

# Image filtering

Thierry Pécot  
Research engineer  
FAIIA

# Outline

Point operations

Image filtering

Frequential decomposition

# Mathematical operations applied to images

**Mathematical operations** (addition, subtraction, multiplication, division, log, ...) applied to **each pixel** of the image:

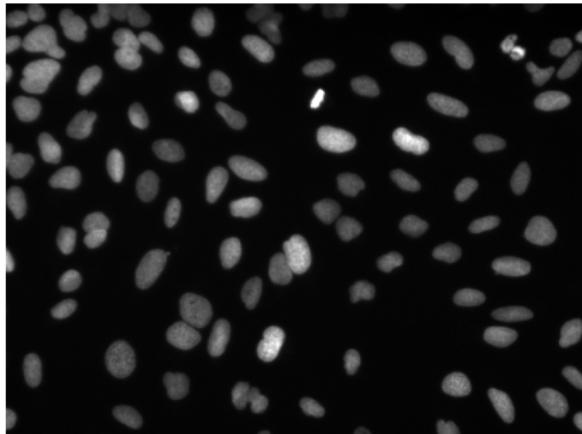
$$\mathbf{I}_1 = f(\mathbf{I}_0), \text{ where } f(I_0(x, y)) = I_0(x, y) + a \text{ for example} \quad (1)$$

# Mathematical operations applied to images

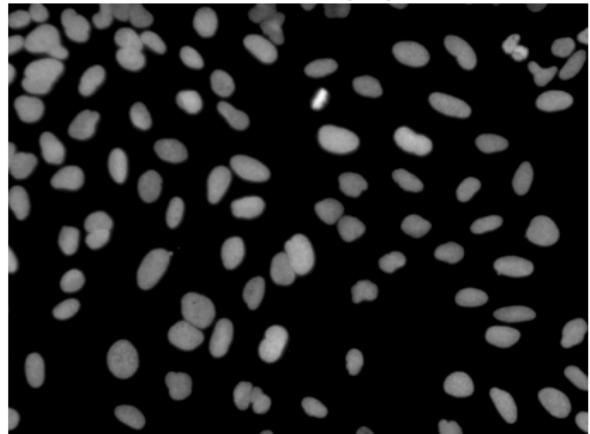
**Mathematical operations** (addition, subtraction, multiplication, division, log, ...) applied to **each pixel** of the image:

$$\mathbf{I}_1 = f(\mathbf{I}_0), \text{ where } f(I_0(x, y)) = I_0(x, y) + a \text{ for example} \quad (1)$$

$\mathbf{I}_0$



$\mathbf{I}_1 = \log(\mathbf{I}_0)$



# Practice

<https://youtu.be/0hV0FHz70HQ>

## Mathematical operations applied between images

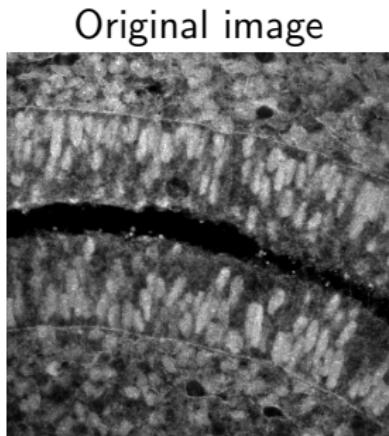
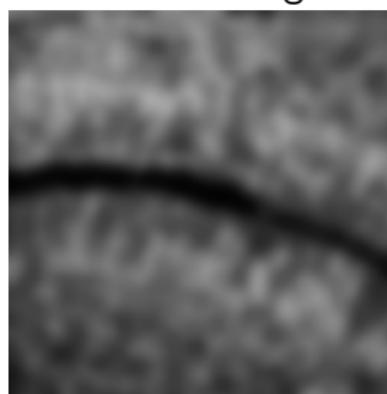
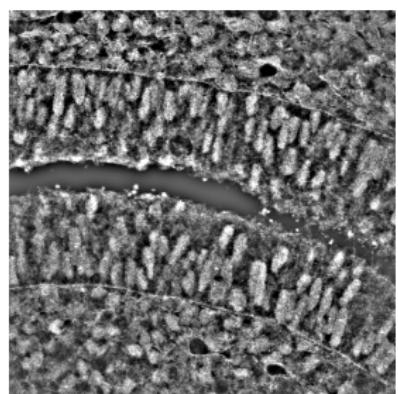
**Mathematical operations** (addition, subtraction, multiplication, division, ...) applied between the **pixels of two images** with the same size

$$\mathbf{I}_3 = g(\mathbf{I}_1, \mathbf{I}_2), \text{ where } g(\mathbf{I}_1, \mathbf{I}_2) = I_1(x, y) + I_2(x, y) \text{ for example} \quad (2)$$

## Mathematical operations applied between images

**Mathematical operations** (addition, subtraction, multiplication, division, ...) applied between the **pixels of two images** with the same size

$$\mathbf{I}_3 = g(\mathbf{I}_1, \mathbf{I}_2), \text{ where } g(\mathbf{I}_1, \mathbf{I}_2) = I_1(x, y) + I_2(x, y) \text{ for example} \quad (2)$$

 $\mathbf{I}_1$  $f(\mathbf{I}_1) \text{ where } f \text{ is a Gaussian blur}$  $\mathbf{I}_2 = g(\mathbf{I}_1, f(\mathbf{I}_1)) \text{ where } g \text{ is a subtraction}$

# Practice

<https://youtu.be/EFZQTzM2c6s>

# Outline

Point operations

Image filtering

Frequential decomposition

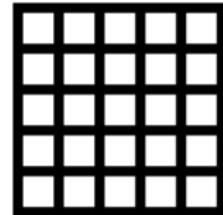
## Linear filtering

**Image**

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

**Filter**

1	1	1
1	1	1
1	1	1

**Filtered image**

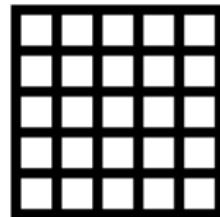
## Linear filtering

**Image**

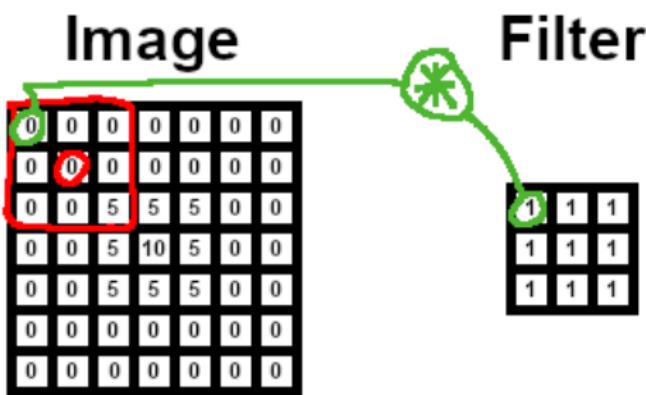
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	5	5	5	0	0	0
0	0	5	10	5	0	0	0
0	0	5	5	5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

**Filter**

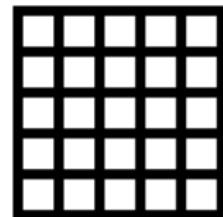
1	1	1
1	1	1
1	1	1

**Filtered image**

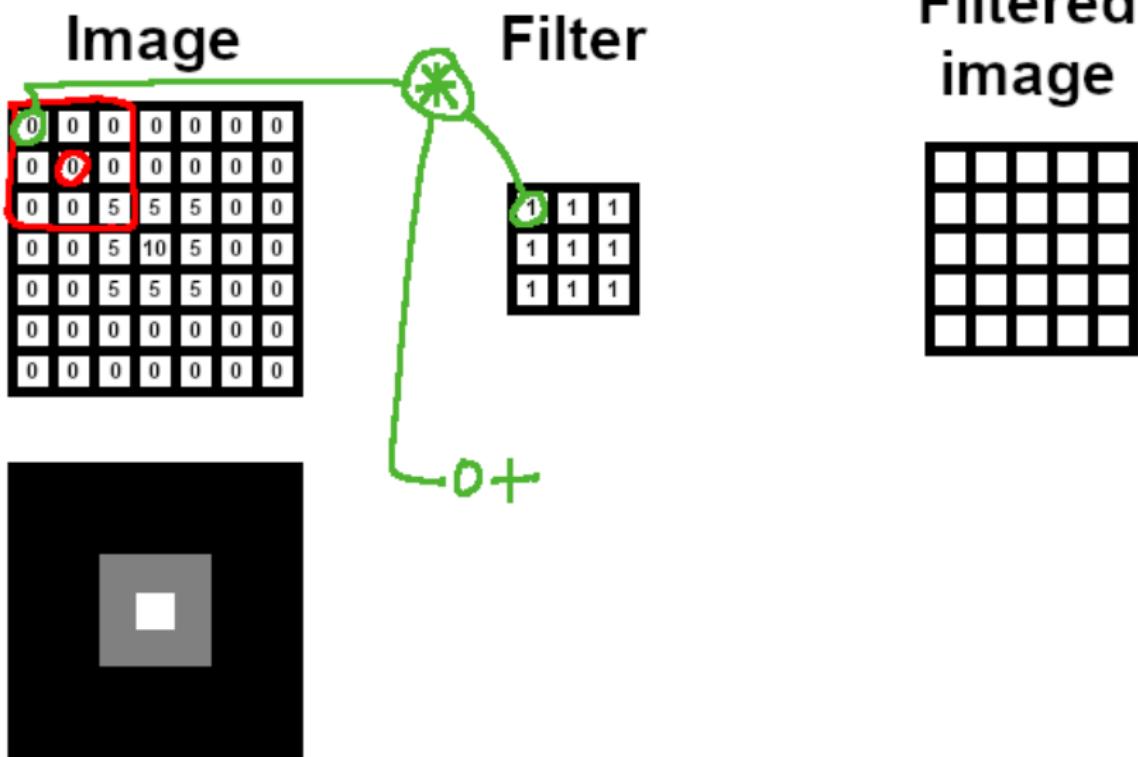
## Linear filtering



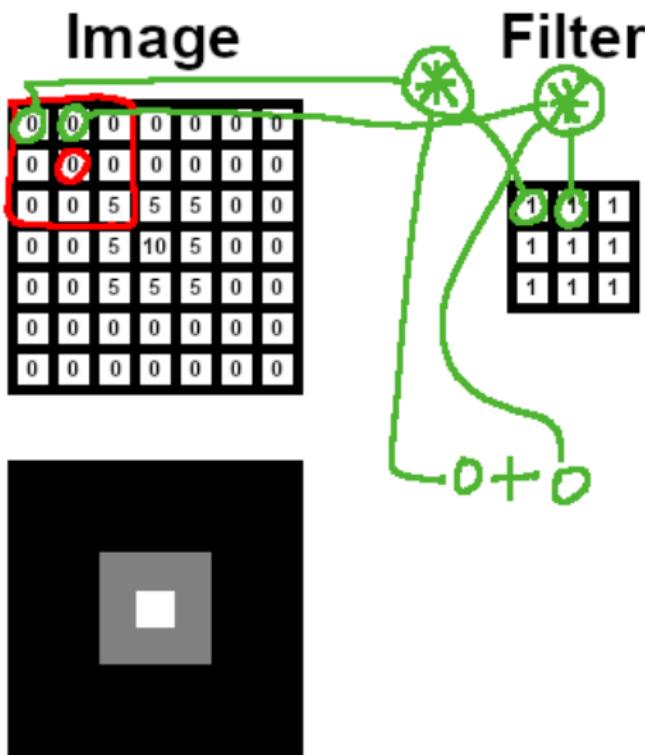
**Filtered image**



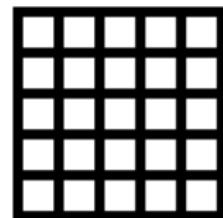
## Linear filtering



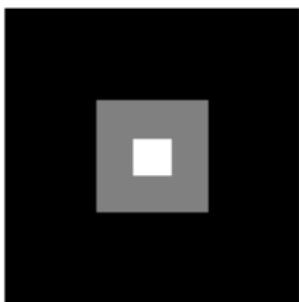
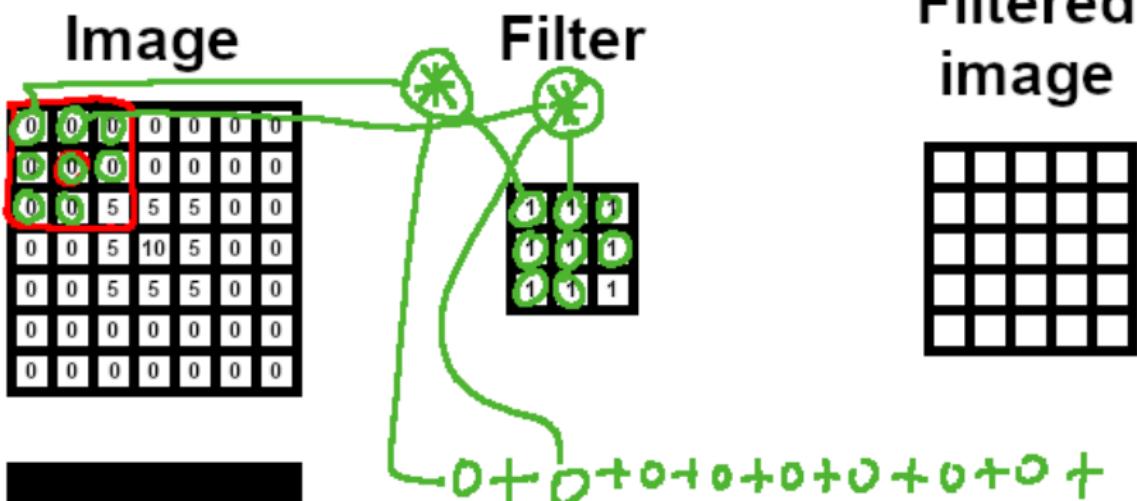
## Linear filtering



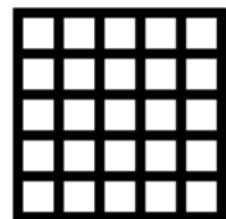
Filtered  
image



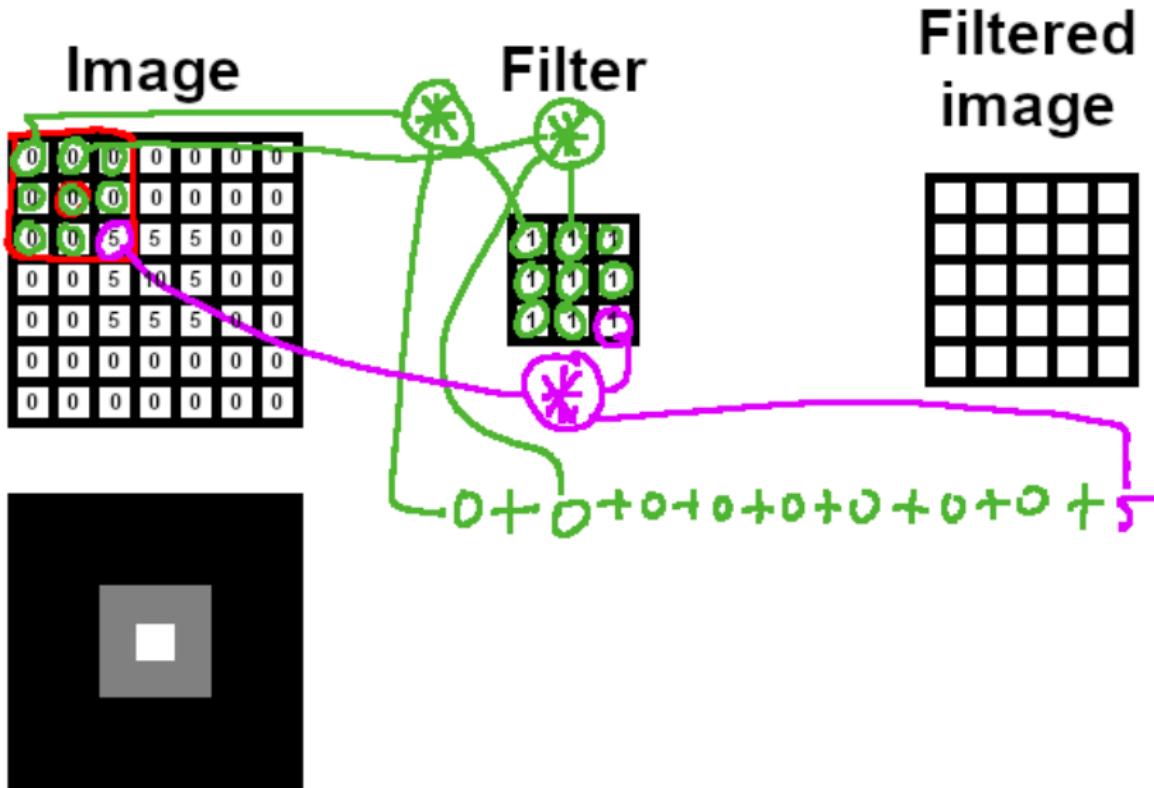
## Linear filtering



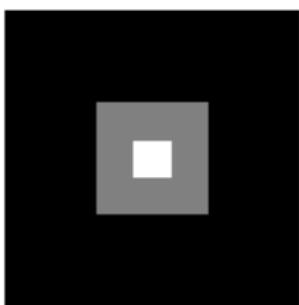
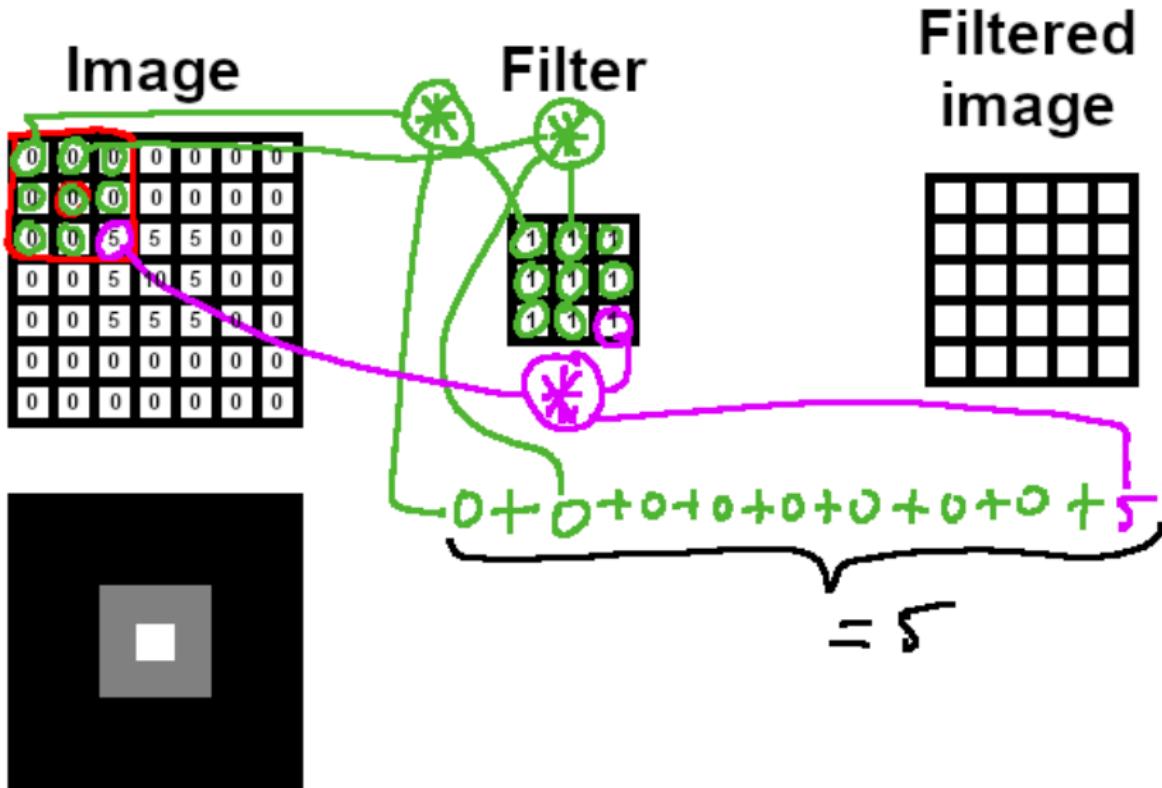
Filtered  
image



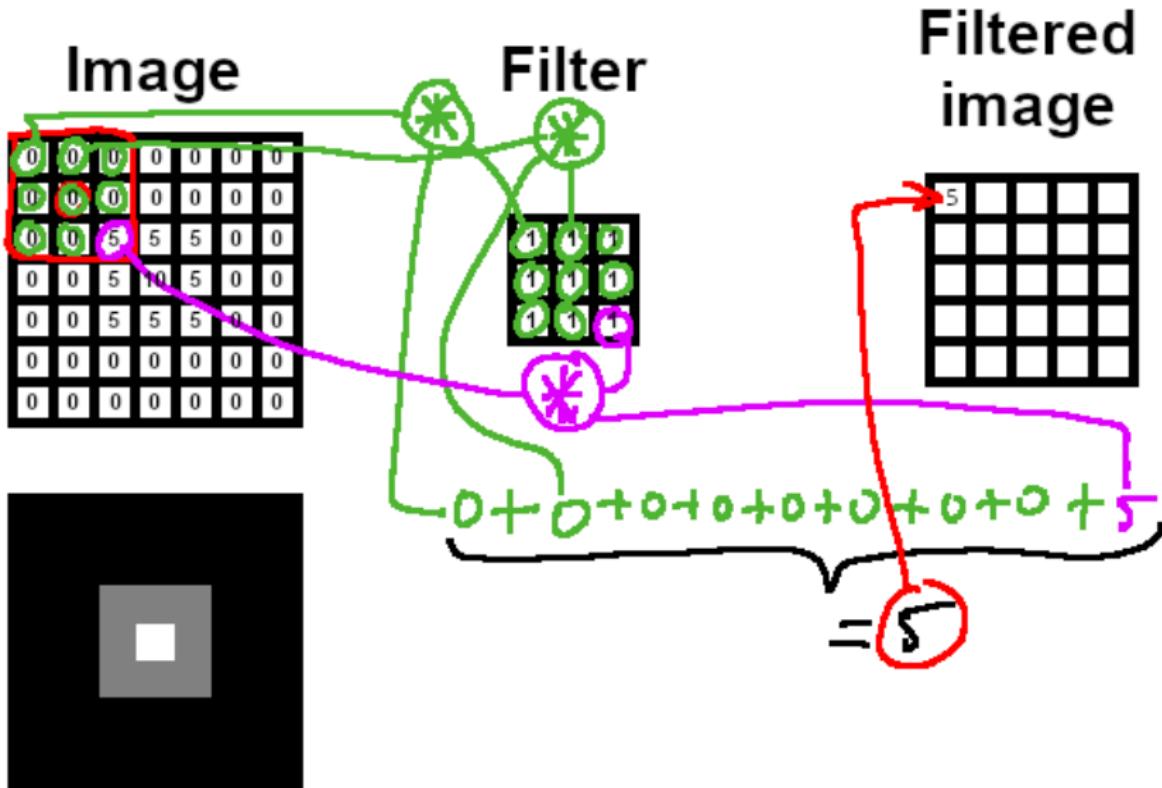
## Linear filtering



## Linear filtering



## Linear filtering



## Linear filtering

**Image**

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

**Filter**

1	1	1
1	1	1
1	1	1

**Filtered image**

5		



## Linear filtering

**Image**

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

**Filter**

1	1	1
1	1	1
1	1	1

**Filtered image**

5		



## Linear filtering

Image

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Filter

1	1	1
1	1	1
1	1	1

Filtered image

5		

$$0+0+0+0+0+0+\frac{5}{15}$$



## Linear filtering

Image

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0



Filter

1	1	1
1	1	1
1	1	1

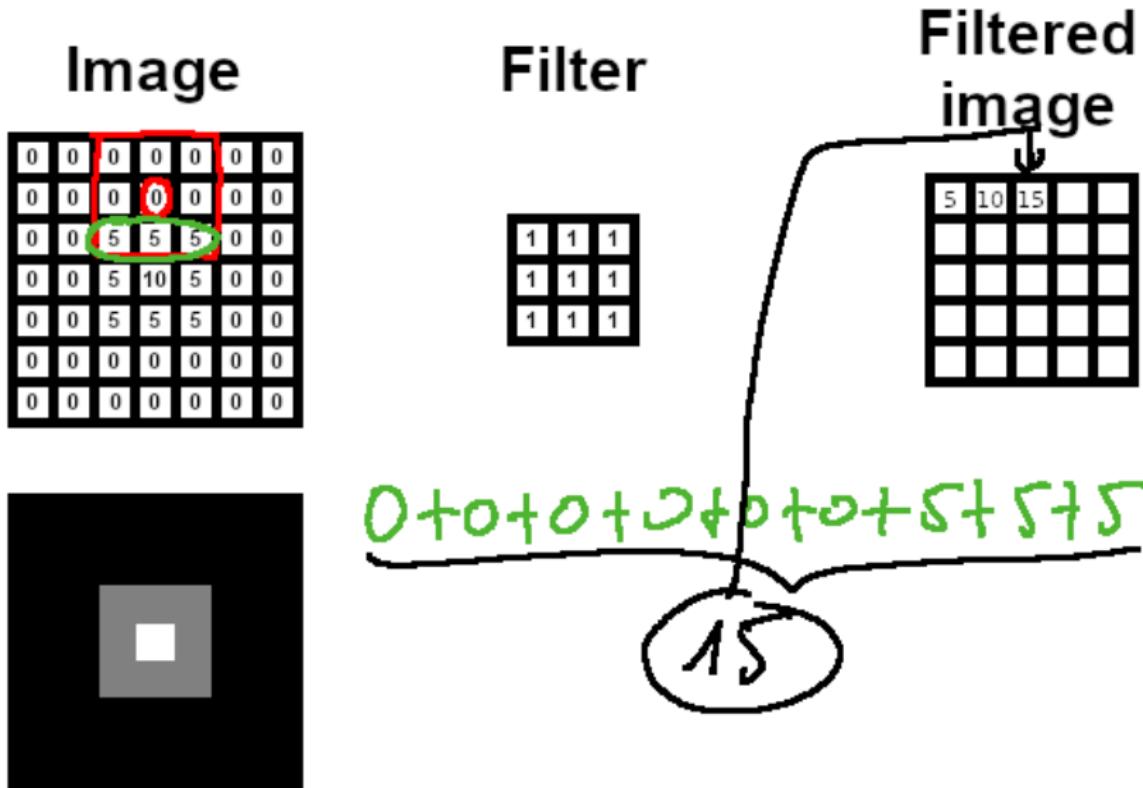
Filtered image

5	10		

$$0 + 0 + 0 + 0 + 0 + 0 + 0 + 10$$

10

## Linear filtering



## Linear filtering

**Image**

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

**Filter**

1	1	1
1	1	1
1	1	1

**Filtered image**

5	10	15	10	5
10	25	35	25	10
15	35			



## Linear filtering

Image

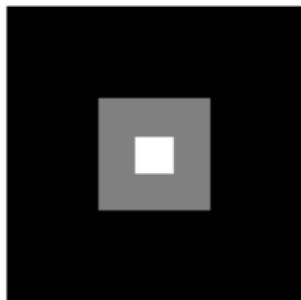
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Filter

1	1	1
1	1	1
1	1	1

Filtered image

5	10	15	10	5
10	25	35	25	10
15	35	50		



$$S + S + S + F + 10 + S + S + S + F$$

50

## Linear filtering

Image

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Filter

1	1	1
1	1	1
1	1	1

Filtered image

5	10	15	10	5
10	25	35	25	10
15	35	50	35	15
10	25	35	25	10
5	10	15	10	5



# Linear filtering

Image

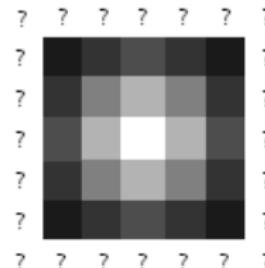
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Filter

1	1	1
1	1	1
1	1	1

Filtered image

5	10	15	10	5
10	25	35	25	10
15	35	50	35	15
10	25	35	25	10
5	10	15	10	5



## Linear filtering

**Image**

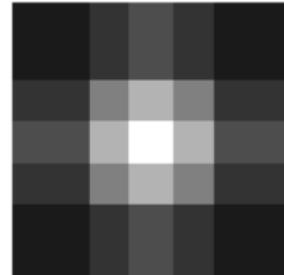
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	5	5	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

**Filter**

1	1	1
1	1	1
1	1	1

**Filtered image**

5	10	15	10	5
10	25	35	25	10
15	35	50	35	15
10	25	35	25	10
5	10	15	10	5



# Linear filtering

Original image



1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

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

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

# Linear filtering

Original image



Mean filter



1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

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

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

# Linear filtering

Original image



Mean filter



1	1	1
1	1	1
1	1	1

Gaussian filter



1	2	1
2	4	2
1	2	1

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

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

# Linear filtering

Original image



Mean filter



1	1	1
1	1	1
1	1	1

Gaussian filter



1	2	1
2	4	2
1	2	1

x contour



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

y contour



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

# Practice

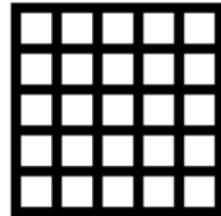
<https://youtu.be/F4xRAaTk1j0>

## Non linear filtering

**Image**

0	0	0	0	0	0	0
0	10	0	0	0	0	0
0	0	5	5	0	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Median

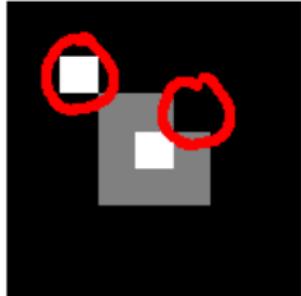
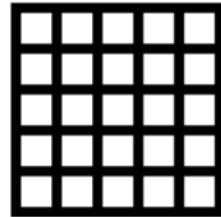
**Filtered image**

## Non linear filtering

**Image**

0	0	0	0	0	0	0
0	10	0	0	0	0	0
0	0	5	5	0	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Median

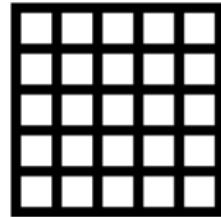
**Filtered image**

## Non linear filtering

**Image**

0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
0	0	5	5	0	0	0	0
0	0	5	10	5	0	0	0
0	0	5	5	5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Median

**Filtered image**

## Non linear filtering

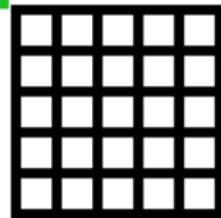
Image

0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
0	0	5	5	5	0	0	0
0	0	5	10	5	0	0	0
0	0	5	5	5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Median

0,0,0,0,10,0,0,0,5

Filtered image



## Non linear filtering

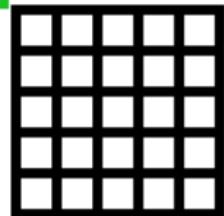
Image

0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
0	0	5	5	5	0	0	0
0	0	5	10	5	0	0	0
0	0	5	5	5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

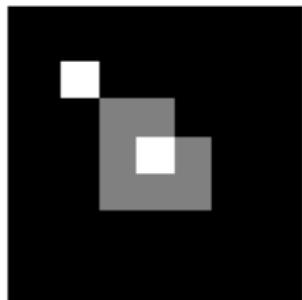
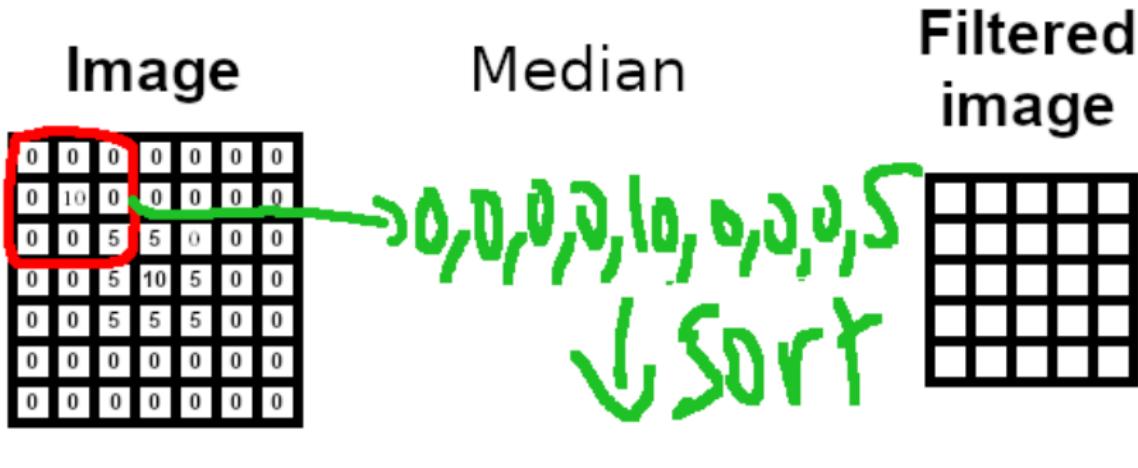
Median

0,0,0,0,10,5,0,0,5  
↓ Sort

Filtered image

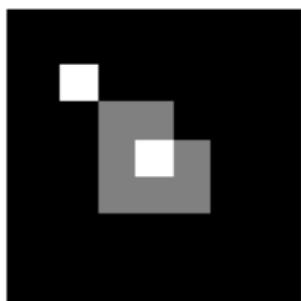
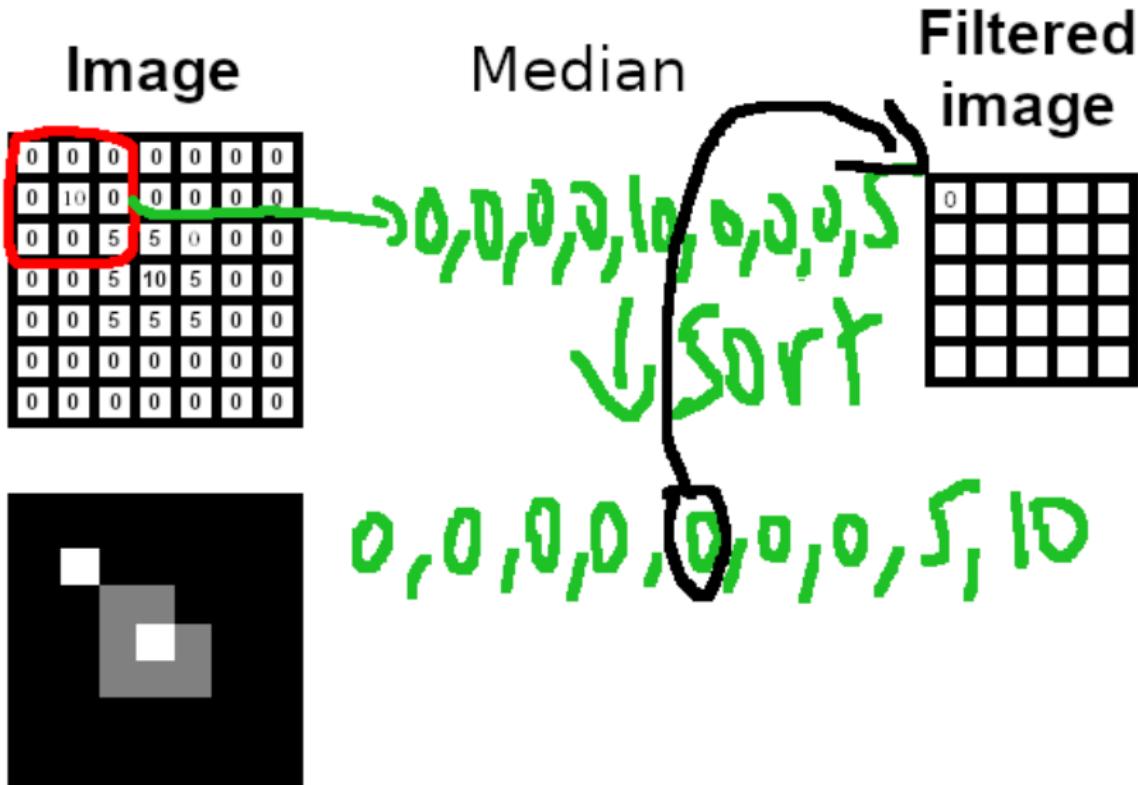


## Non linear filtering



**0, 0, 0, 0, 0, 0, 5, 10**

## Non linear filtering



## Non linear filtering

**Image**

0	0	0	0	0	0	0
0	10	0	0	0	0	0
0	0	5	5	0	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Median

**Filtered image**

0			



## Non linear filtering

**Image**

0	0	0	0	0	0	0
0	10	0	0	0	0	0
0	0	5	5	0	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Median

**Filtered image**

0	0	0	0	0
0				



## Non linear filtering

Image

0	0	0	0	0	0	0
0	10	0	0	0	0	0
0	0	5	5	0	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Median

0, 0, 0, 0, 5, 5, 0, 5, 10

↓ sort

Filtered image

0	0	0	0	0
0				



0, 0, 0, 0, 5, 5, 5, 10, 10

## Non linear filtering

Image

0	0	0	0	0	0	0
0	10	0	0	0	0	0
0	0	5	5	0	0	0
0	0	5	10	5	0	0
0	0	5	5	5	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Median

0, 0, 0, 0, 5, 5, 0, 5, 10

↓ sort

Filtered image

0	0	0	0	0
0	5			

0, 0, 0, 0, 5, 5, 5, 10, 10



## Non linear filtering

**Image**

0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
0	0	5	5	0	0	0	0
0	0	5	10	5	0	0	0
0	0	5	5	5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Median

**Filtered image**

0	0	0	0	0	0
0	5	5	0	0	0
0	5	5	5	0	0
0	0	5	0	0	0
0	0	0	0	0	0



## Non linear filtering

**Image**

0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
0	0	5	5	0	0	0	0
0	0	5	10	5	0	0	0
0	0	5	5	5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Median

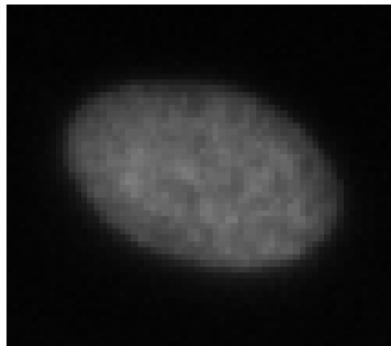
**Filtered image**

0	0	0	0	0	0
0	5	5	0	0	0
0	5	5	5	0	0
0	0	5	0	0	0
0	0	0	0	0	0

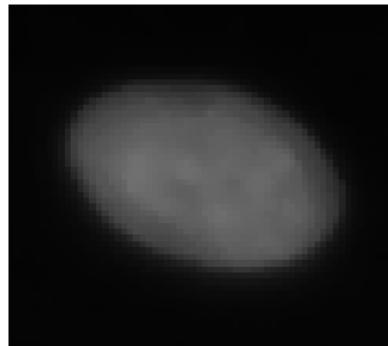


## Non linear filtering

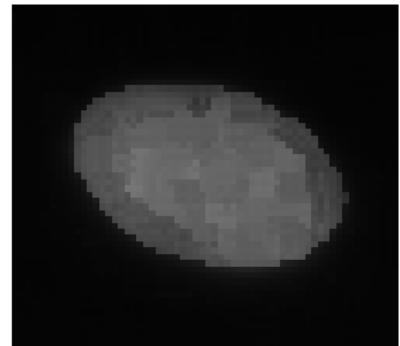
Original image



Median filter



Kuwahara filter



# Practice

<https://youtu.be/NOiNLsJ93Jk>

# Outline

Point operations

Image filtering

Frequential decomposition

# Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions

# Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions

Fourier transform of  $f(t)$ :

$$\hat{f}(\nu) = \sum_{-\infty}^{+\infty} f(t) e^{-i2\pi\nu t} dt \quad (3)$$

Inverse Fourier transform of  $\hat{f}(\nu)$ :

$$f(t) = \sum_{-\infty}^{+\infty} \hat{f}(\nu) e^{i2\pi\nu t} dt \quad (4)$$

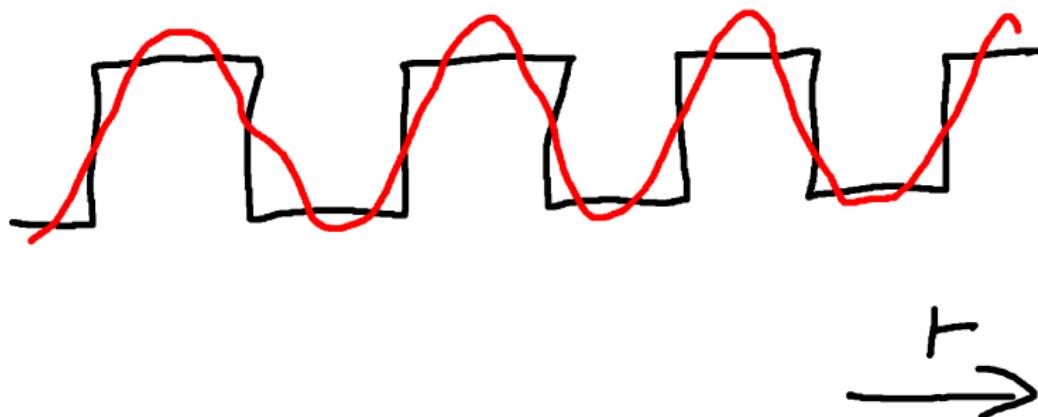
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions



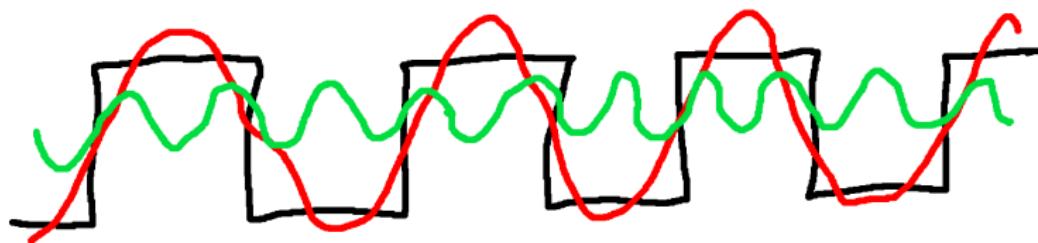
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions



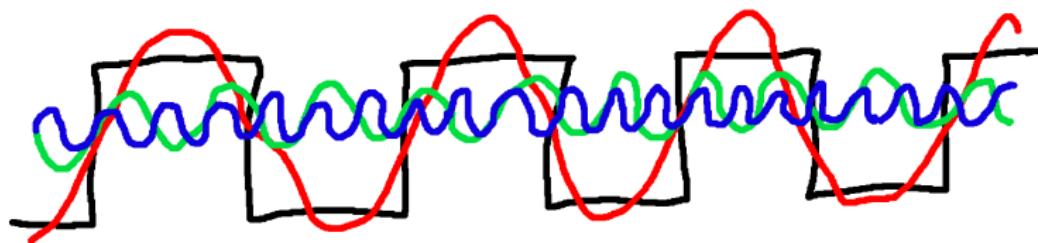
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions



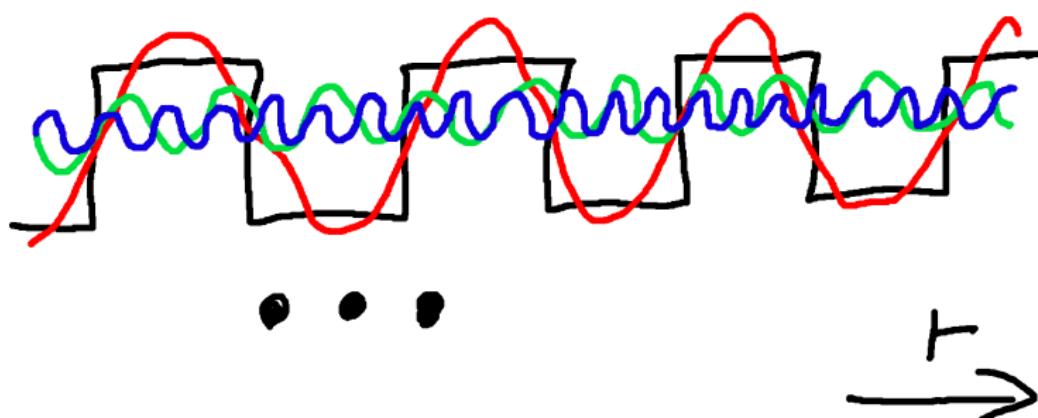
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions



## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions



## Fourier transform

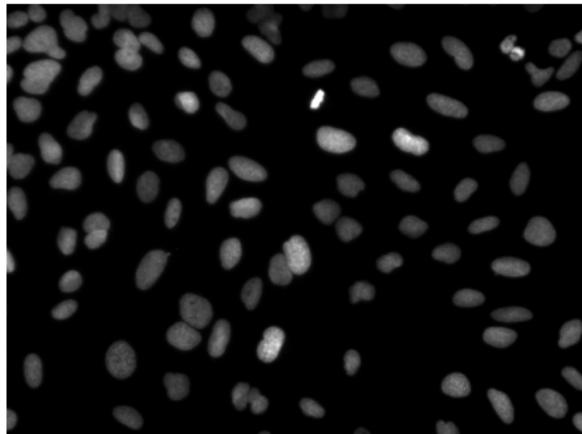
In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions



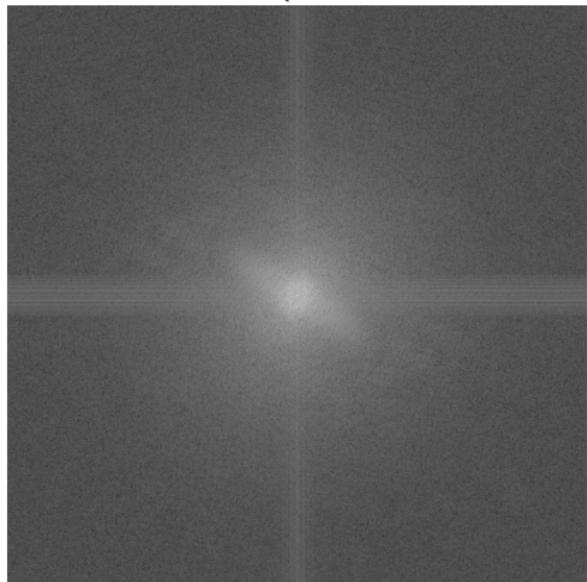
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions

Original image (spatial domain)



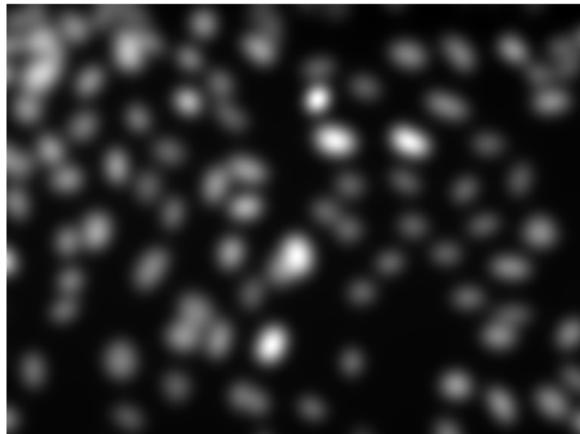
Fourier transform (frequential domain)



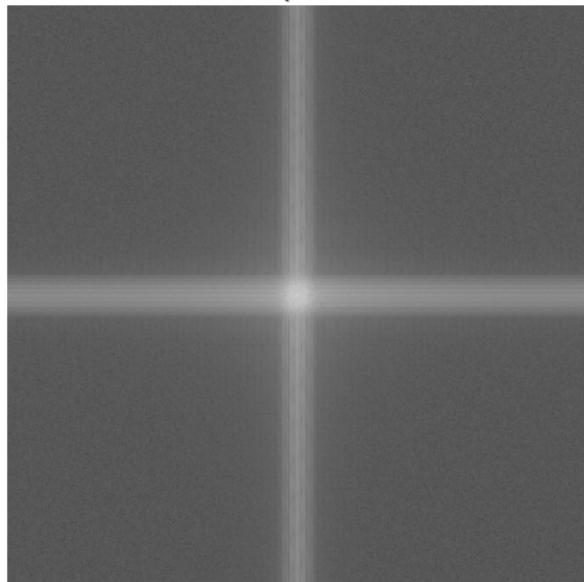
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions

Blurred image (spatial domain)



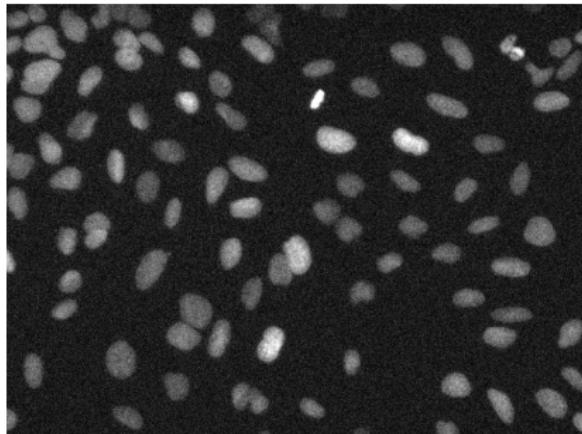
Fourier transform (frequential domain)



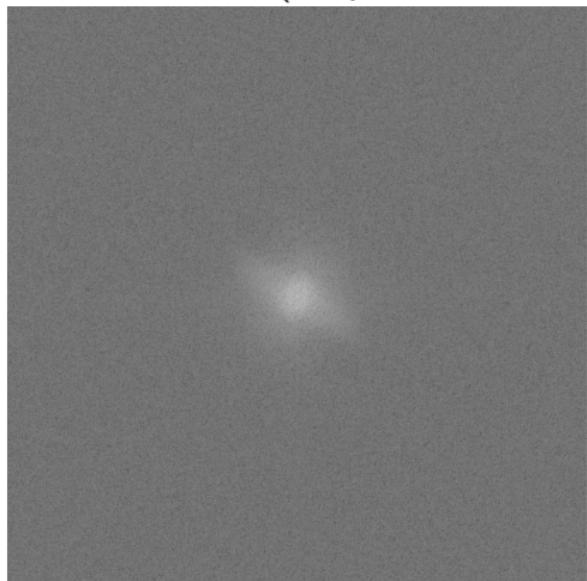
## Fourier transform

In signal processing, **Fourier analysis** consists in approximating a function by **sums** of simple **trigonometric** functions

Noisy image (spatial domain)



Fourier transform (frequential domain)



# Practice

[https://youtu.be/E\\_Vt5djPCfE](https://youtu.be/E_Vt5djPCfE)