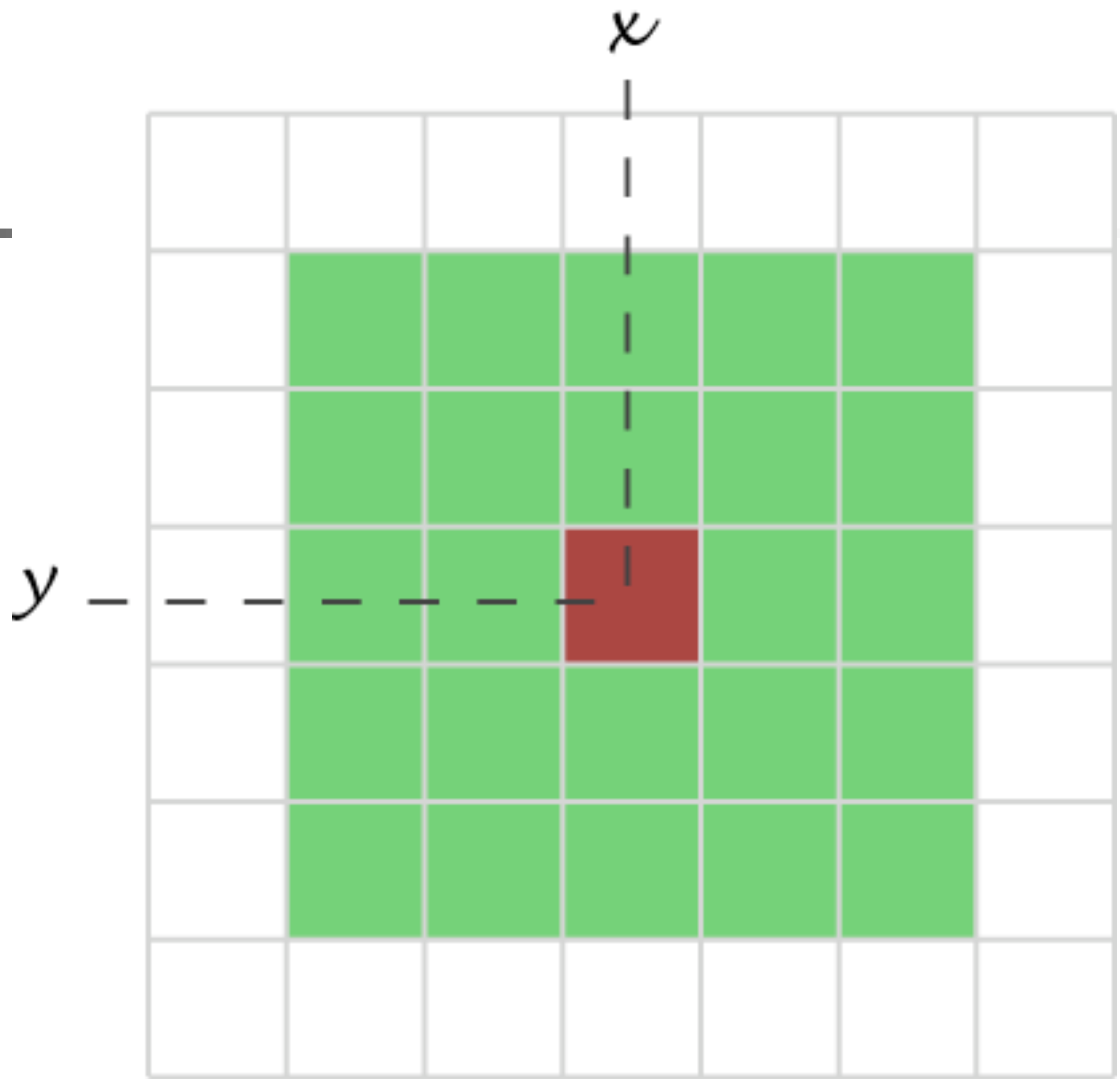
“ Transformation or set of transformations where a new image is obtained by *neighborhood operations*.”

→ The intensity of a pixel in the new image depend on the intensity values of “neighbor pixels”.

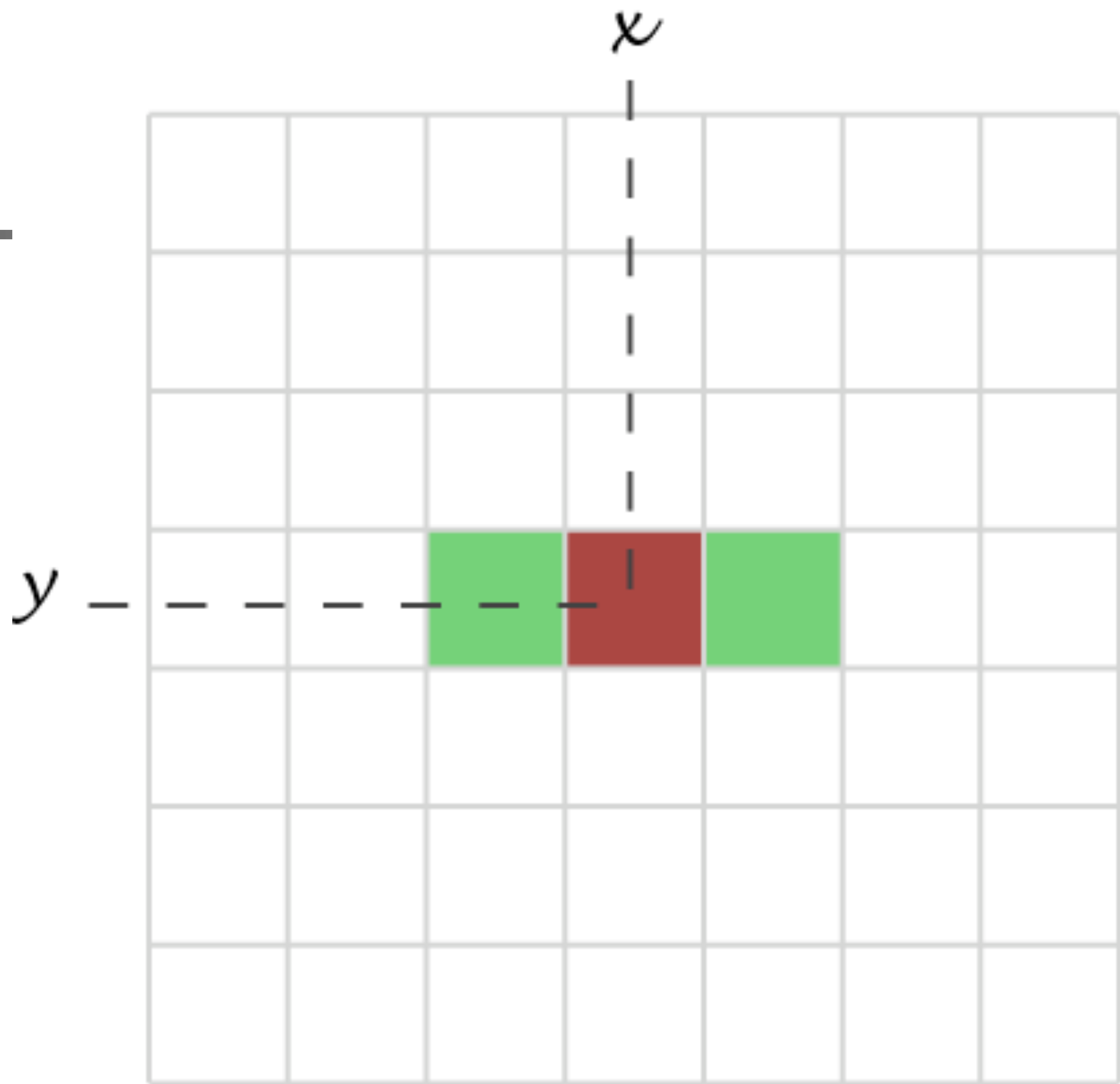
3x3



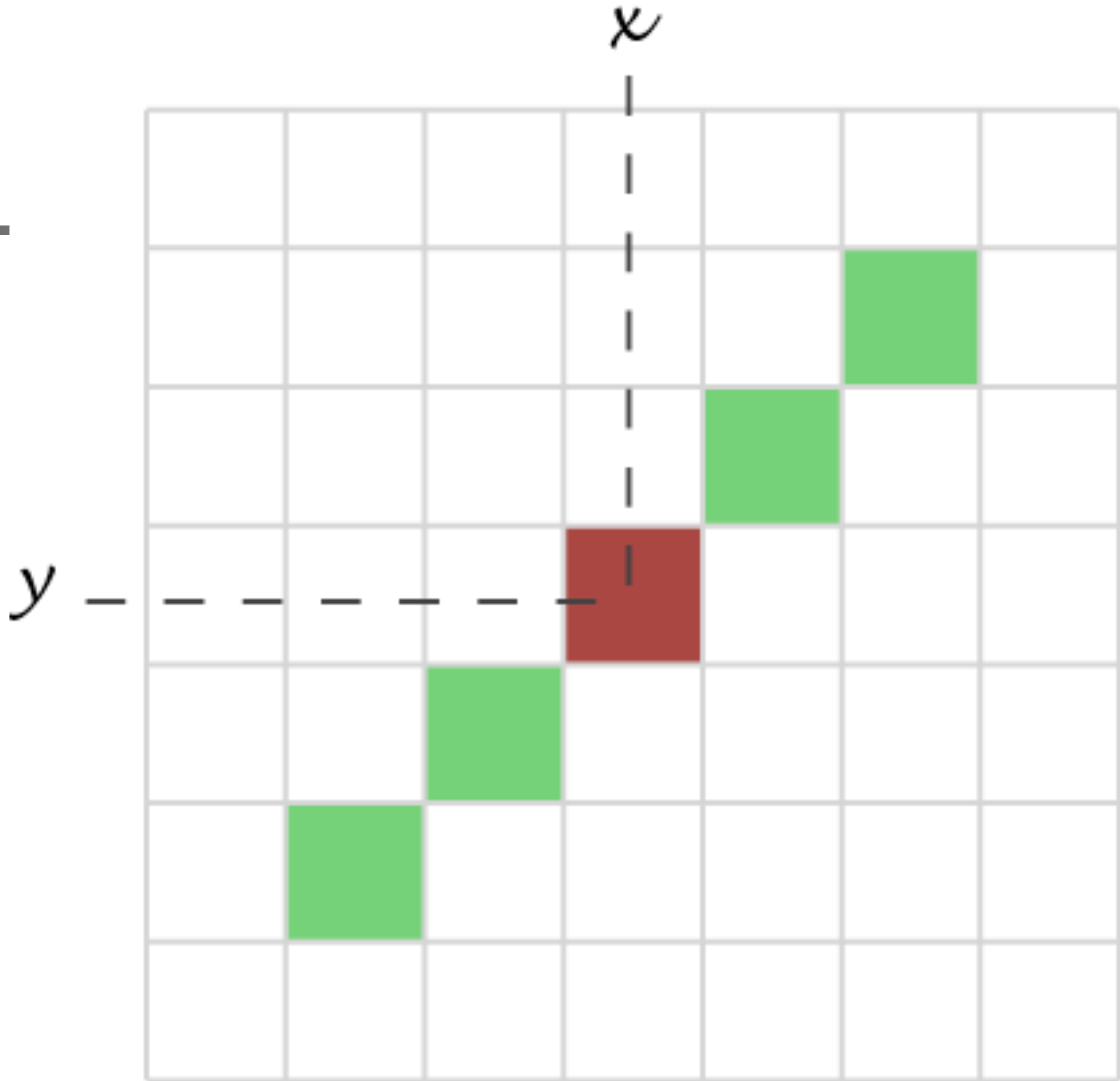
5x5



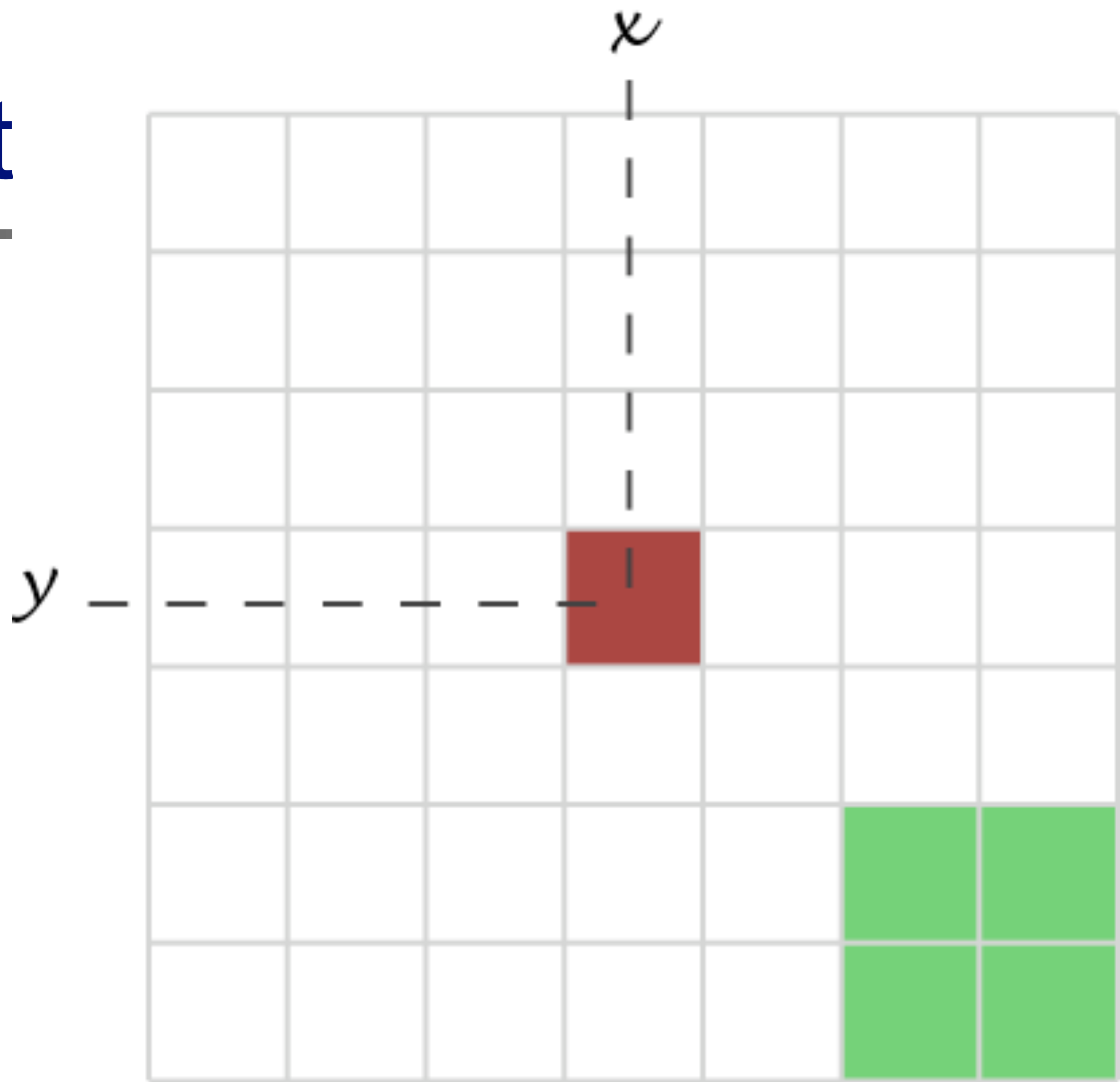
1x3



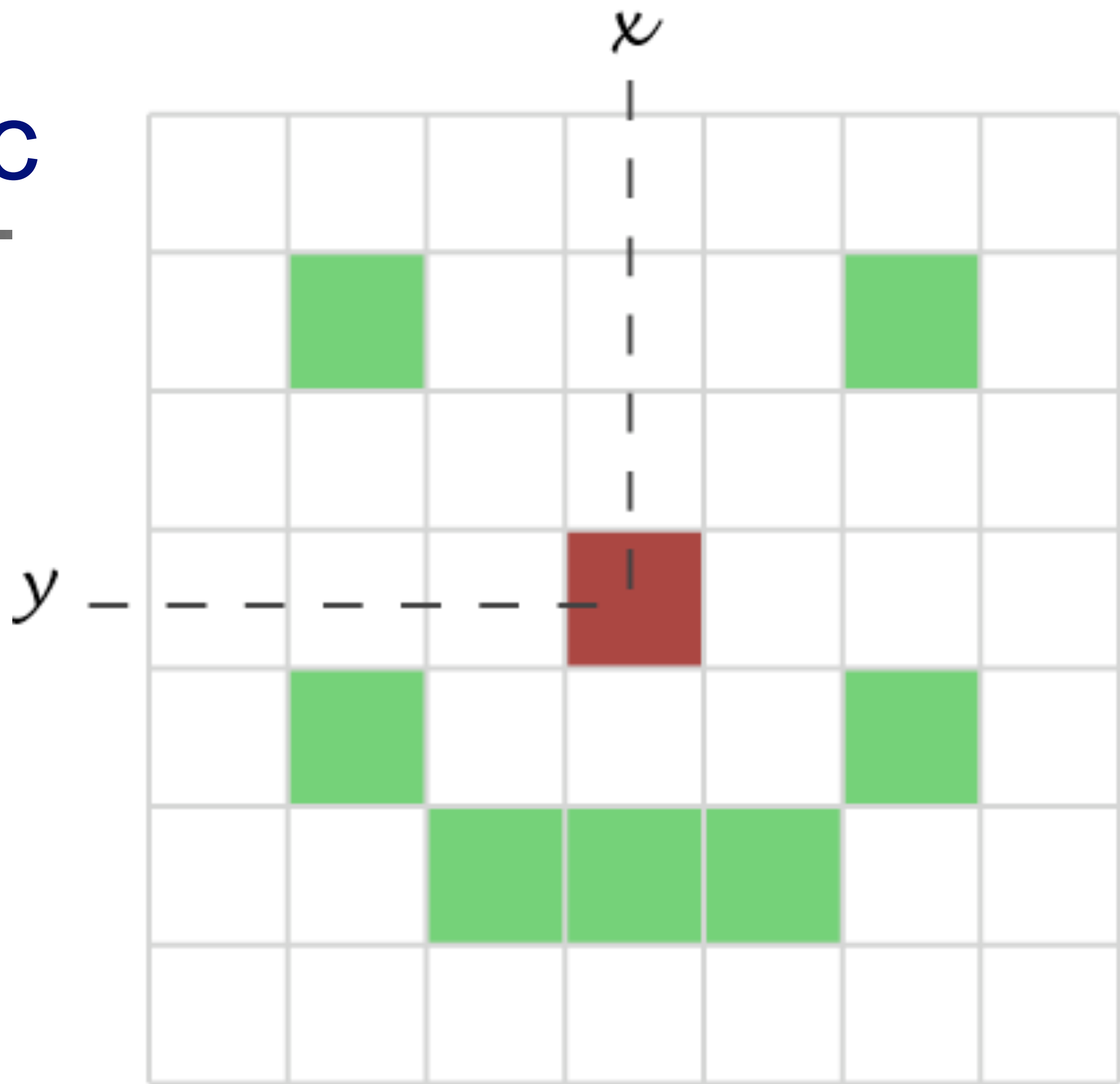
1x5



2x2 shift



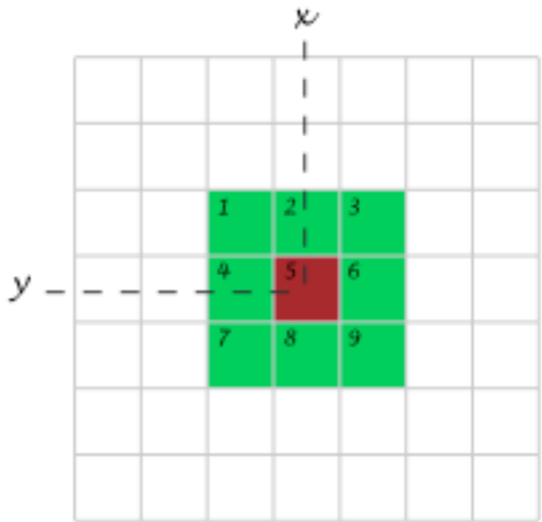
Misc



B. The mean filter

Simplest filter: the value of a pixel is replaced by the intensity mean computed over neighbors pixels

$$a_i^* = \frac{1}{N_\Omega} \sum_{j \in \Omega} a_j$$



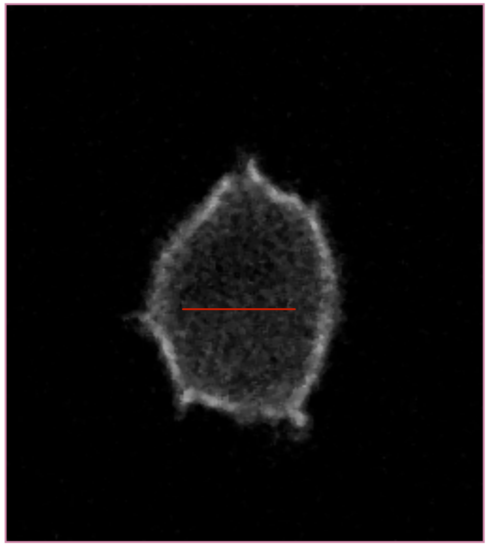
3x3 example:

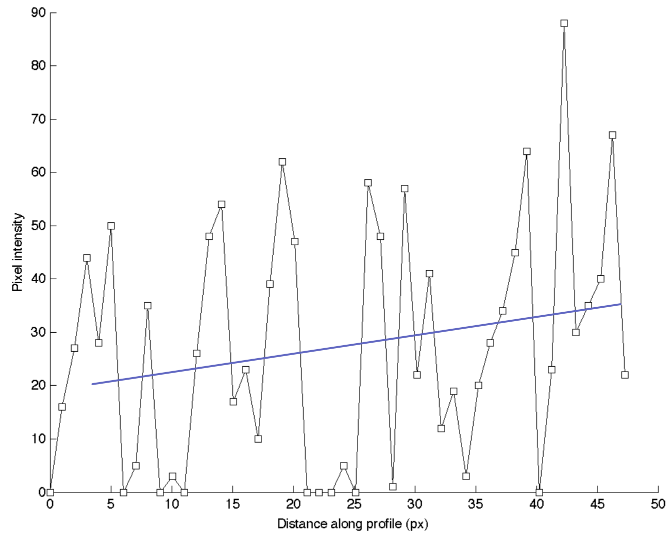
$$a_i^* = \frac{1}{9}\left(a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9\right)$$

The mean filter

what is it good for?

Noise removal - typically Gaussian noise.





(typ. Appears for strong labeling, short exposure time)

The mean filter

properties

Main property: low-pass filter

- kernel size influence
- number of successive applications

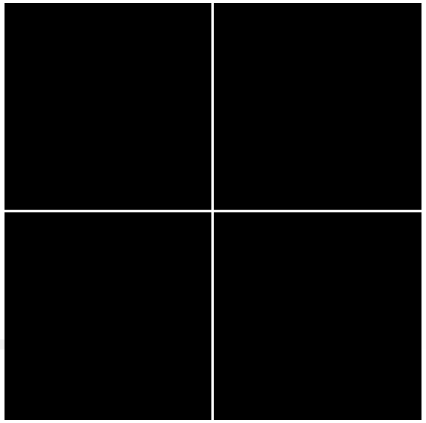
Cases where it fails

- salt & pepper noise

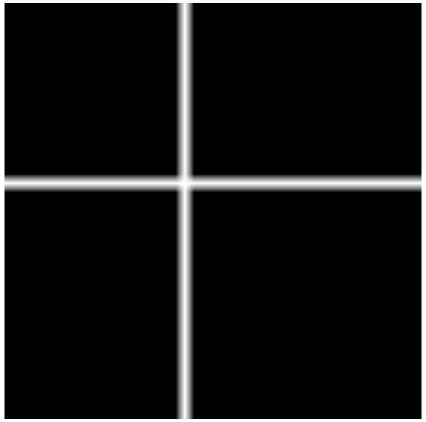
practical

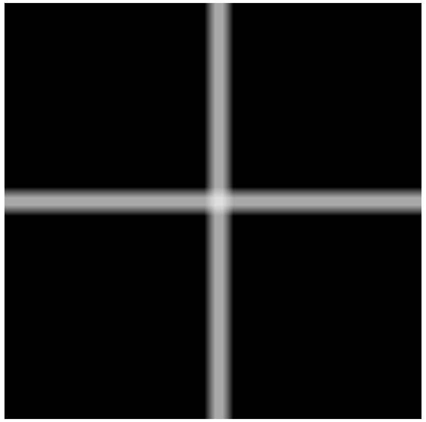
The mean filter

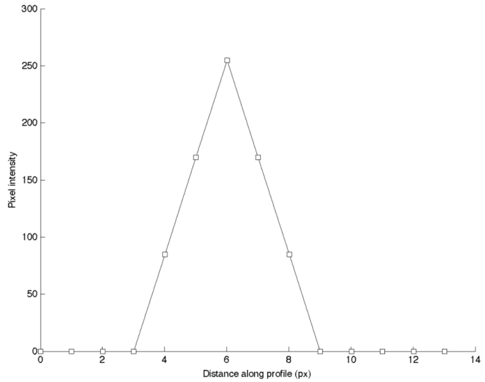
properties - kernel size

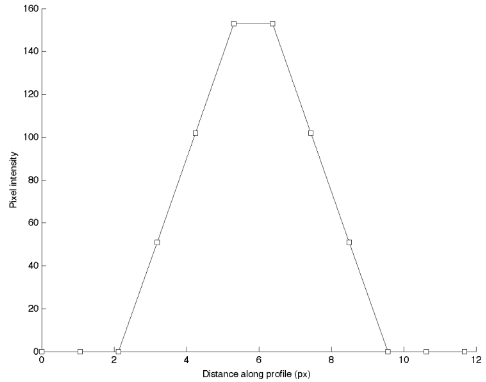


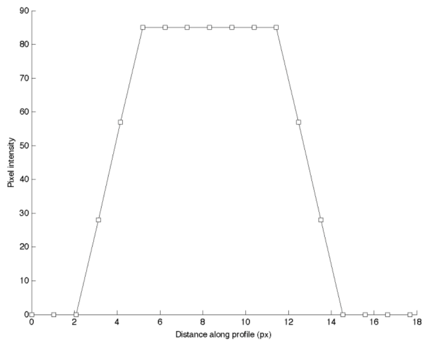
Zoom 10x





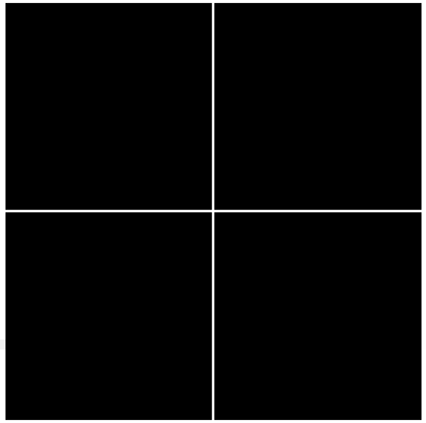




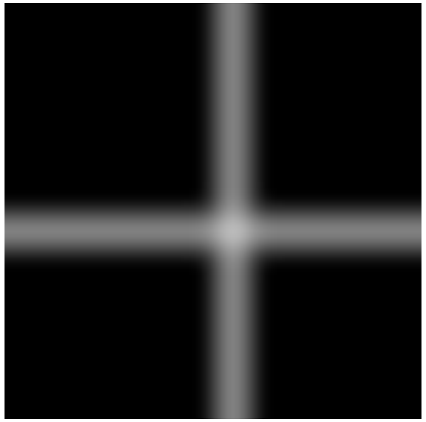


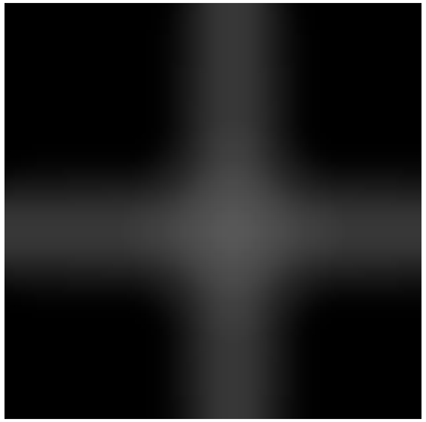
The mean filter

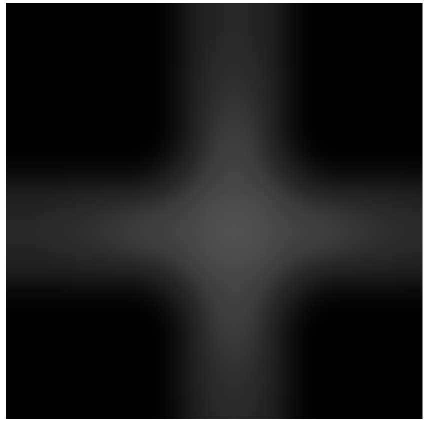
properties - number of application

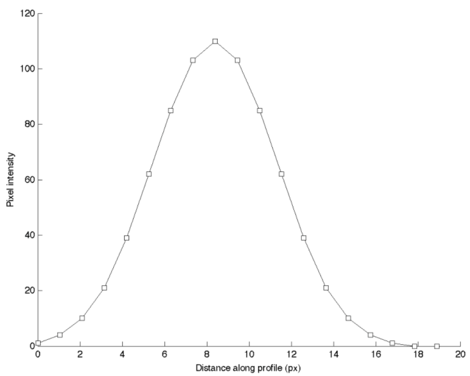


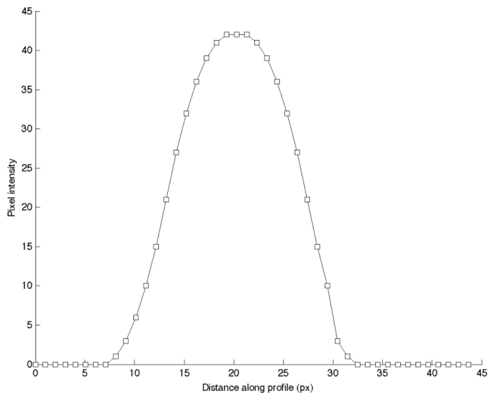
Zoom 10x

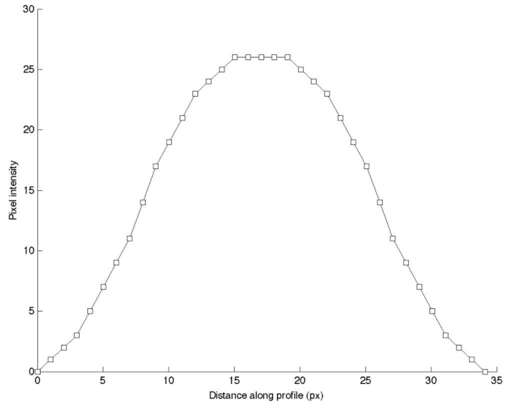












The mean filter

properties - linear filtering

The mean filter is a <u>linear filter</u>:

"*The new pixel value depend on a linear combination of neighbor pixel values*"

$$i^{*}_{x,y} = \sum_{i,j \in \Omega} \alpha_{ij} \; i_{x+i,y+j}$$

Neighbor pixels

Filter coefficients

$\alpha_{1,1}$	$\alpha_{1,2}$	$\alpha_{1,3}$
$\alpha_{2,1}$	$\alpha_{2,2}$	$\alpha_{2,3}$
$\alpha_{3,1}$	$\alpha_{3,2}$	$\alpha_{3,3}$

➔ another notation for 3x3 kernel

The mean filter

properties - linear filtering 2

Lot of properties:

- Suppose one image is made of a combination of images (*e.g.* $I = a_1 \times I_1 + a_2 \times I_2$, like in color combine or contrast change) ➔ applying a linear filter to it is the same as applying it to all <u>parts</u>, then making the combination.

- When applying a succession of linear filters ➔ <u>order</u> in which linear filters are applied does not matter.

- Mathematical framework underlying it ➔ <u>convolution</u> (see later).

The mean filter

summary

- simplest filter - fast
- is a linear filter
- average noise, does not eliminate it
- good against Gaussian noise
- blur images - small details are lost in the average
- smooth edges dramatically
- trivial convergence

Low-pass filter

C. The median filter

The value of a pixel is replaced by the *median* of the pixel intensity in neighbors pixels

Take neighborhood (*e.g.* 3x3)

5	112	86
235	88	211
137	233	108

Sort it

5
86
88
108
112
137
211
233
235

Take median

112 → 112

The median filter

noise elimination

Original:

5	9	6	6	9	5	9
9	5	9	7	8	7	9
8	9	8	6	7	9	9
9	9	7	200	9	6	9
6	5	8	6	9	6	7
9	7	9	9	8	6	7
7	9	5	6	7	6	6

outlier

Median filtered:

0	5	6	6	6	7	0
5	8	7	7	7	9	7
8	9	8	8	7	9	7
6	8	8	8	7	9	6
6	8	8	9	8	7	6
6	7	7	8	6	7	6
0	7	6	6	6	6	0

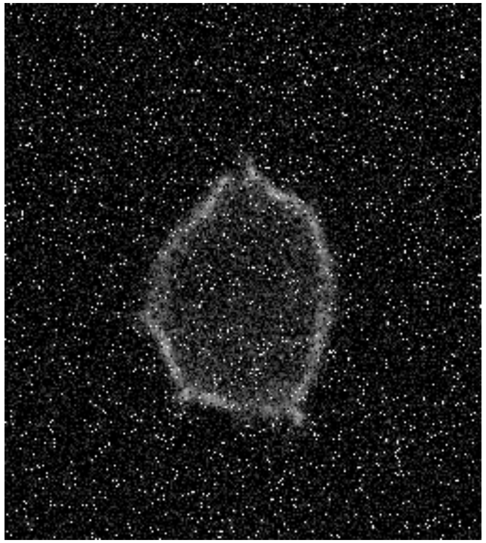
The outlier value has completely been removed from the dataset

The median filter

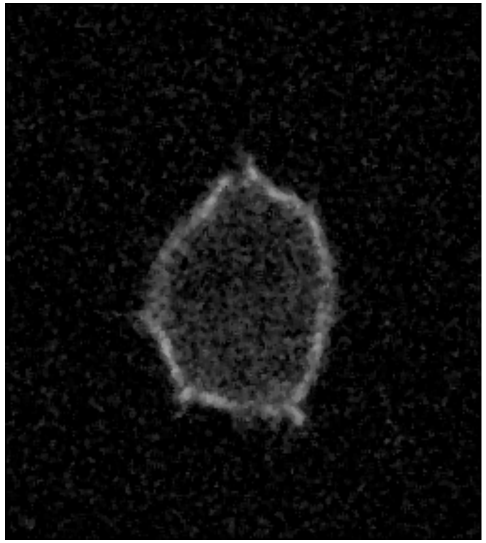
what is it good for?

"Salt & pepper" noise removal

Original:



Median filtered:



(typ. Appears for very weak labeling)

The median filter

properties

- Typically good for “Salt & pepper” noise removal
- *Eliminates* noise
- Slower than mean and similar (not such a problem anymore)
- Not linear
- Edge-preserving
- Non-trivial convergence

D. The Sobel filter

Combination of 2 filters:

$$g_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad g_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$\text{output} = \sqrt{{g_x}^2 + {g_y}^2}$$

The Sobel filter

What does it do? (on g_x only)

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

Uniform image:

1	1	1	[illegible]	[illegible]
1	1	[illegible]	[illegible]	[illegible]
1	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	1	1	1

$\rightarrow$ 0

Edge:

1	1	1	0	0
1	1	1	0	0
1	1	$\boxed{1}$	0	0
1	1	1	0	0
1	1	1	0	0

$\rightarrow$ 4

Edge detector!

The Sobel filter
edge detector

g_x detects edges in the X direction
g_y detects edges in the Y direction
The output is ~ the <u>edge gradient magnitude</u>

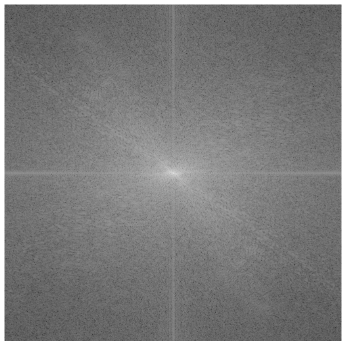
$$\text{output} = \sqrt{g_x^{\,2} + g_y^{\,2}}$$

Relax…

II. Image processing in the Fourier domain

Before:

After:



- A. The Fourier transform
  - Introduction
  - Examples and measurements
- B. The *inverse* Fourier transform
  - Principles
  - Filtering

A. The Fourier transform

- The Fourier transform is a way to obtain a new *representation* of the data.
- It is best suited for data with *repetitive patterns* and highlights these patterns.
- It's mathematical definition (1D, discrete) is:

$$\hat{I}_k = \sum_n I_n \mathrm{e}^{-2i\pi k/n}$$

(we don't have to care too much)

- It is best explained with a music player.

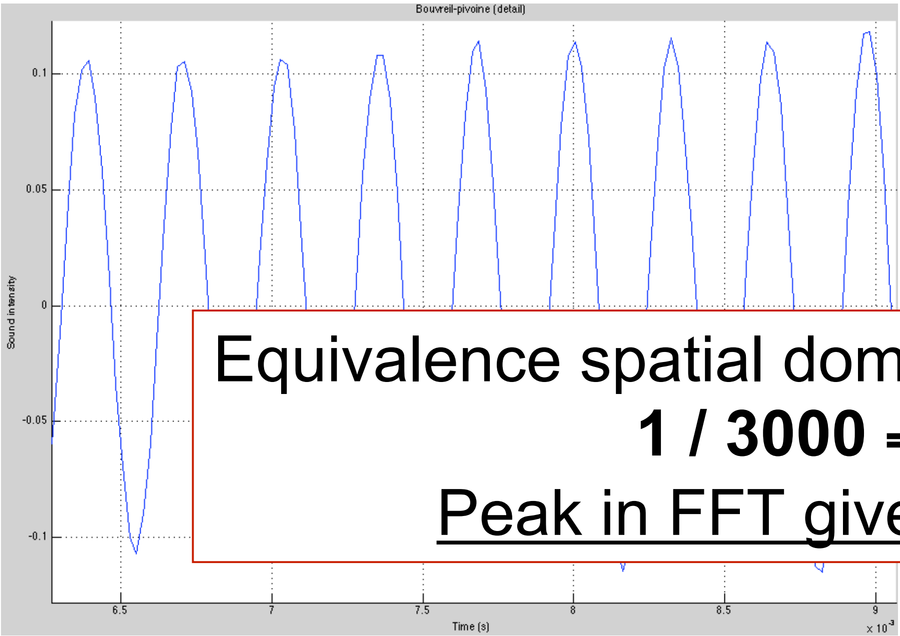
The Fourier transform
the Bouvreuil-Pivoine

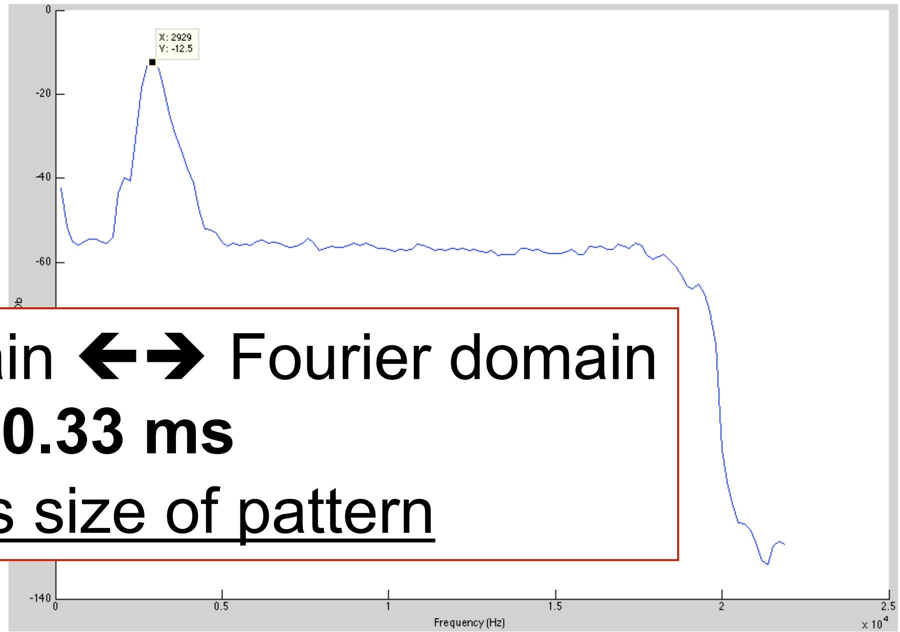
Bird song.

Delay between peaks:
~ 0.35 ms

FFT of this looks like:

Peak in FFT:
~ 3 kHz



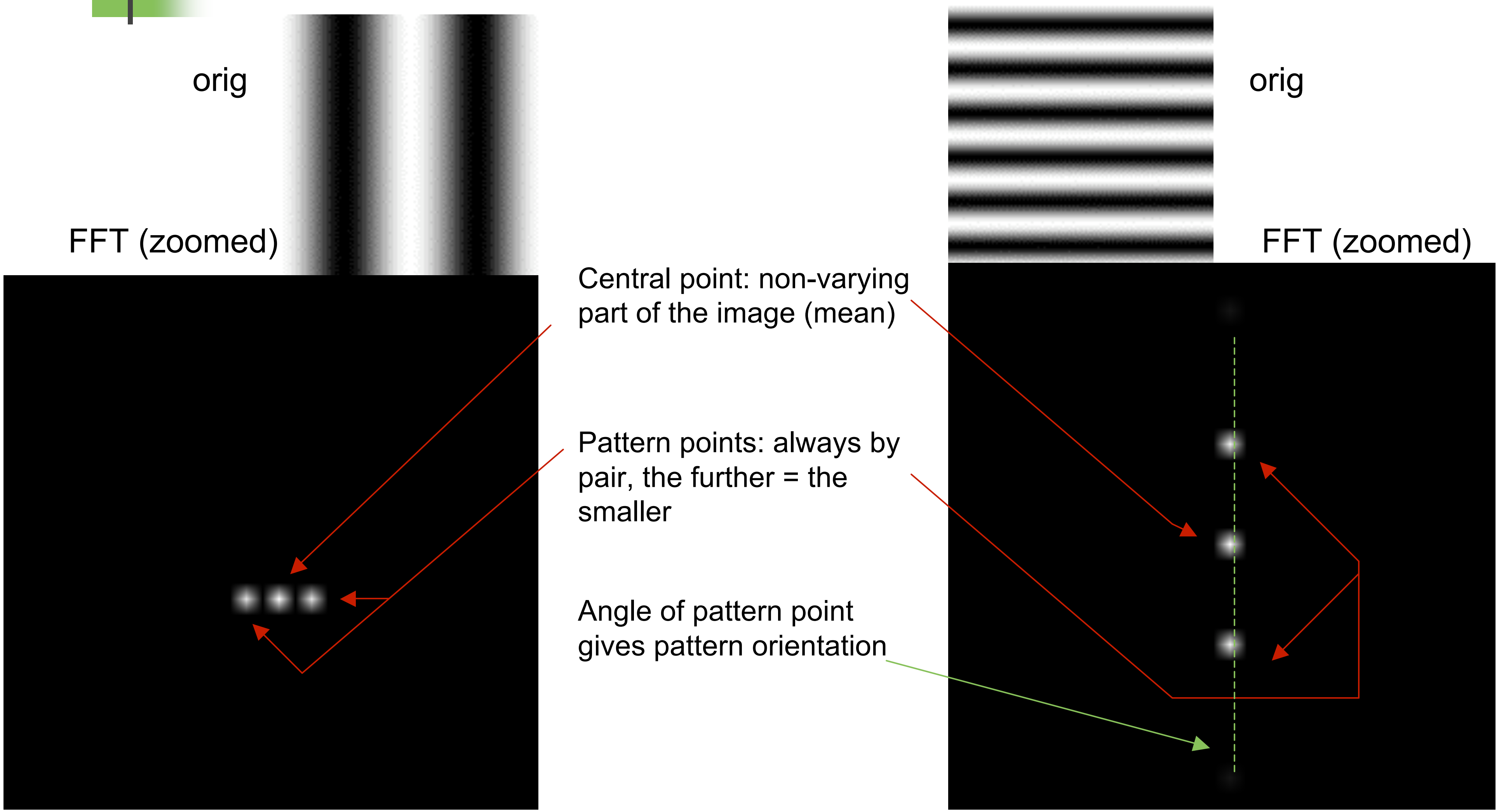


Equivalence spatial domain ←→ Fourier domain

1 / 3000 = 0.33 ms

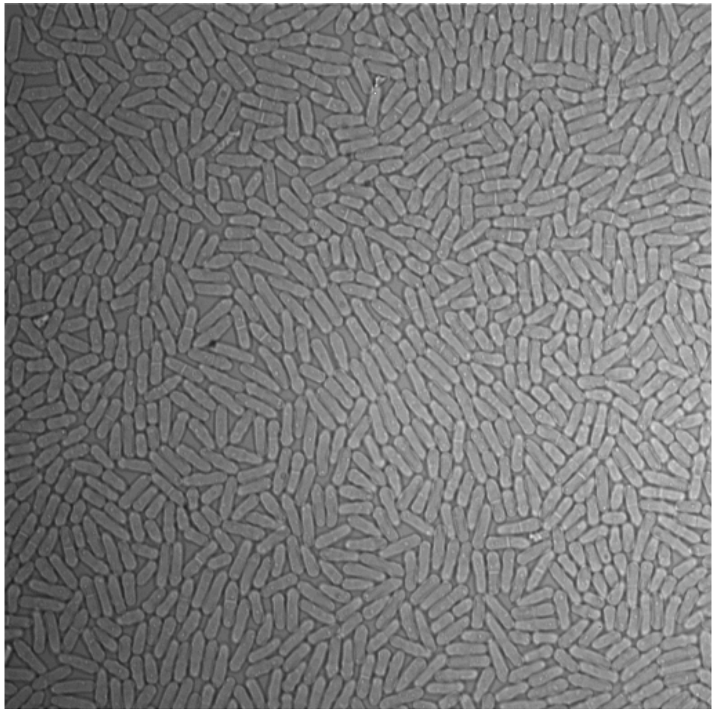
<u>Peak in FFT gives size of pattern</u>

The Fourier transform
in 2D (images)

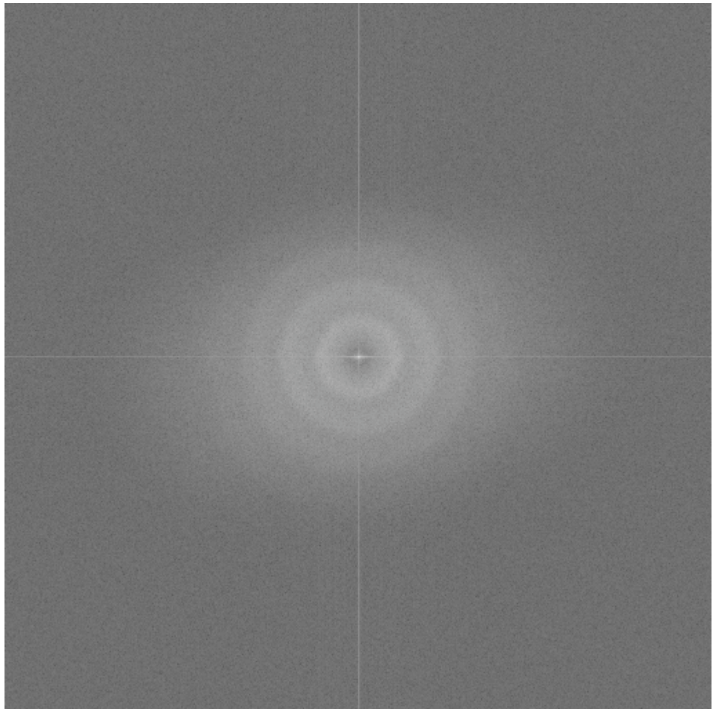


The Fourier transform
real images

… are rarely that clear



S pombe cells (*Tolic lab*)



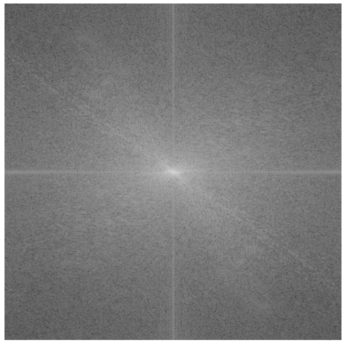
FFT

B. The *inverse* Fourier transform

Because the Fourier image and the real image contain the same amount of info, it is possible to generate a real image from its Fourier representation:

Before:

After:

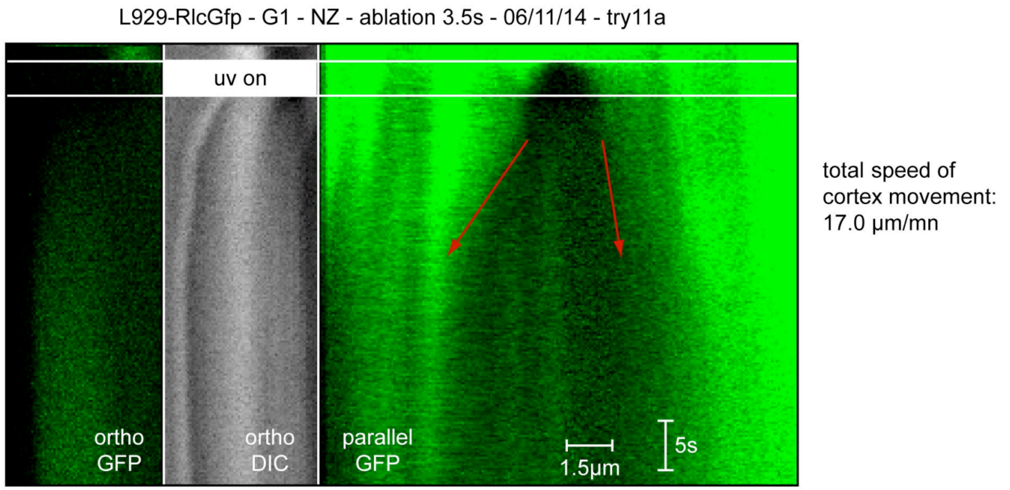


Changed her mind:

Relax…

III. Time domain

Dealing with multiple images files (a.k.a. *stacks*):
timelapse movies, 3D stacks, …



- Intensity over time
- Kymographs

Motion blur

Motion blur = 1x10 neighborhood + average