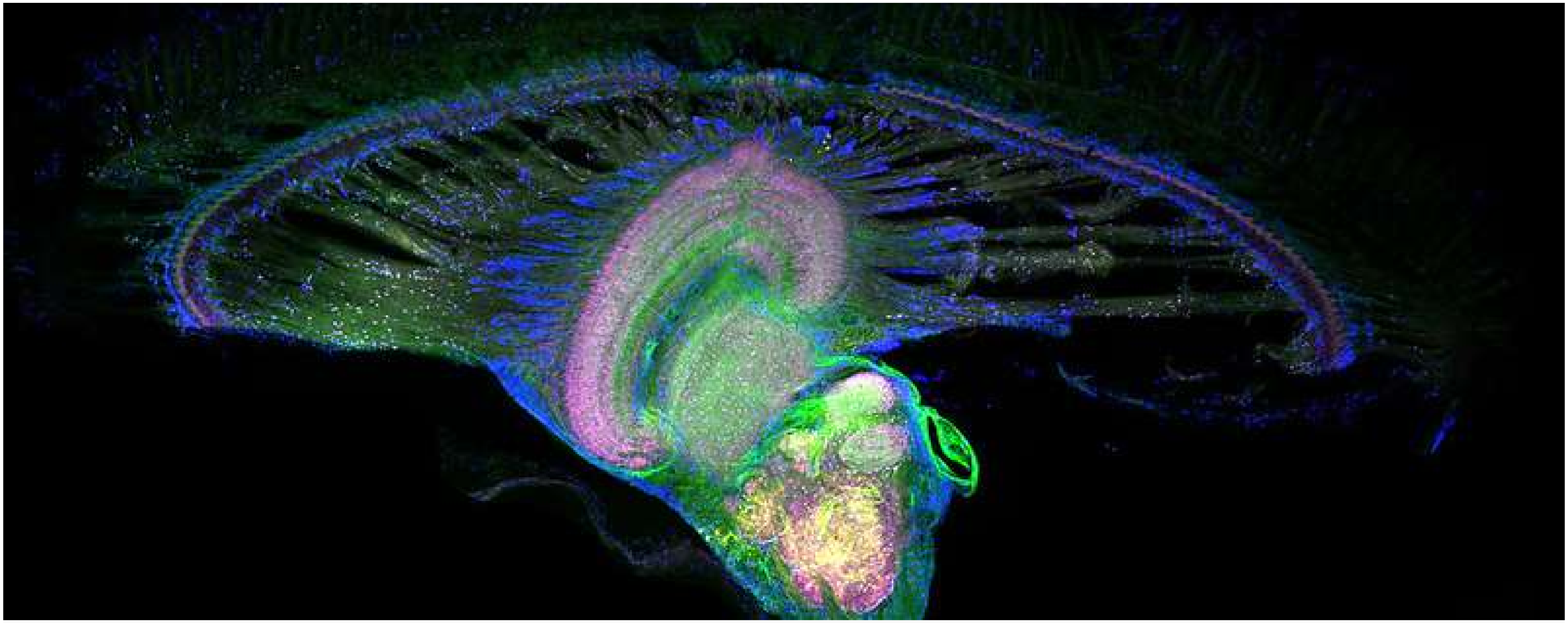


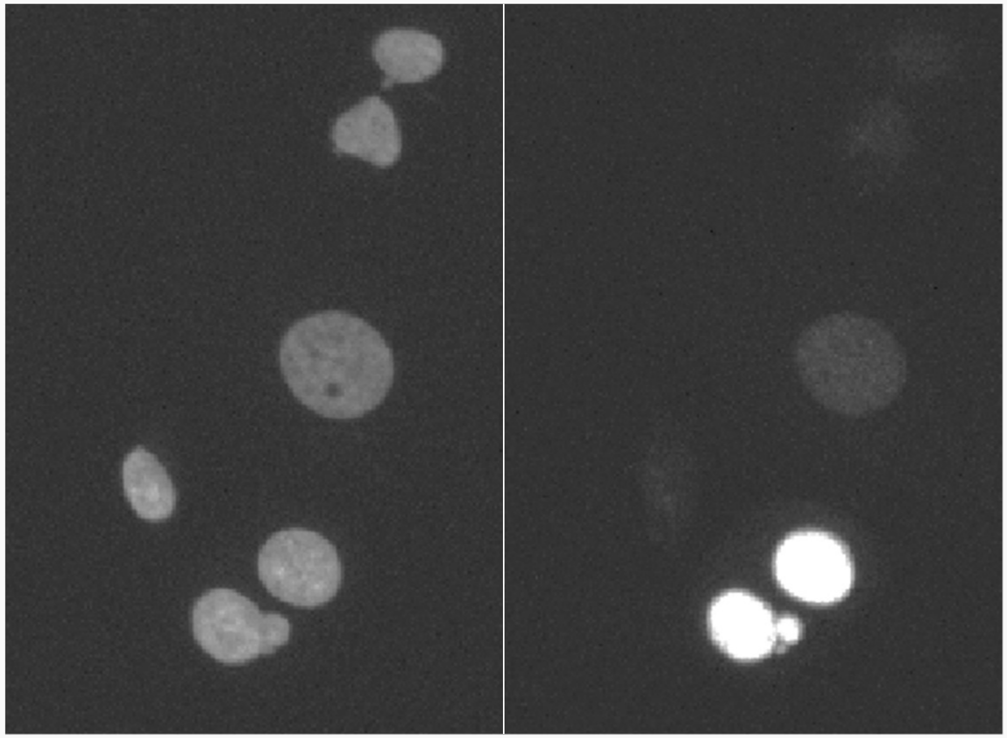
MATLAB lecture 7: Image intensity measurements and corrections

MCDB/BCHM 4312/5312

Homework: Hoechst or pRB?

Hoechst

Nuclear marker



pRB

Cell division marker

- Rule of thumb: don't segment on the same channel you're trying to measure
- Why? Because if the signal changes, you might end up missing some cells

Learning goals

- Image data types revisited

- Common sources of intensity noise and how to address
  - Uneven contrast
  - Salt-and-pepper

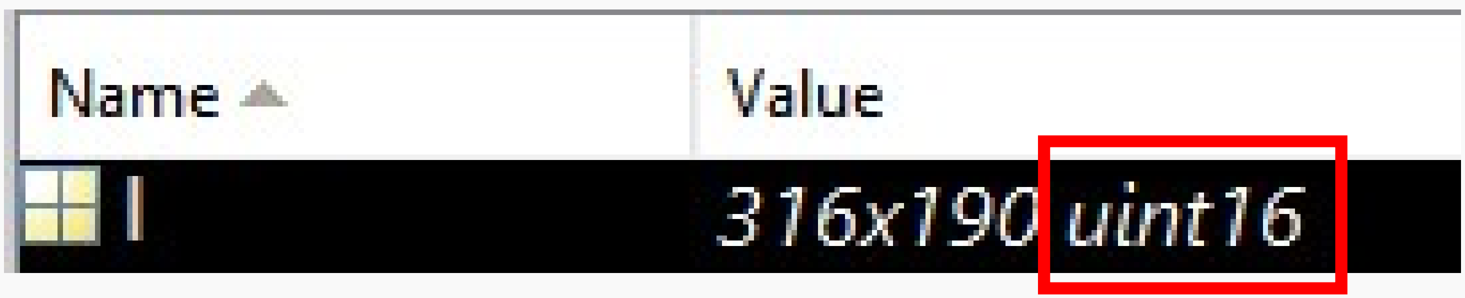
- Adjusting image brightness and contrast

Image data types

- All information in a computer is stored as bits

1 0

- The data type of a variable describes how the bits are interpreted by the computer



- Two main data types: integers and floating-point numbers (doubles)

Integers

0	1	0	0	0	0	0	1

What is the value of the unsigned integer above?

(A) 20

(B) 131

(C) 65

(D) 1000001

Integers

0	1	0	0	0	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$= 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + \ldots$$

$$+ 0 \times 2^1 + 1 \times 2^0$$

$$= 64 + 1$$

$$= 65$$

Properties of integers

Which of the following is true?

(A) Integers can only represent whole numbers (no decimals)

(B) Unsigned integers, can only represent positive numbers and zero

(C) Largest unsigned number is (2^N – 1) where N = number of bits

(D) All the above

Properties of integers

Which of the following is true?

(A) Integers can only represent whole numbers (no decimals)

(B) Unsigned integers, can only represent positive numbers and zero

(C) Largest unsigned number is (2^N – 1) where N = number of bits

(D) All the above

Integers

0	1	0	0	0	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$= 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + \ldots$$

$$+ 0 \times 2^1 + 1 \times 2^0$$

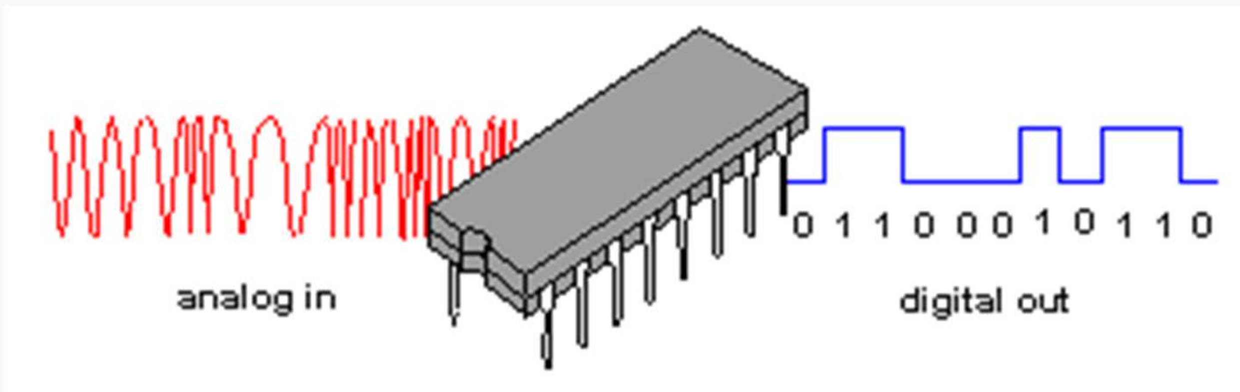
$$= 64 + 1$$

$$= 65$$

Images are usually saved as integers

- Cameras have a component called an Analog-to-Digital converter (A/D converter or ADC)

- Converts the signal from the detector (e.g. PMT, CCD arrays) which is analog voltage into a digital number

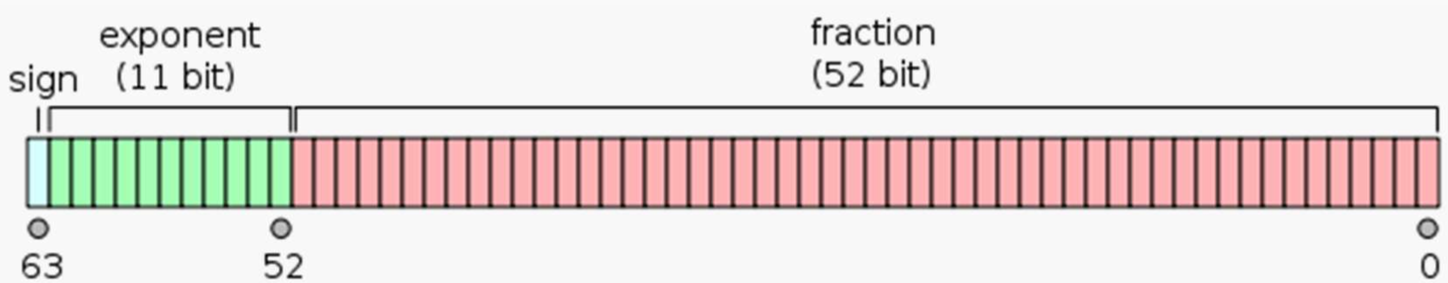


- Current gen microscopes typically record images as uint16

Floating point numbers

- Floating point numbers can have both signs and decimal places

- Decimal numbers are represented by defining some number of bits to represent the exponent



$$3.1421 = +31421 \times 10^{-5}$$

Exponent

Sign Fraction

- In MATLAB, the default number type is a **double**, which is a floating point number
  - A double is short for double-precision or 64-bits

Why is this important?

- Be careful when doing operations with integer numbers

- For example:

```
>> uint8(5)/2
 ans = 3

>> uint8(5) * 100
 ans = 255

>> uint8(5) - 10
 ans = 0
```

Converting between integer and double

- Avoid potential issues by converting integer images into double format **before carrying out mathematical operations**

```
>> I = imread(‘cameraman.tif’);
>> I_double = double(I);
```

- Some functions will behave differently depending on whether the image data is a double or an unsigned integer

CHECK THE DOCUMENTATION FOR DETAILS

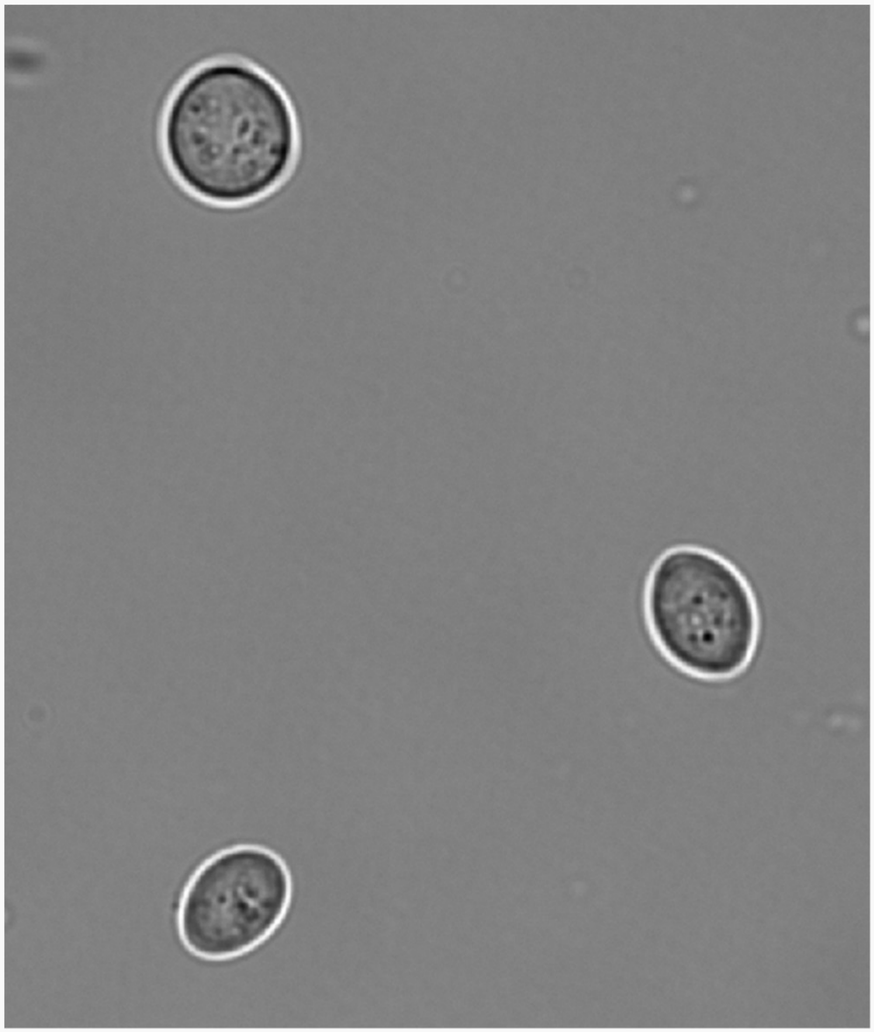
Low intensity image corrections

- As you saw in the last homework, segmentation is much easier when there is a bright marker that is in every cell

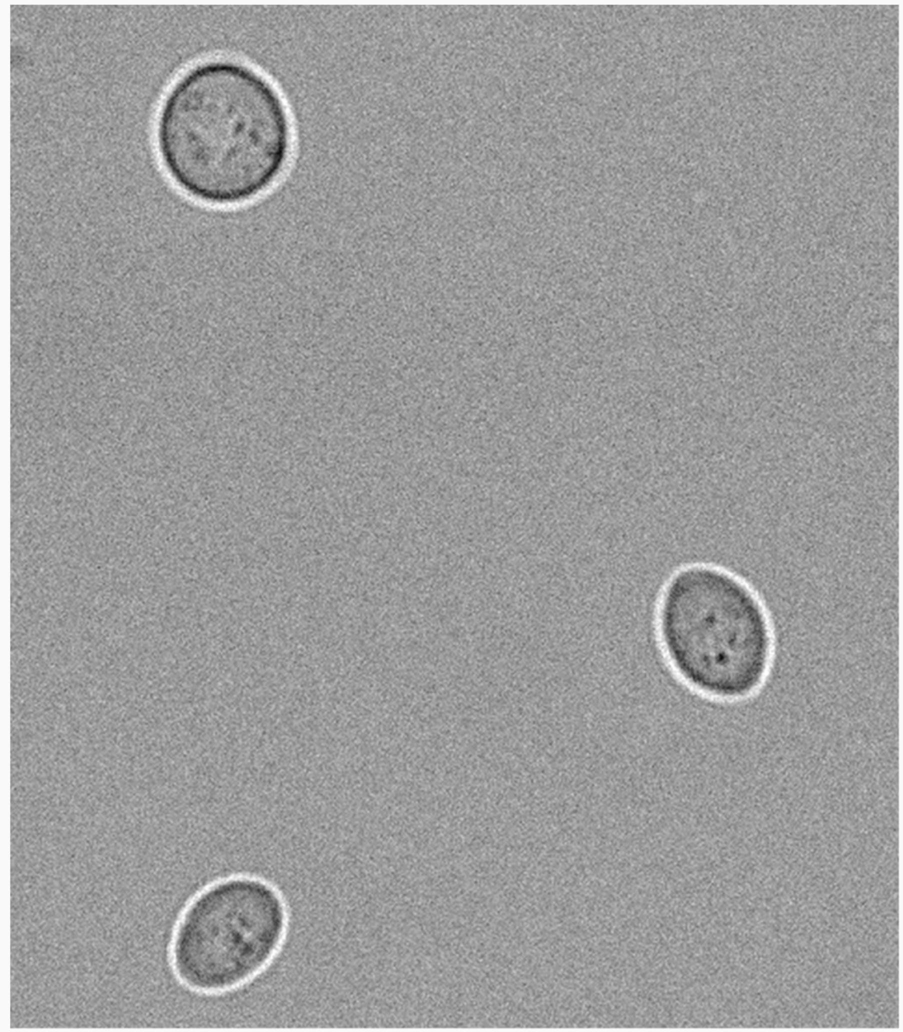
- However, sometimes things do not go as planned and images are dim:
  - Poor dye permeability
  - Poor antibody labelling
  - Low expression of fluorescent protein
  - Or mistakes during imaging – this is pretty common

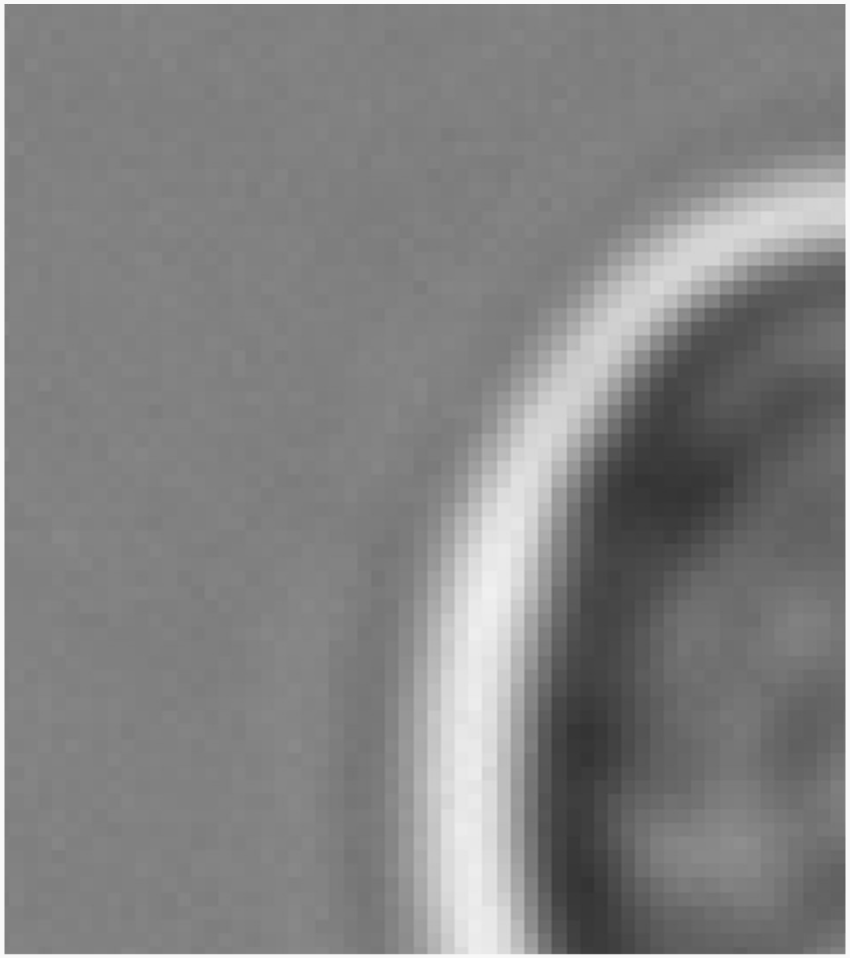
Which of these images has the lower intensity?

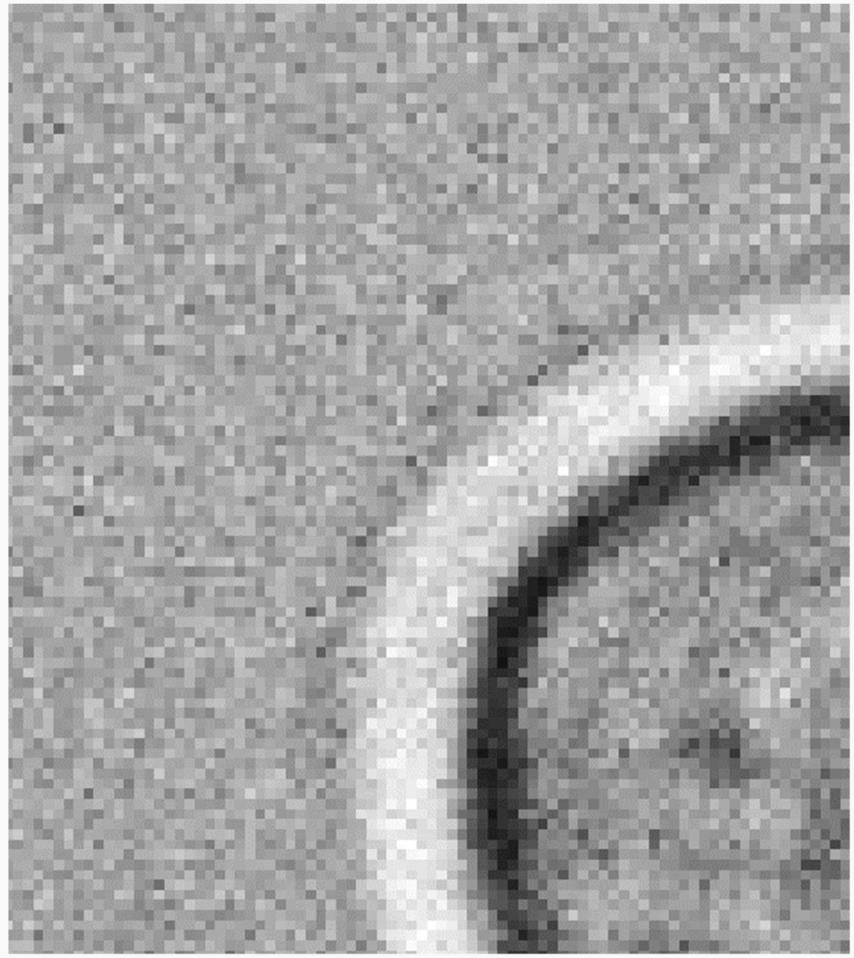
A



B







Noise in images

- Dim images typically have a speckle pattern
- The pattern is mostly formed by inherent noise from the detector or the light source
  - Shot noise
  - Read noise
  - Digitization noise

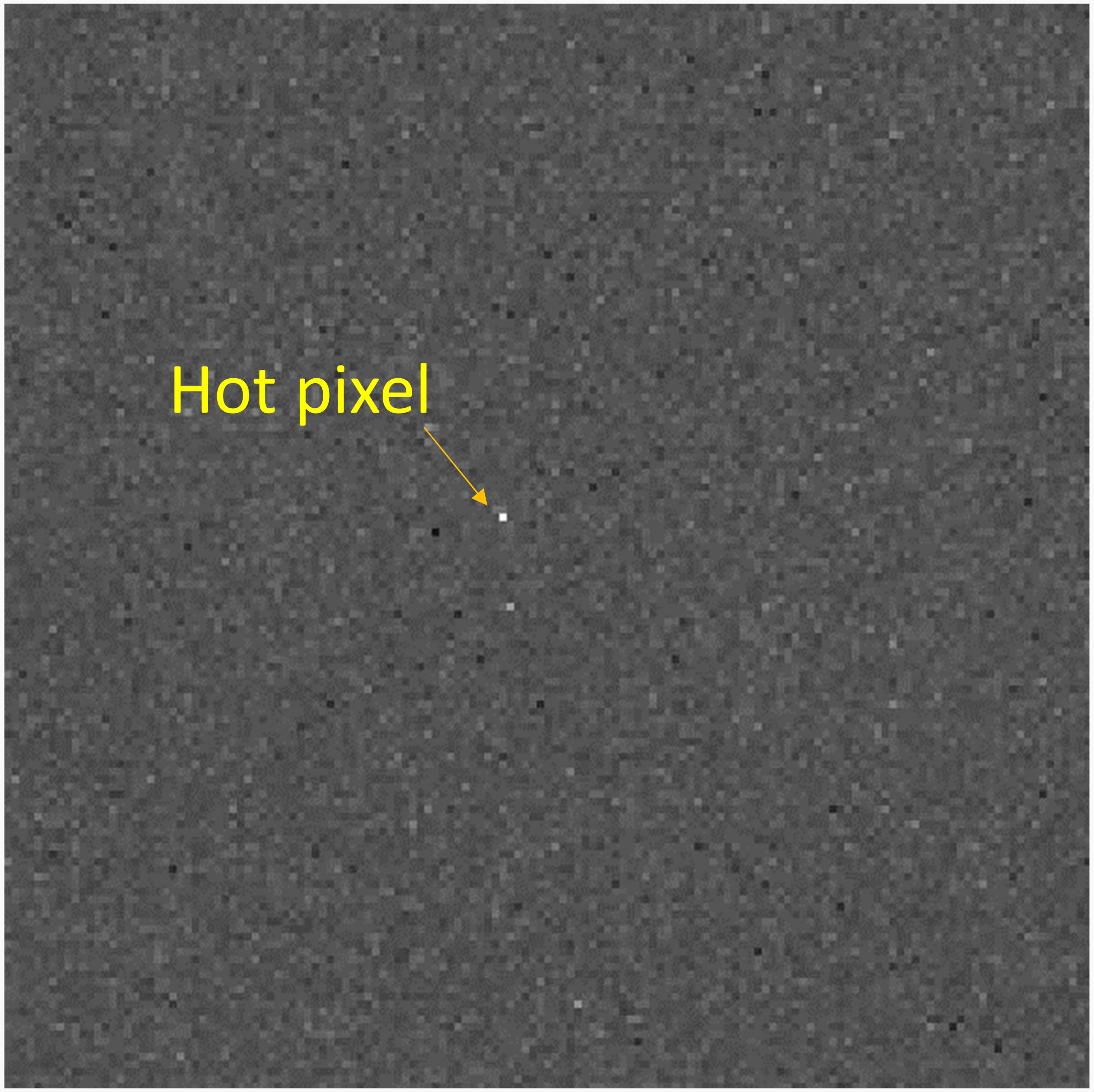
Random noise

Intensity changes randomly over time

Joe will cover these in detail next week

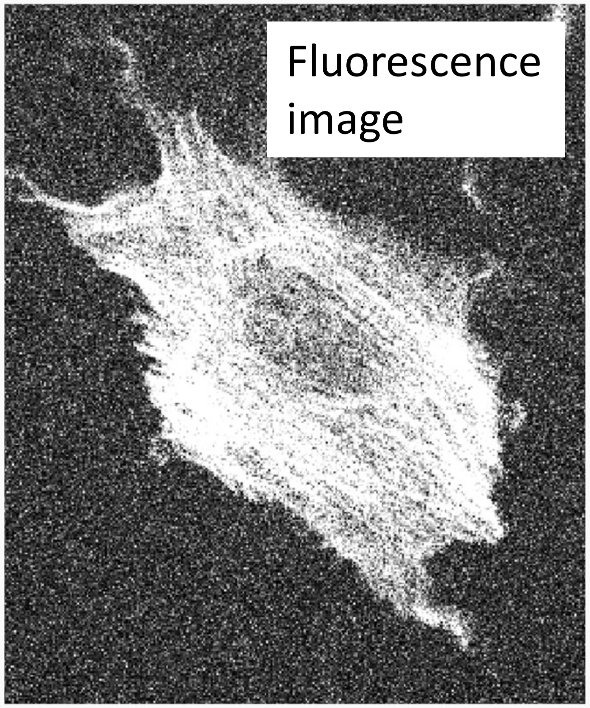
  - Hot pixels – pixels which are always bright
- Noise is typically a single pixel in "size"

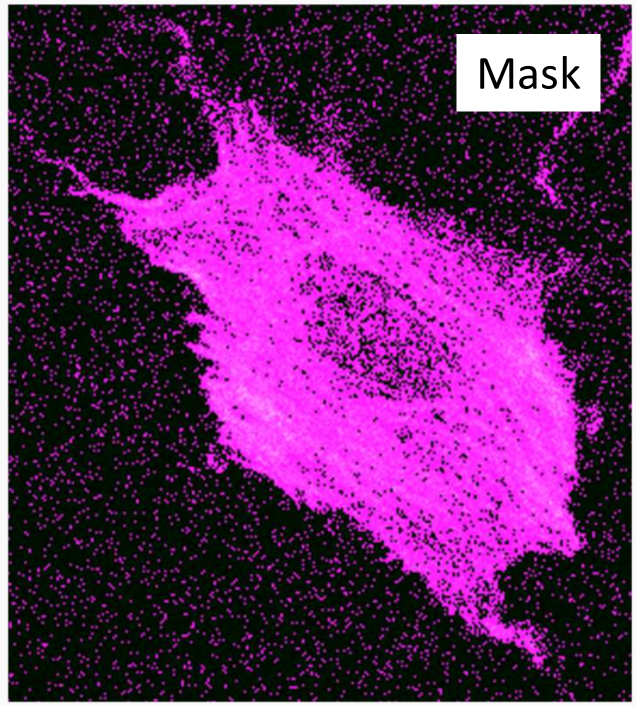
Example of noise



Why get rid of noise?

- Noise will show up in segmentation

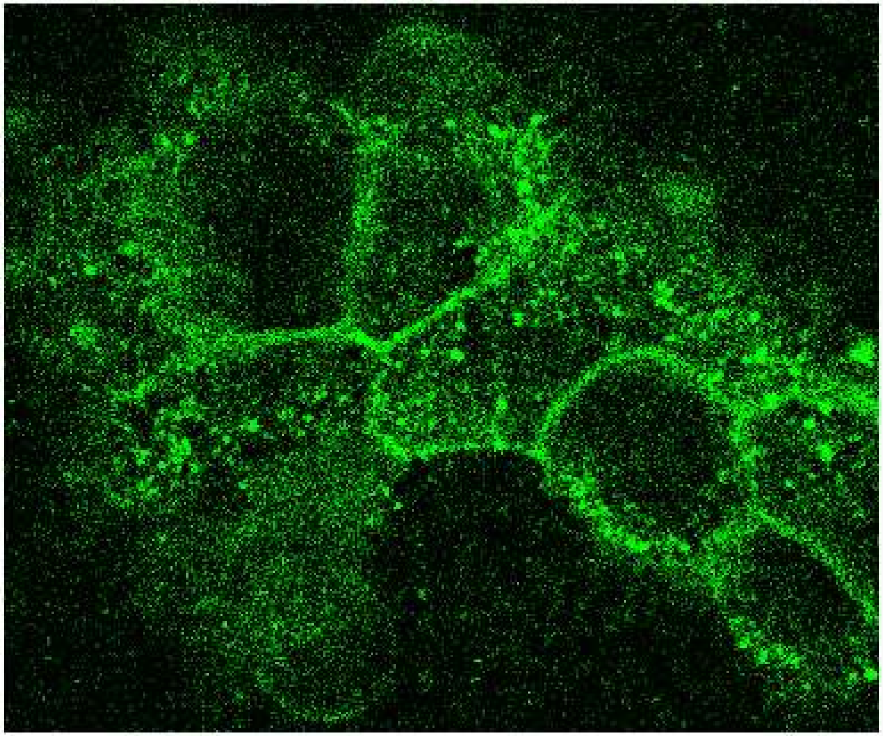




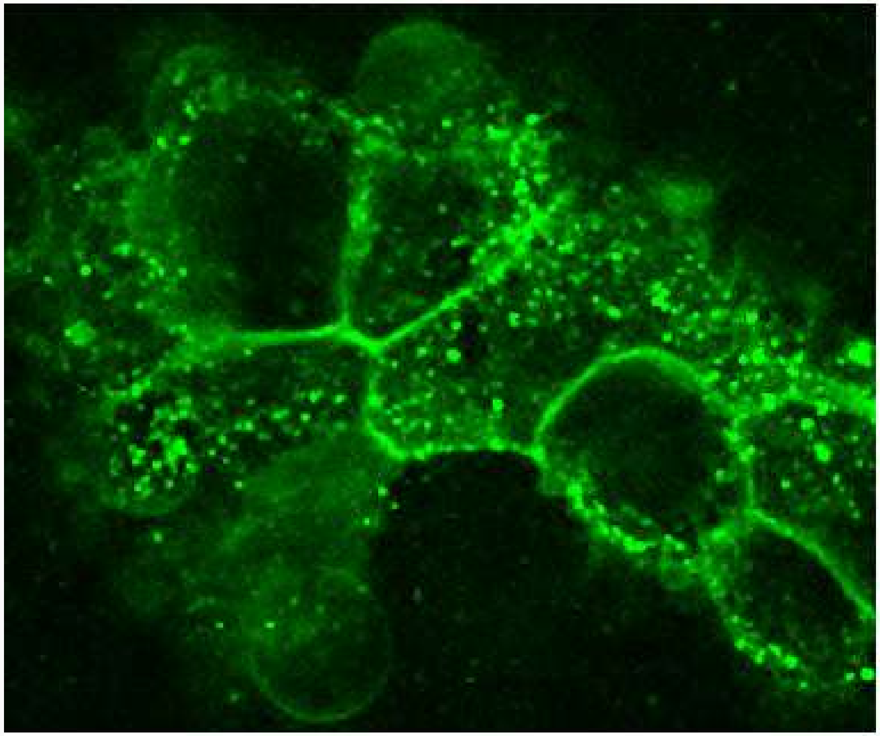
- Obscures smaller details – i.e. sub-cellular structures

Noisy image

Trying to identify puncta (spots)



"Denoised" image



You can only clean up an image if the structures are > 2 pixels

Why? Random noise is typically 1 pixel in size

http://bigwww.epfl.ch/research/projects/denoising.html

Image filters

- Image filters are digital operations which are applied to the image data

- Two main filters:
  - Gaussian filter
  - Median filter

- These are not physical filters (e.g. fluorescence filters). They are mathematical functions which transform the image.

Gaussian filter

Which of the following equations describes a Gaussian?

(A) $y = mx + c$

(B) $f(r) = A \exp(-r^2/w^2)$

(C) $x^2 + y^2 = r^2$

(D) $y = Ax^3 + Bx^2 + Cx$

Gaussian filter

Which of the following equations describes a Gaussian?

(A) $y = mx + c$

(B) $f(r) = A\exp(-r^2/w^2)$

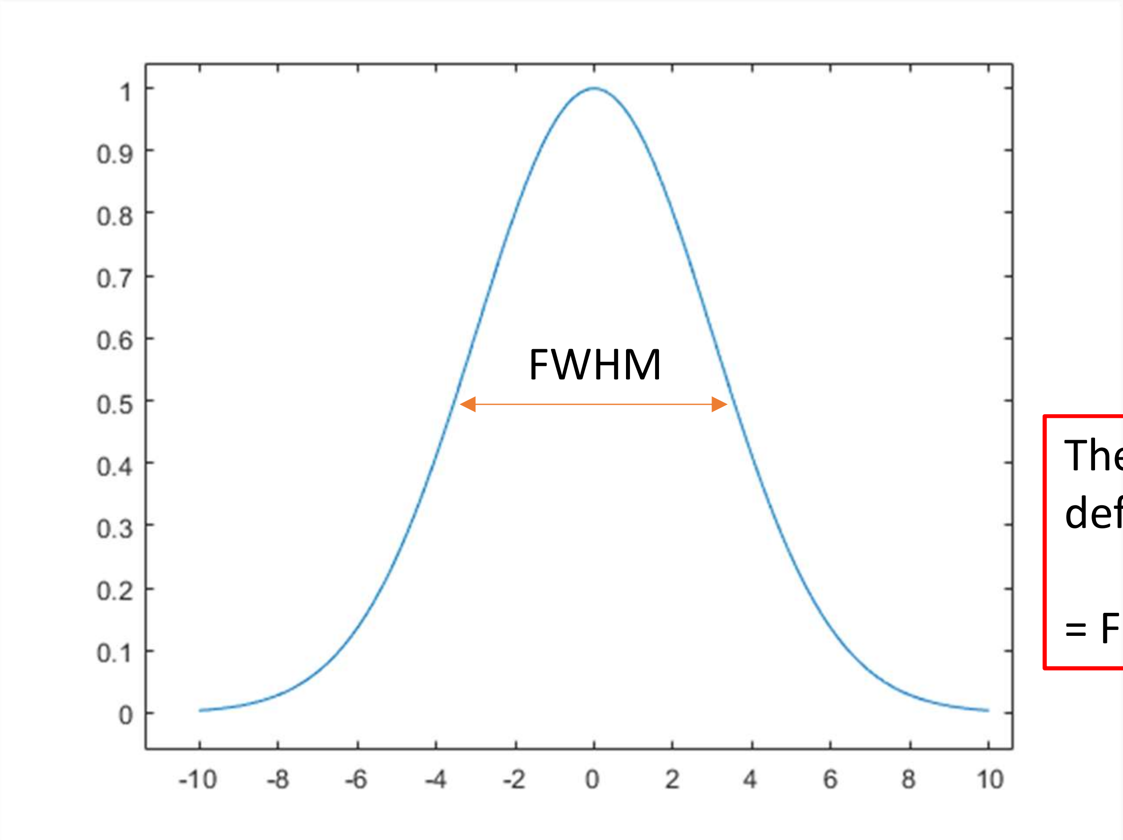
(C) $x^2 + y^2 = r^2$

(D) $y = Ax^3 + Bx^2 + Cx$

Gaussian equation

1-D Gaussian equation

$$y = A\exp(-(x - x0)^2/2\sigma^2)$$



A = amplitude

x0 = x-offset

σ = width

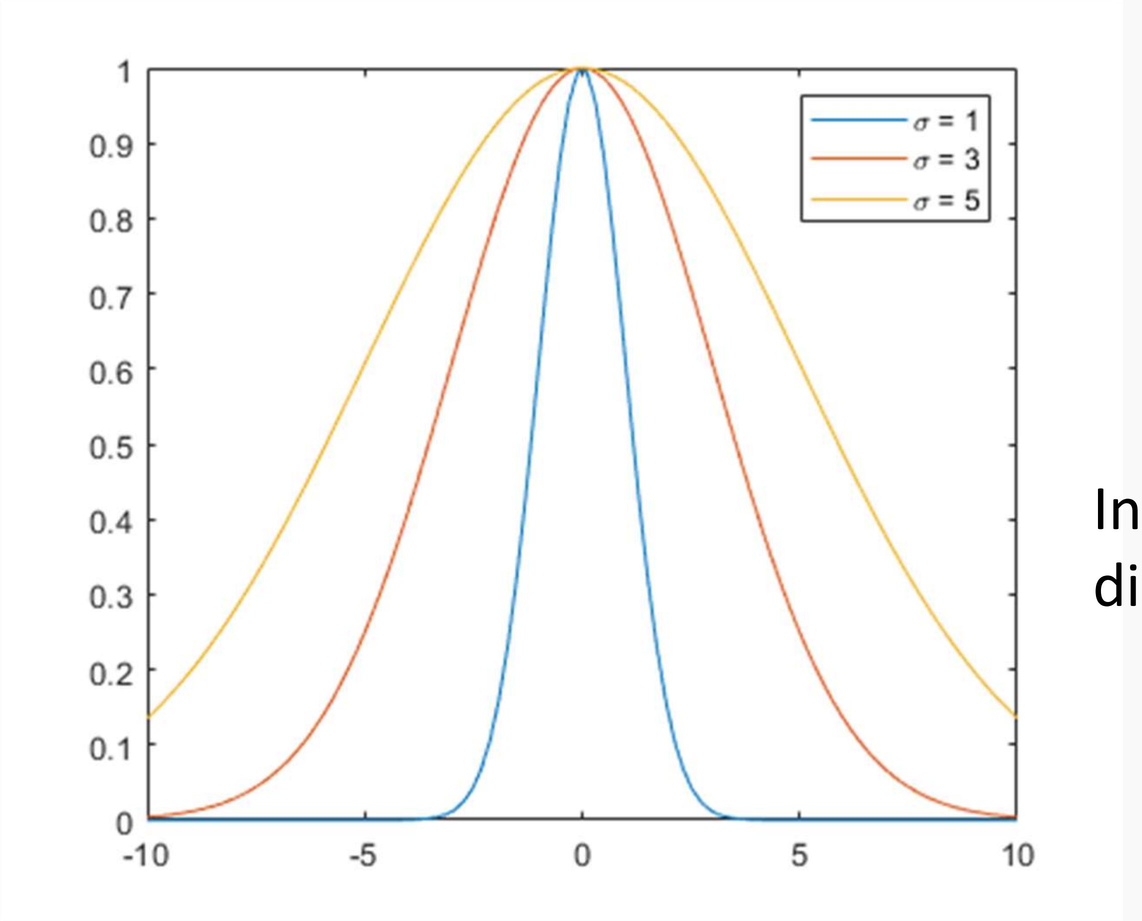
The width of a Gaussian is commonly defined at 0.5A

= Full width at half max (FWHM)

Gaussian equation

1-D Gaussian equation

$$y = A\exp(-(x - x0)^2/2\sigma^2)$$



A = amplitude

x0 = x-offset

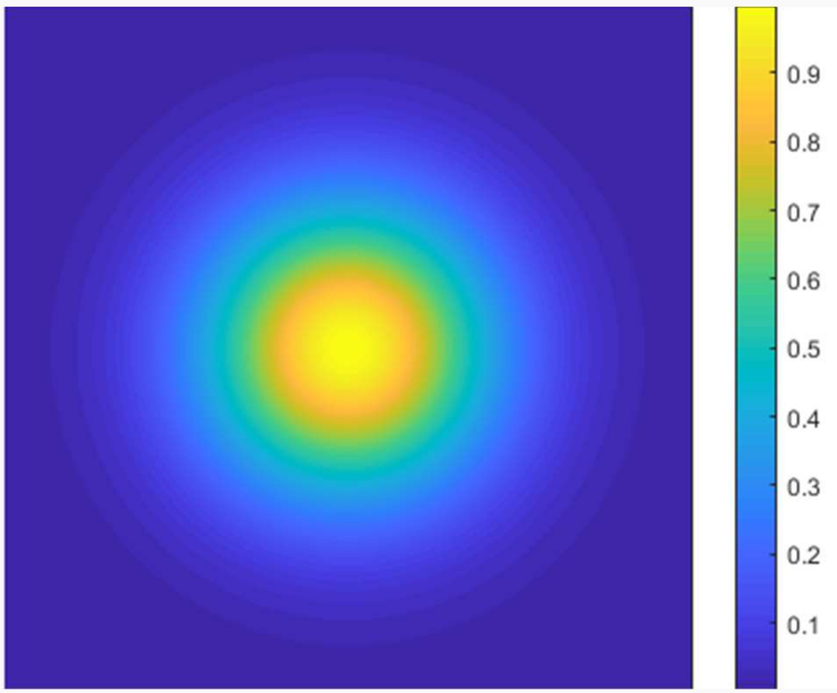
σ = width

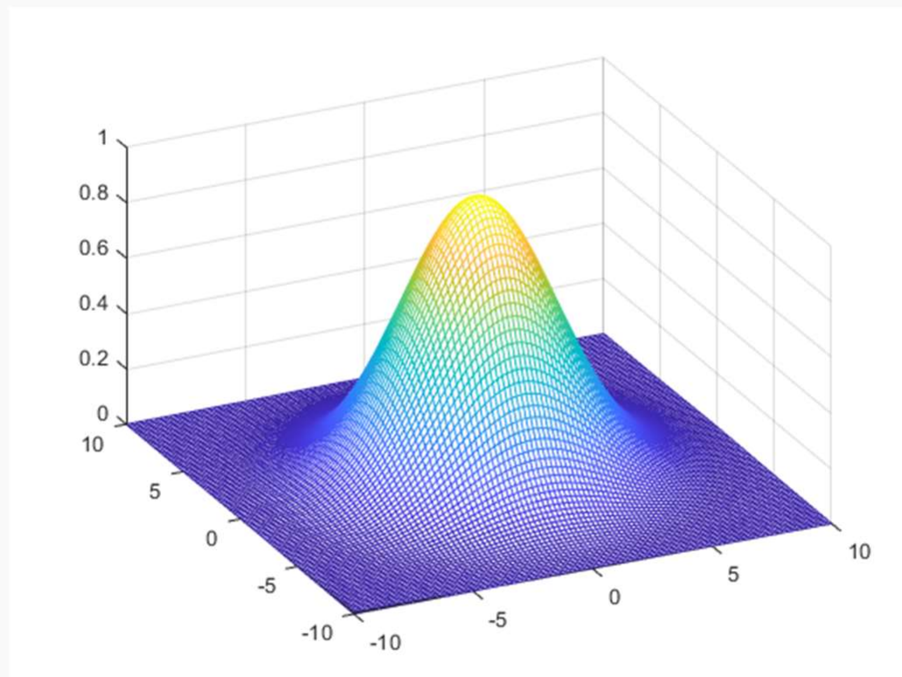
Increasing σ gives wider distributions

Gaussian equation

2-D Gaussian equation

$$f(x, y) = A \exp\left(-\frac{(x - x0)^2 + (y - y0)^2}{2\sigma^2}\right)$$





Gaussian filter

Original image

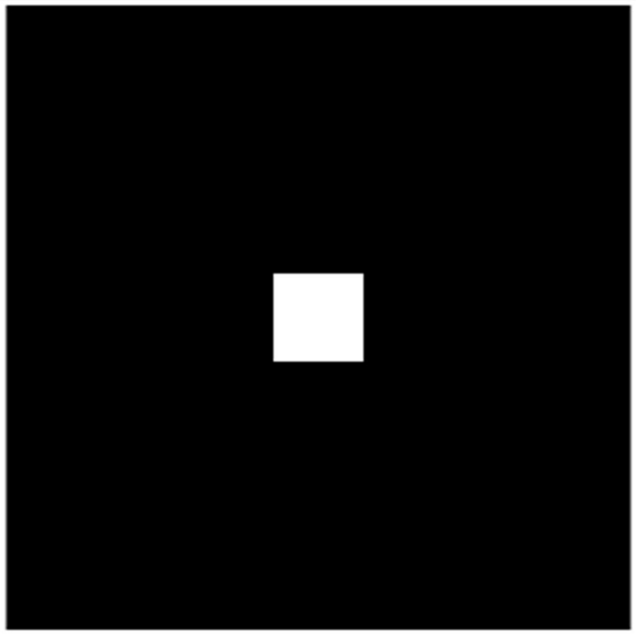
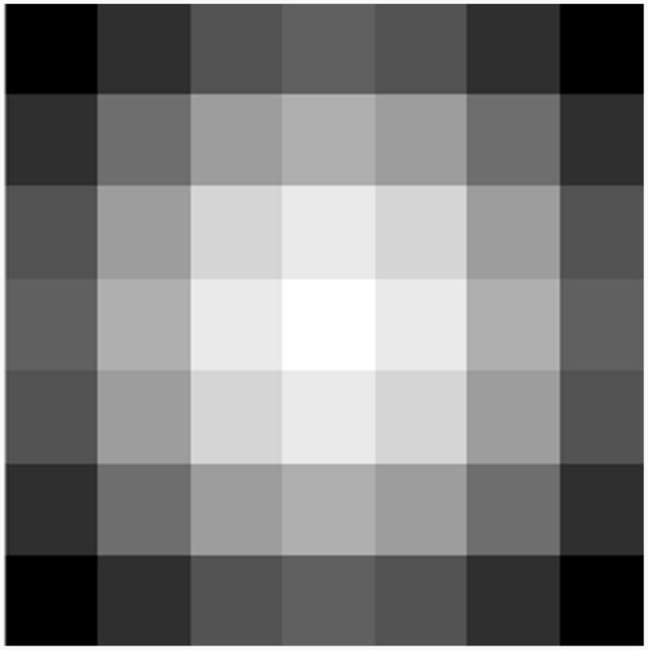


Image after Gaussian filter
σ = 3



The Gaussian filter spreads out pixel intensities in the image following a Gaussian distribution

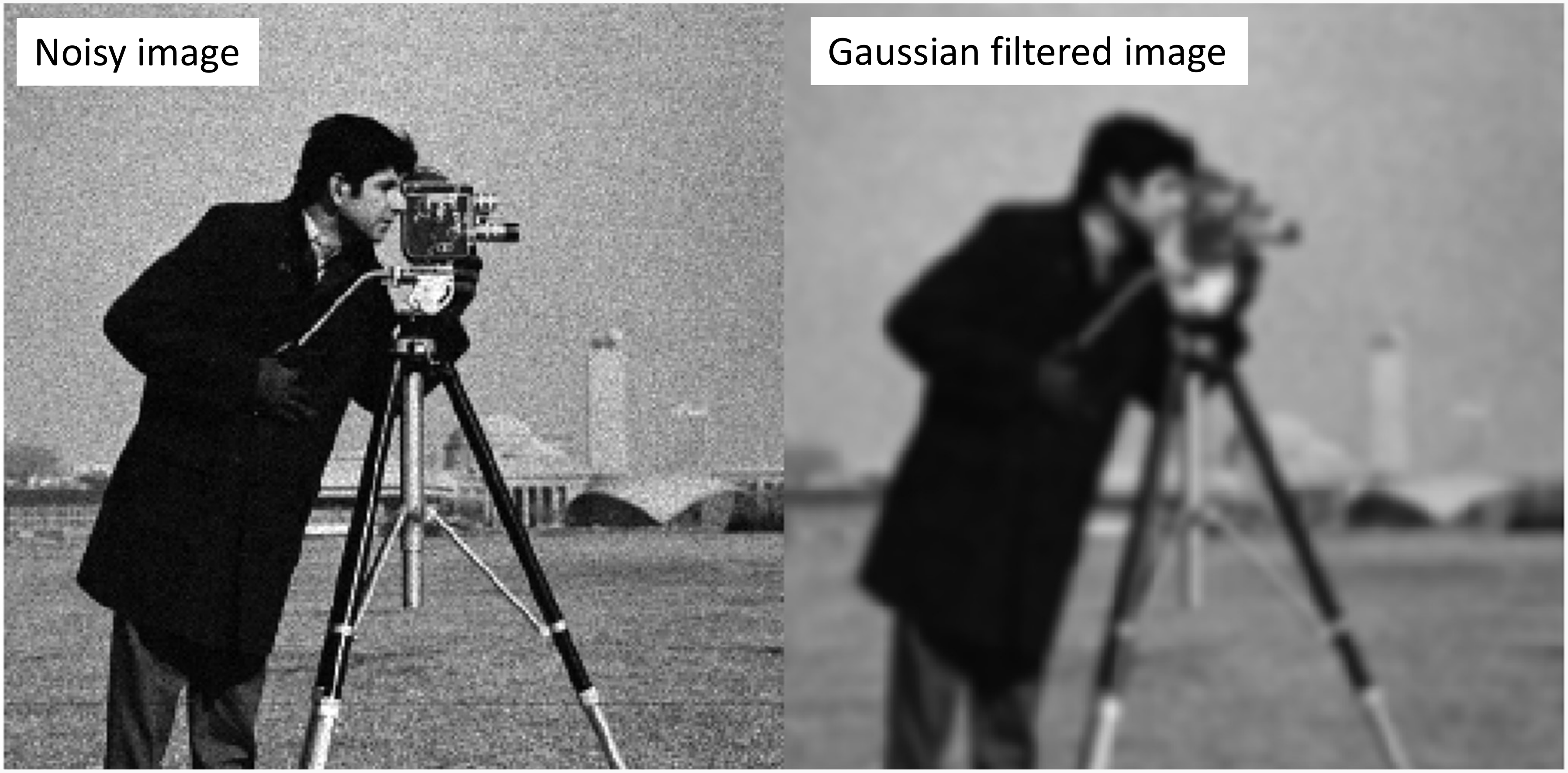
Gaussian filter

Syntax:

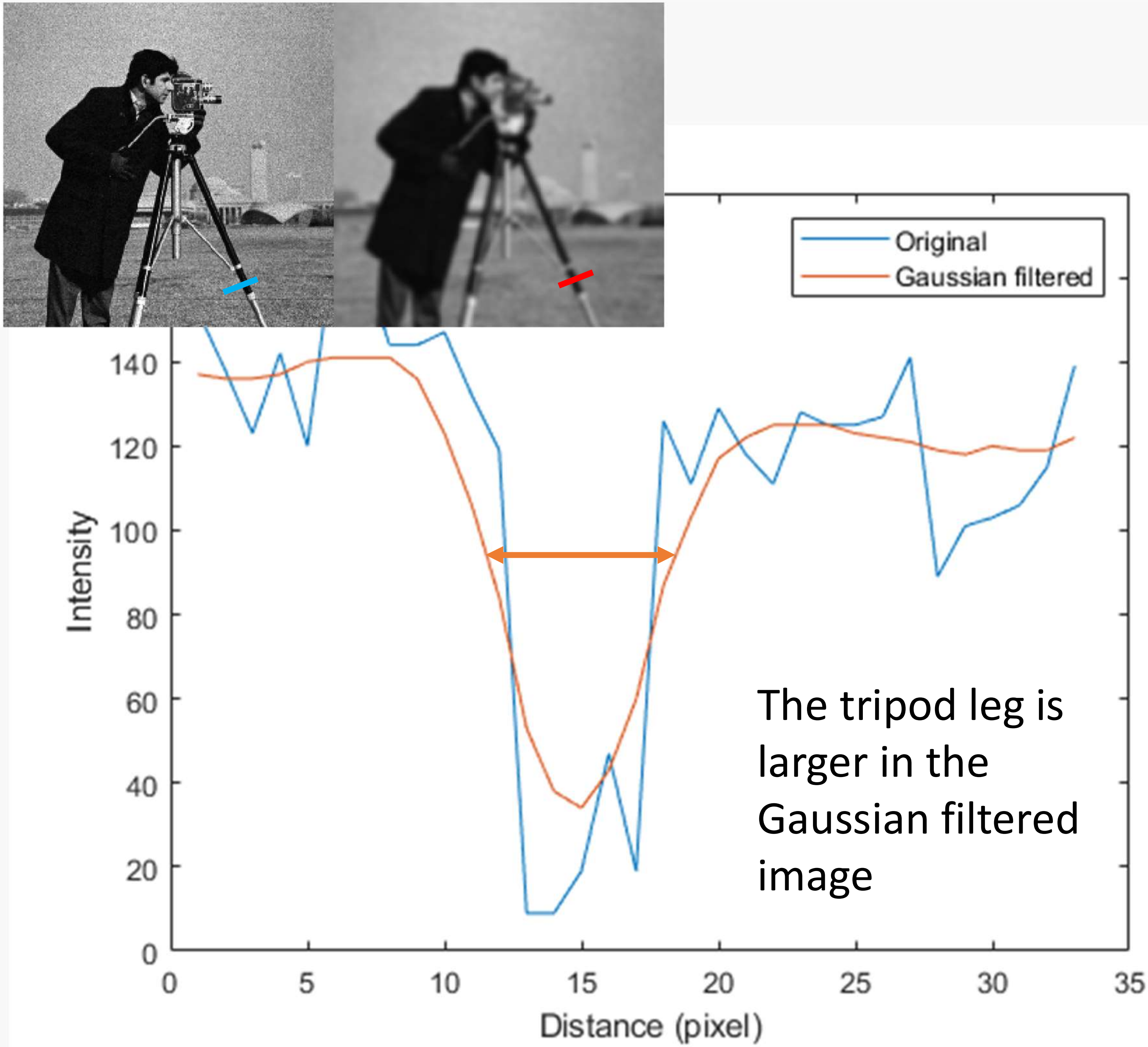
```
Ifilt = imgaussfilt(I, sigma)
```

sigma = width of filter

```
I = imread('cameraman.tif');
%Add noise to the image
I = imnoise(I, 'Poisson');


Ifilt = imgaussfilt(I, 2);
imshowpair(I, Ifilt, 'montage')
```



For obvious reasons, this function is also called a "Gaussian blur"



The tripod leg is larger in the Gaussian filtered image

Full example code

```
I = imread('cameraman.tif');
I = imnoise(I, 'Poisson');
Igauss = imgaussfilt(I, 2);

imshow([I, Igauss])
imshow(I)

%Measure line profile across a camera leg
[CX, CY, C1] = improfile; %Chosen by hand
C2 = improfile(Igauss, CX, CY); %Measure the same spot

figure;
plot(C1)
hold on
plot(C2)
```

Median filter

```
Iout = medfilt2(I, [M, N])
```

- For each input pixel, the output is computed as the median of the numbers in the specified neighborhood of `M x N` pixels
- M x N also called the "window"
- M x N should be odd numbers

Example of a 3x3 median filter

Input image

10	5	10	2
5	2	3	1
8	1	5	1

10	5	10	2
5	2	3	1
8	1	5	1

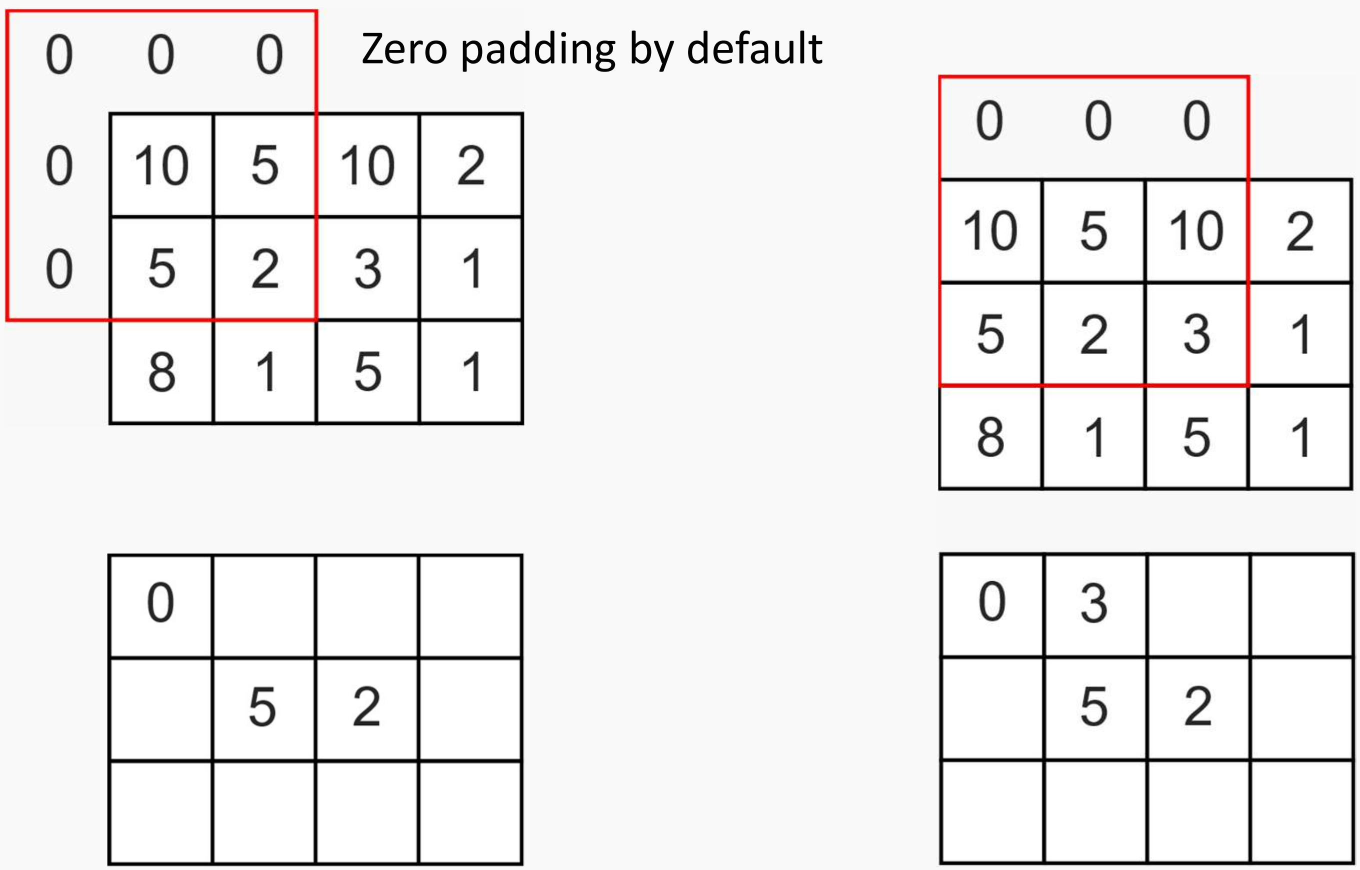
10	5	10	2
5	2	3	1
8	1	5	1

Output image

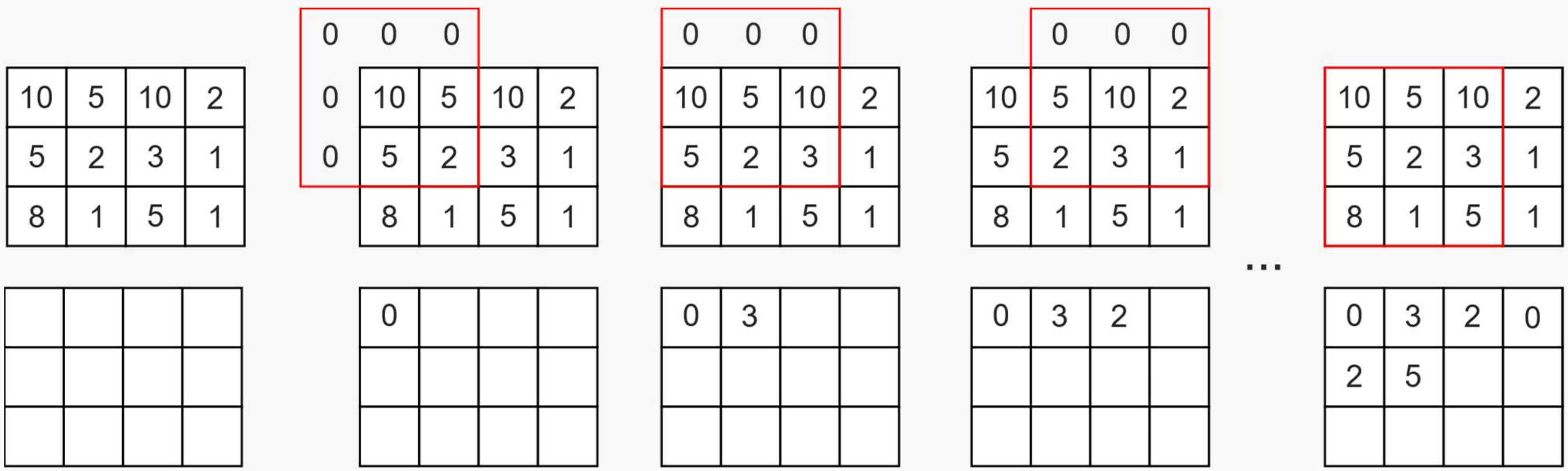
	5		

	5	2	

Example of a 3x3 median filter



The median filter is applied pixel-by-pixel



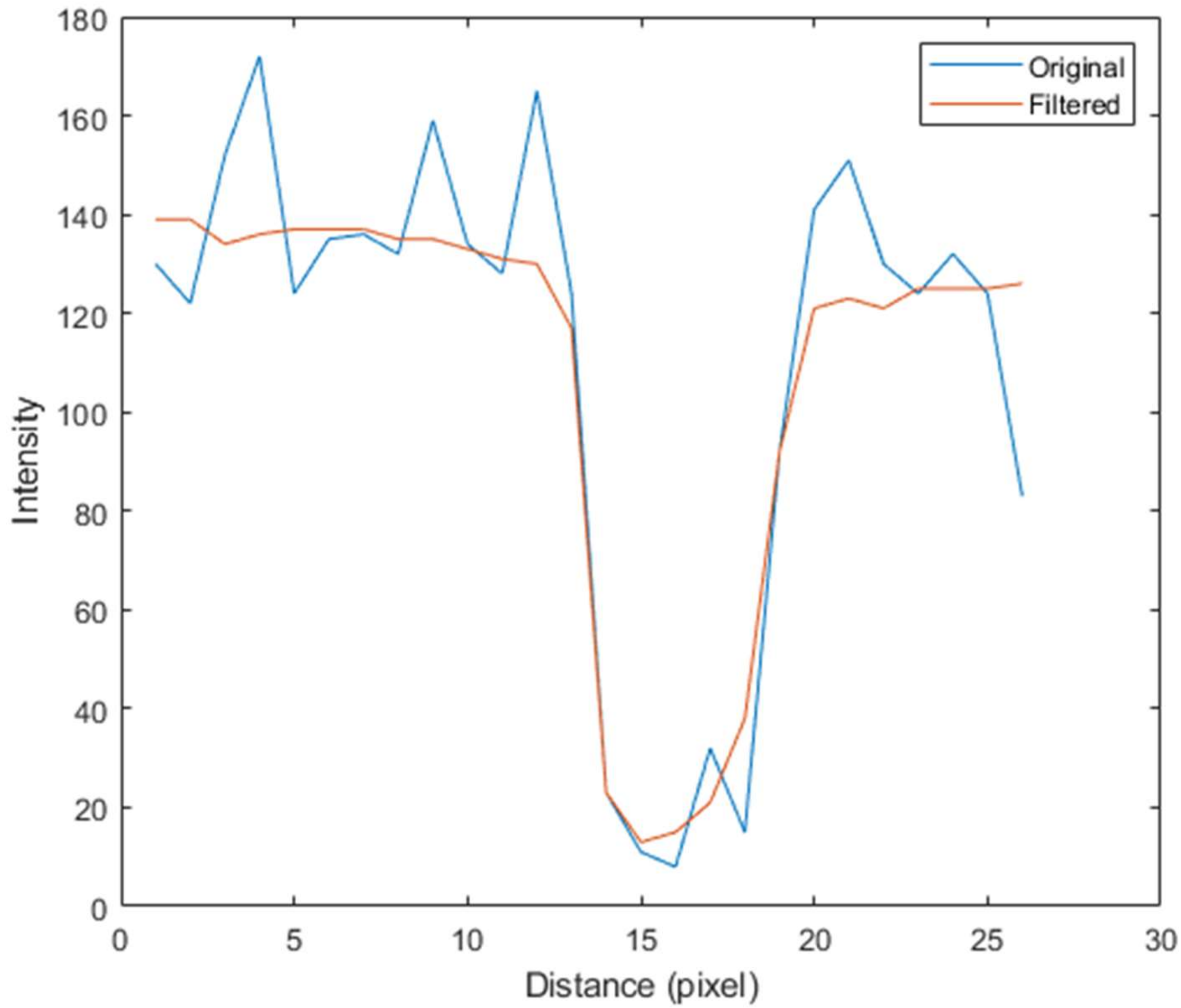
Output =

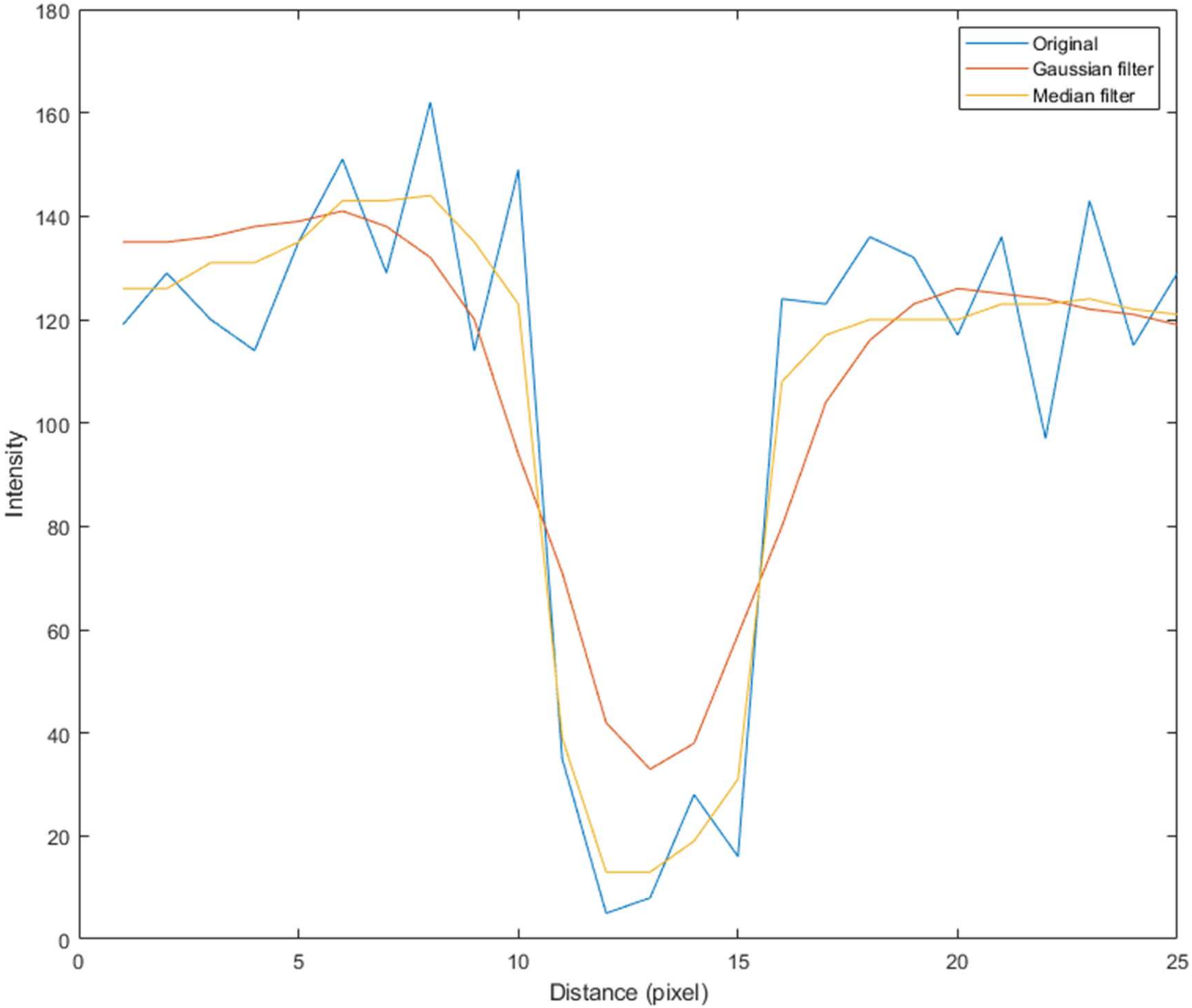
0	3	2	0
2	5	2	1
0	2	1	0

Example

Change your script to use the median filter with a window of 5x5 instead of the Gaussian

```
Ifilt = medfilt2(I, [M, N])
```

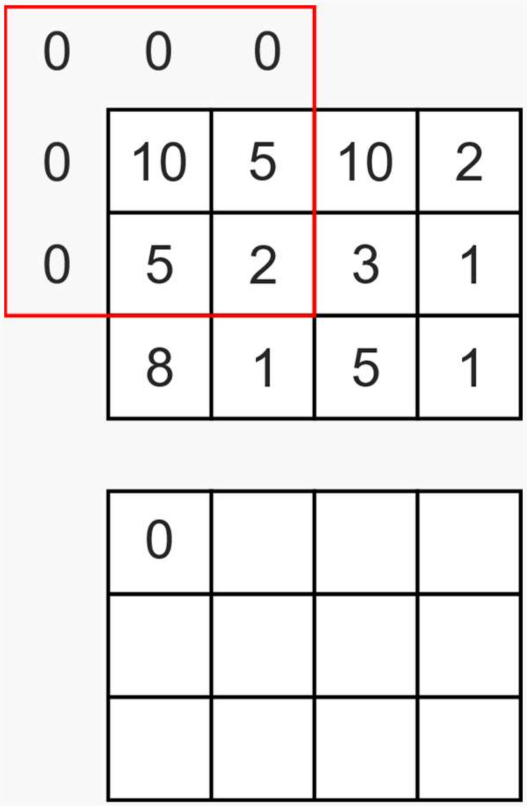

Original
Filtered
Intensity
Distance (pixel)
180
160
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30


Original
Gaussian filter
Median filter
Intensity
Distance (pixel)
0
20
40
60
80
100
120
140
160
180
0
5
10
15
20
25

Example

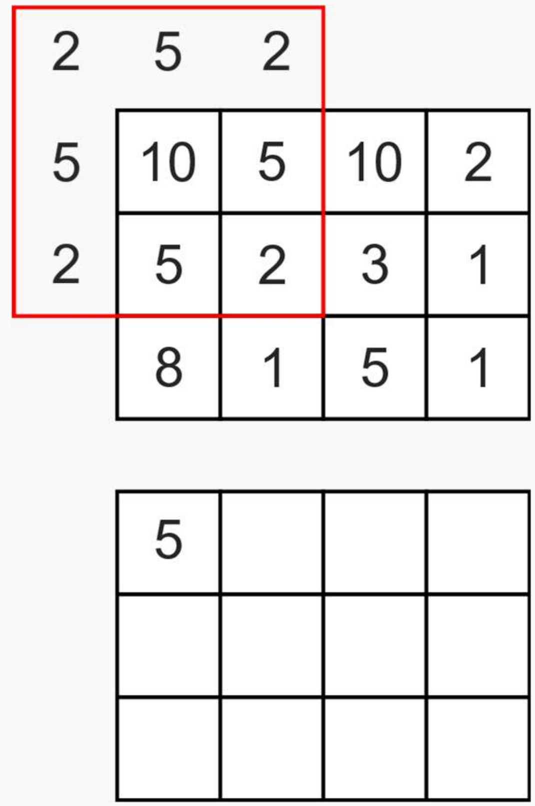
```
I = imread('cameraman.tif');
I = imnoise(I, 'Poisson');
Igauss = imgaussfilt(I, 2);

%Measure line profile across a camera leg
imshow(I)
[CX, CY, C1] = improfile; %Chosen by hand
C2 = improfile(Igauss, CX, CY); %Measure the same spot

Imedfilt = medfilt2(I, [5, 5]);

%Measure line profile across a camera leg
C3 = improfile(Imedfilt, CX, CY); %Measure the same spot

figure;
plot(C1)
hold on
plot(C2)
plot(C3)
hold off
legend('Original', 'Gaussian filter', 'Median filter')
xlabel('Distance (pixel)')
ylabel('Intensity')
```

Edge padding options



Default zero padding

`medfilt2(I, [3 3])`



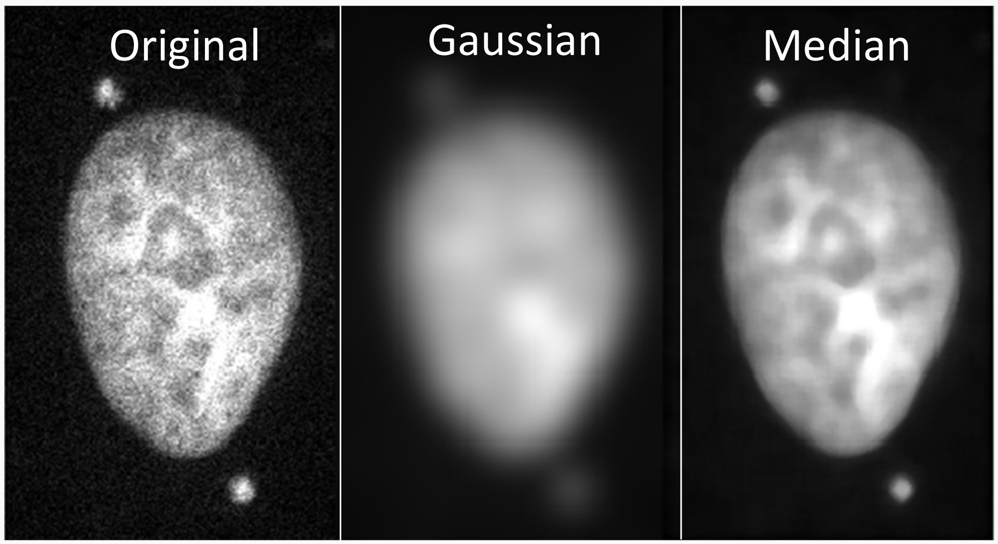
'Symmetric'

Reflects matrix

`medfilt2(I, [3 3], 'symmetric')`

When would you pick Gaussian over median filters?

- Gaussian filters can be useful when you are trying to segment a large object, but its intensity is uneven



- Gaussian filters are also used when detecting particles – more on this in a later lecture!

Choosing size of the filters

- The size of the filters chosen depends on the size of the feature you are trying to detect

- Size = sigma for Gaussian filter, the window [M, N] for the median filter

- In general, the filters should be larger than the noise you are trying to remove, but usually smaller than the object or feature you are trying to detect