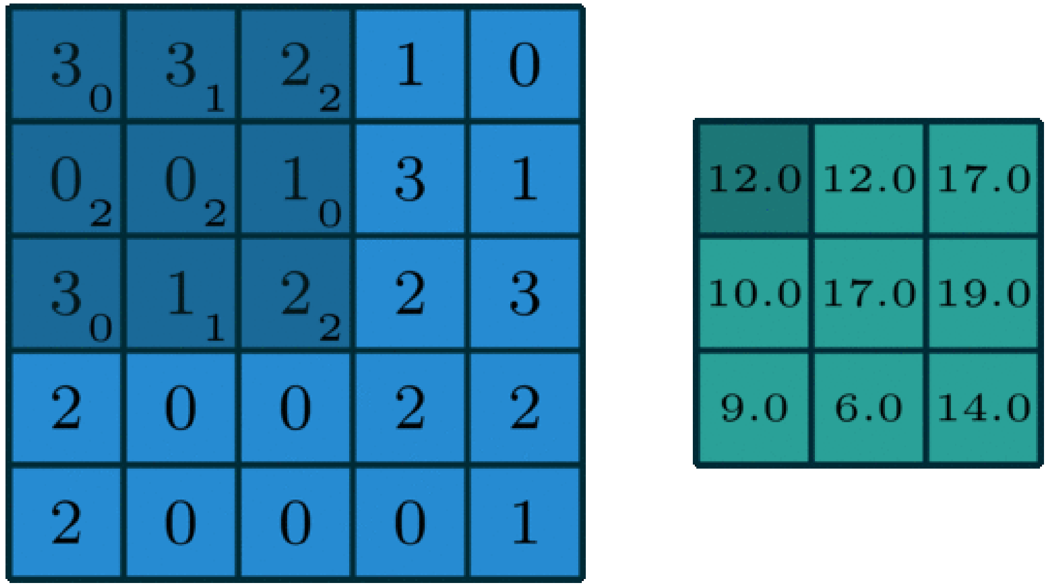
COMS20017 – Algorithms & Data



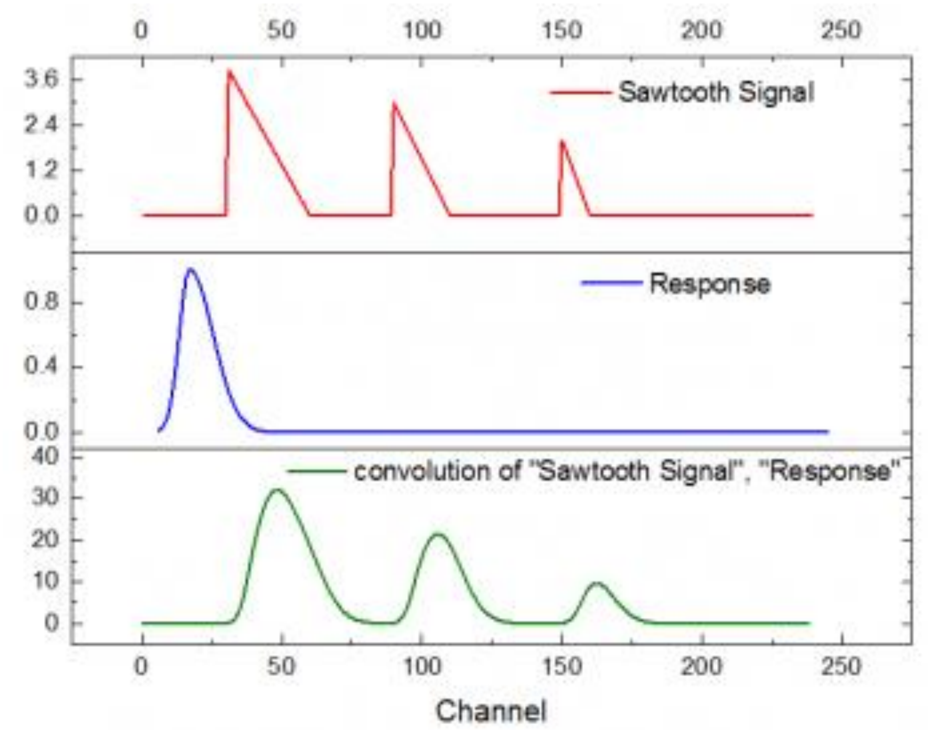
Convolutions

March 2025

Majid Mirmehdi

Lecture MM-08

Next in DDCS



Feature Selection and Extraction

- Signal basics and Fourier Series
- 1D and 2D Fourier Transform
- Another look at features
- PCA for dimensionality reduction
- **Convolutions**

https://www.originlab.com/doc/Origin-Help/Convolution

Correlation

- We are often interested in the degree of similarity between two sequences in terms of their variation (independent of their absolute values)

- Correlation of two sequences u and v of length N

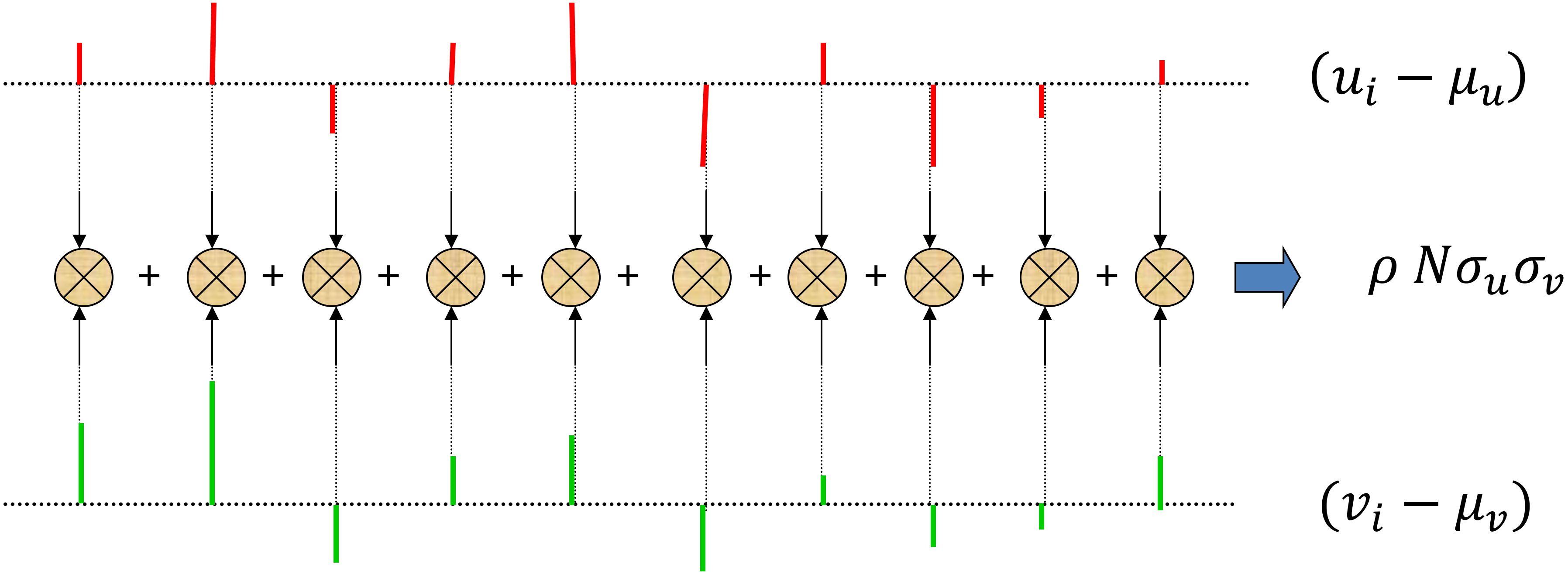
$$\rho = \frac{1}{N\sigma_u\sigma_v}\sum_{i=0}^{N-1}(u_i - \mu_u)\,(v_i - \mu_v)$$

Division by variance product $\sigma_u\sigma_v$ normalises measure to be independent of absolute value → captures similarity in variation (or *structure*).

Note similarity with covariance!

Example: Correlating Coefficient for Two Sequences

$$\rho = \frac{1}{N\sigma_u\sigma_v}\sum_{i=0}^{N-1}(u_i - \mu_u)\,(v_i - \mu_v)$$

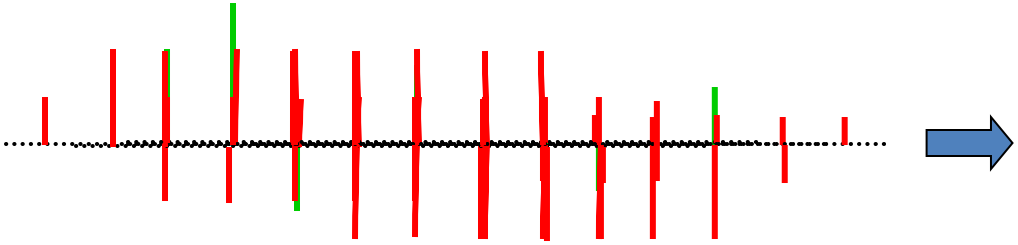


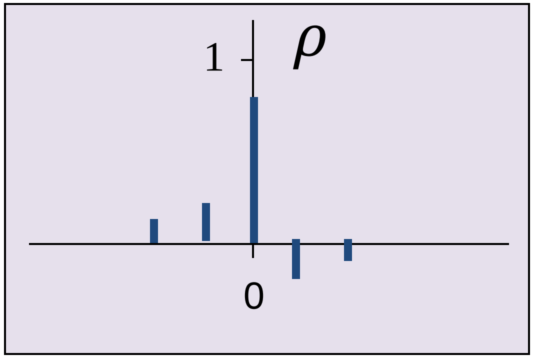
Correlation – shifting similarity

- Sometimes sequences are similar after applying a shift.

- Can be measured with **cross-correlation**

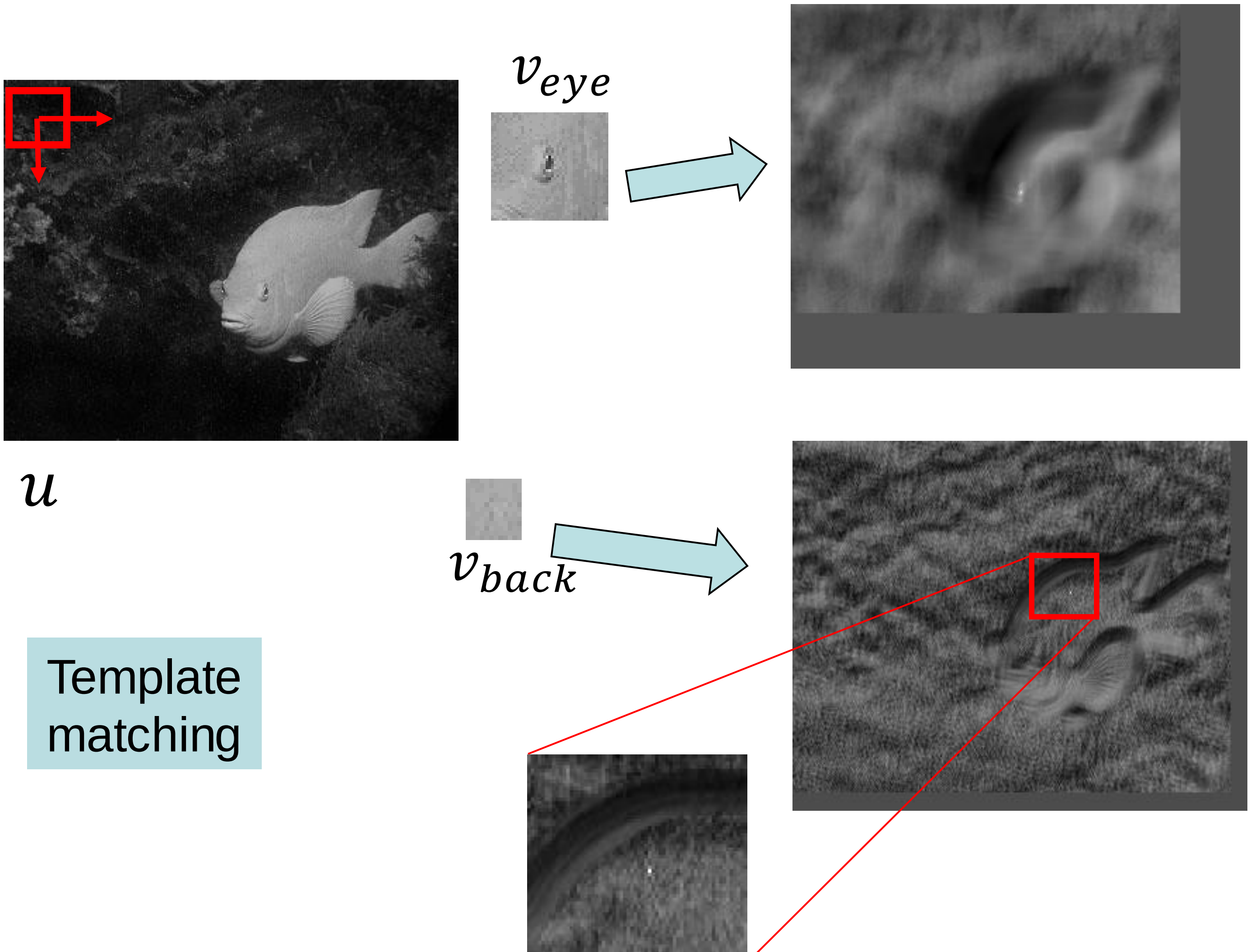
$$\rho = \frac{1}{N\sigma_u\sigma_v}\sum_{i=0}^{N-1}(u_i - \mu_u)(v_{i-j} - \mu_v)$$





'most likely shift' given by position of maximum value in cross-correlation

Example: Correlating Fish Parts!



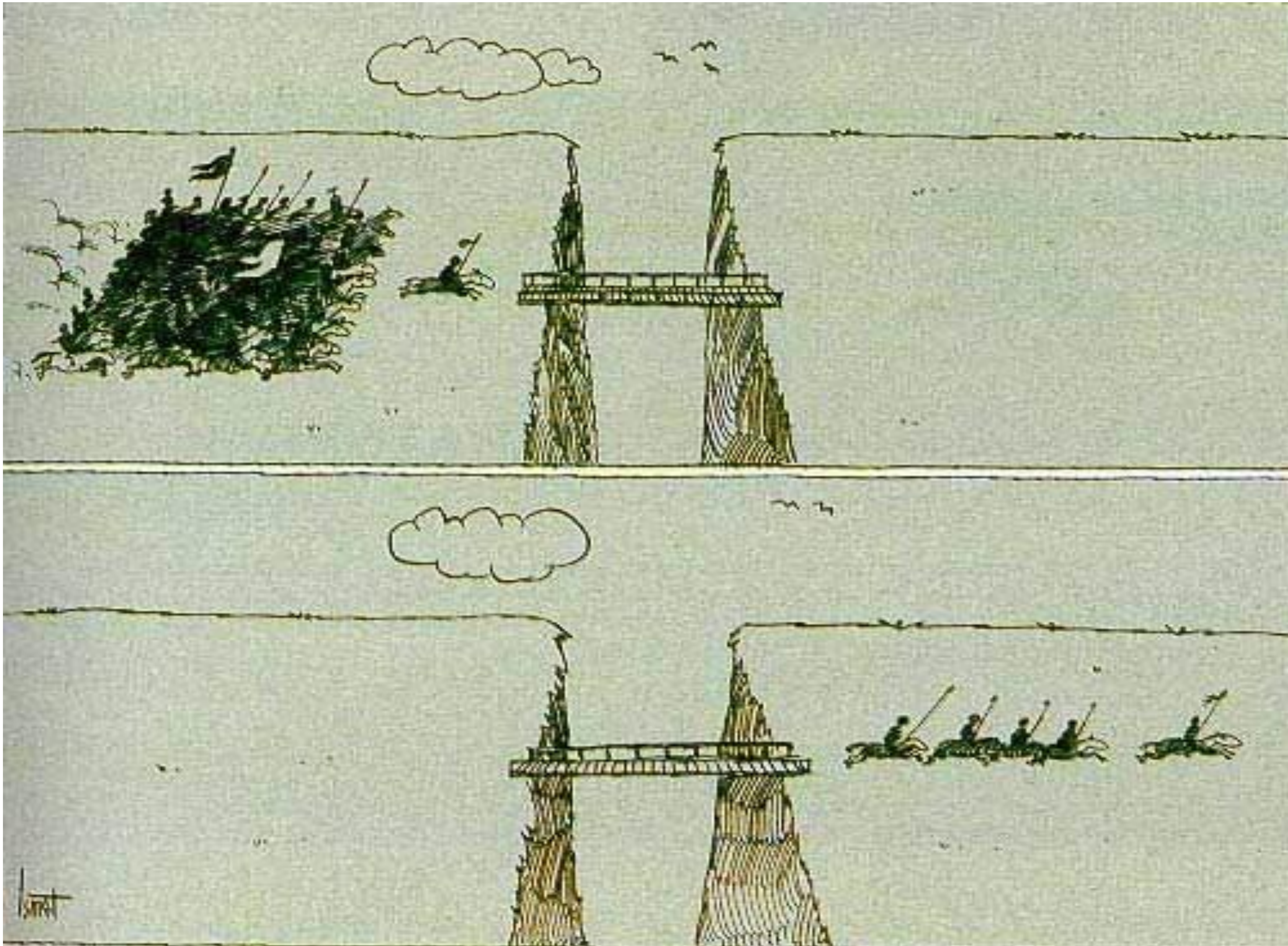
Template matching

Spatial Filtering

We can filter signals and symbols in the spatial/time domain:

- introduce some form of enhancement
  - remove noise/outliers
  - smoothing/averaging out detail
  - sharpening/highlighting detail
- prepare for next stage of processing
  - feature extraction

Filters are also referred to as *kernels* or *masks*.



Spatial Filtering

Many spatial filters are implemented with ***convolution*** masks.

To do convolution, we need to know about ***neighbourhoods***.

Symbolic Data

ATAGACATGGC .....

neighbours of G?

1D signal data

3	2	4	4	2	6

.....

2D signal data

3	2	4	4	2	6
3	4	5	4	3	6
4	2	5	4	3	3
3	0	4	1	2	6
3	2	4	5	2	6

Convolution mask is applied to each signal sample and its neighbourhood.

Convolution

$$g(x) = h(x)*f(x)$$

Convolution Kernel $h(x)$

1	2	1

Signal $f(x)$

1	1	2	2	1	1	0	1	2	2

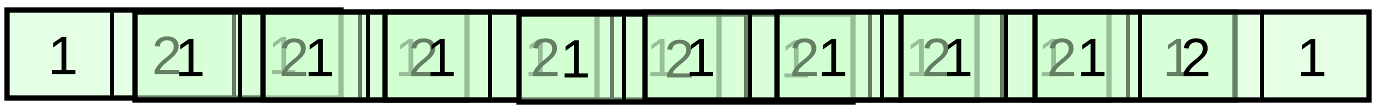
Filter Response $g(x)$

	$\frac{5}{4}$								

Convolution

$$g(x) = h(x) * f(x)$$

Convolution Kernel *h(x)*



Signal *f(x)*

1	1	2	2	1	1	0	1	2	2

Filter Response *g(x)*

☹	$\frac{5}{4}$	$\frac{7}{4}$	$\frac{7}{4}$	$\frac{5}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{4}{4}$	$\frac{7}{4}$	☹

Convolution

- f is the signal, h is the convolution filter
  - h has an origin

$$\frac{1}{5}\begin{bmatrix} -1 & 3 & -1 \end{bmatrix}$$

Example 1D kernel

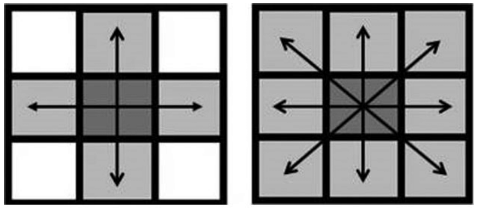
  - Normalization factor (sum of the absolute values of the filter) is also part of the filter!

$$g = f(t) * h(t) = \int_{-\infty}^{\infty} f(t-\tau)h(\tau)d\tau = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau$$

- The discrete version of convolution is defined as:

$$g(x) = \sum_{m=-s}^{s} f(x-m)h(m) \qquad \text{for } s \geq 1$$

2D Spatial Filtering - Connectivity



(a) (b)

Determine the connectivity for ***neighbourhoods:***

2D signal data

4–connectivity **8–connectivity**.

3	2	4	4	2	6
3	4	5	4	3	6
4	2	5	4	3	3
3	0	4	1	2	6
3	2	4	5	2	6

3	2	4	4	2	6
3	4	5	4	3	6
4	2	5	4	3	3
3	0	4	1	2	6
3	2	4	5	2	6

2D Convolution

➢ The discrete version of 2D convolution is defined as

$$g(x,y)=\sum_{m=-1}^{1}\sum_{n=-1}^{1} f(x-m,y-n)h(m,n)$$

Shorthand form:

$$g = f * h$$

Convolution symbol

h

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

f

		$y-1$	y	$y+1$	
$x-1$		**43**	**12**	**61**	
x		**44**	**45**	**60**	
$x+1$		**43**	**50**	**61**	

-68

$$\begin{aligned}
& f(x+1,y+1)h(-1,-1) \\
&+ f(x+1,y)h(-1,0) \\
&+ f(x+1,y-1)h(-1,1) \\
&+ f(x,y+1)h(0,-1) \\
&+ f(x,y)h(0,0) \\
&+ f(x,y-1)h(0,1) \\
&+ f(x-1,y+1)h(1,-1) \\
&+ f(x-1,y)h(1,0) \\
&+ f(x-1,y-1)h(1,1)
\end{aligned}$$

2D Correlation

- The discrete version of 2D correlation is defined as

$$g(x,y)=\sum_{m=-1}^{1}\sum_{n=-1}^{1} f(x+m,y+n)h(m,n)$$

h

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

f

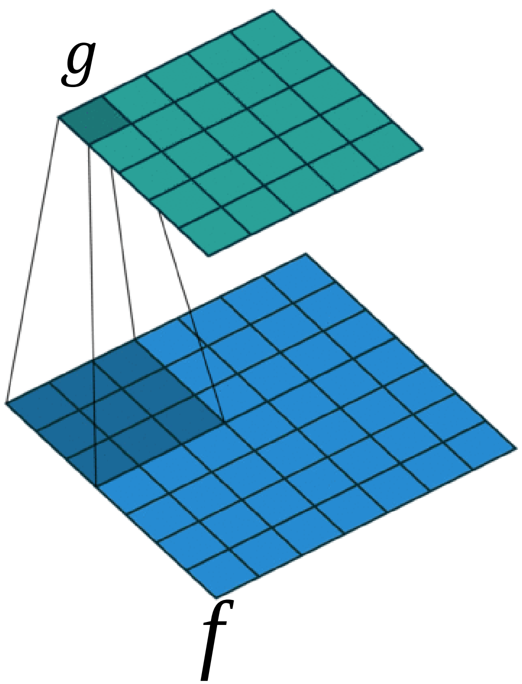
		$y-1$	y	$y+1$	
$x-1$		**43**	**12**	**61**	
x		**44**	**45**	**60**	
$x+1$		**43**	**50**	**61**	

68

Correlation=Convolution when kernel is symmetric under 180° rotation, e.g.

1	**2**	**1**

Spatial Filtering using 2D Convolution (actually Correlation)



Simple example

f

3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

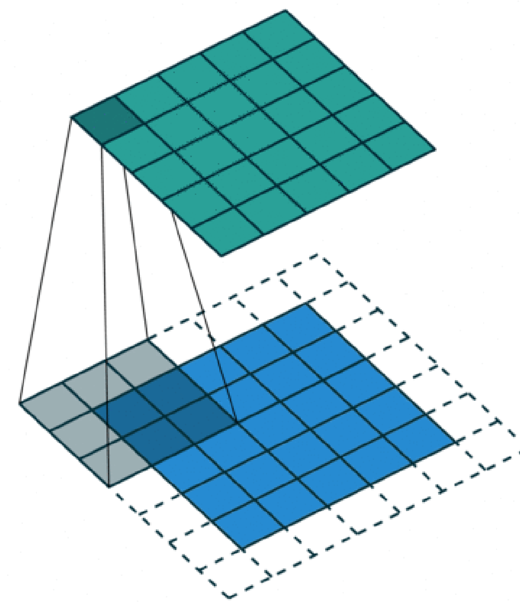
h

0	**1**	**2**
2	**2**	**0**
0	**1**	**2**

g

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Use padding for
same size result

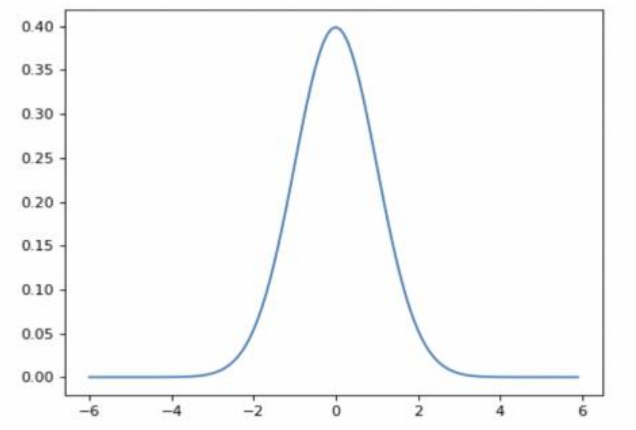


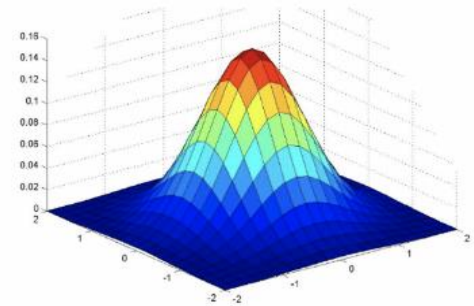
Example: Spatial Low/High Pass Filtering

- 1D: turning the treble/bass knob down on audio equipment!
- 2D: smooth/sharpen image

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

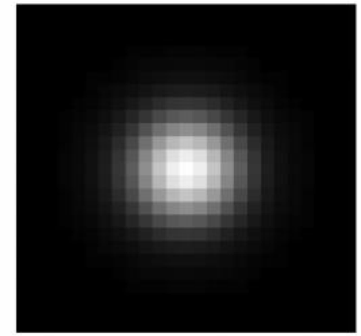
Gaussian Low Pass Filter





Gaussian kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



Example:

$\frac{1}{16}$ x

1	2	1
2	4	2
1	2	1

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \frac{1}{16}\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

Example: Edge Features

$$f(x, y)$$

- Edges occur in images where there is discontinuity (or change) in the intensity function.

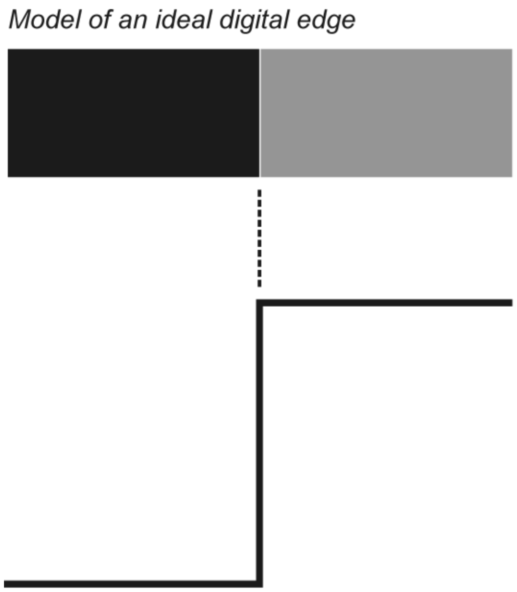
- Gradient points in the direction of most rapid change in intensity

- Biggest change → derivative has maximum magnitude.

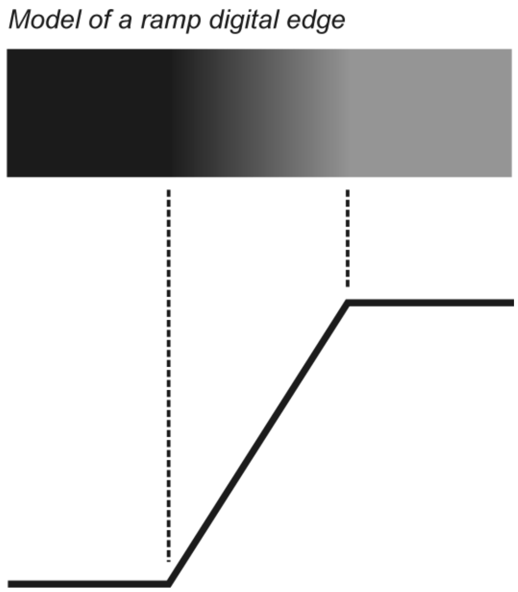
$$\left\|\nabla f\right\| = \sqrt{\left[\frac{\partial f}{\partial x}\right]^2 + \left[\frac{\partial f}{\partial y}\right]^2}$$

Edge detection: basic concept

- Edge: a boundary between two image regions having distinct characteristics according to some feature (e.g., gray level, color, or texture).
- In grayscale 2D images: a significant variation of the intensity function across a portion of the image.



Gray level profile of a horizontal line through the image

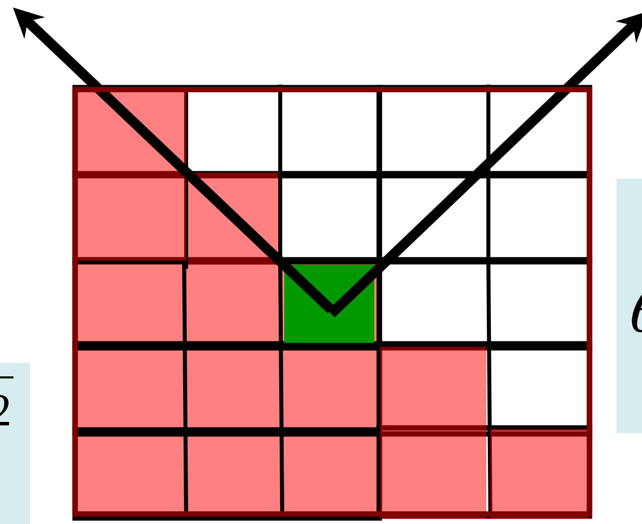


Gray level profile of a horizontal line through the image

Edge Measures

Edge direction

$$\phi = \theta - \frac{\pi}{2}$$



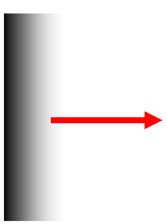
gradient direction

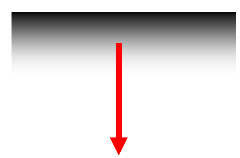
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

Edge magnitude

$$\left\| \nabla f \right\| = \sqrt{\left[\frac{\partial f}{\partial x} \right]^2 + \left[\frac{\partial f}{\partial y} \right]^2}$$

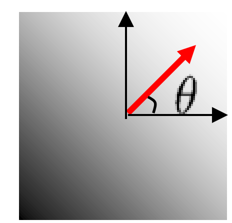
$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$$





$$\nabla f = \left[0, \frac{\partial f}{\partial y} \right]$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$



Sobel Edge Detector

- 2D gradient measurement in two different directions.
- Uses these 3x3 convolution masks:

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

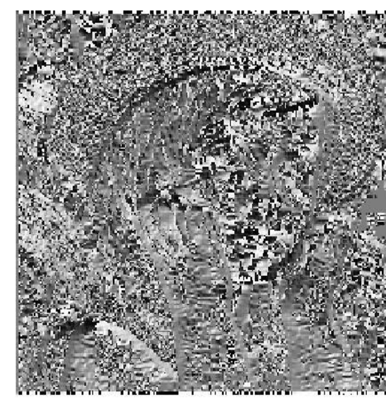
$*$

$\frac{\partial f}{\partial x} =$

Thresholded result

$\|\nabla f\|$

θ

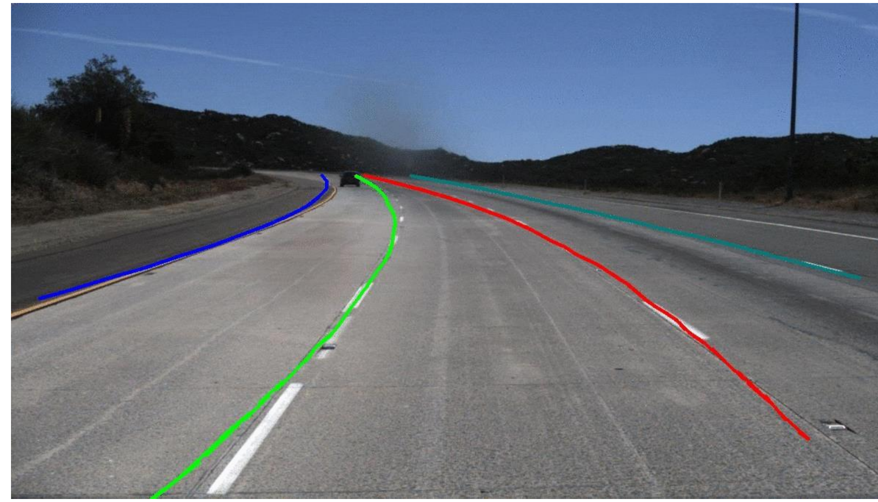


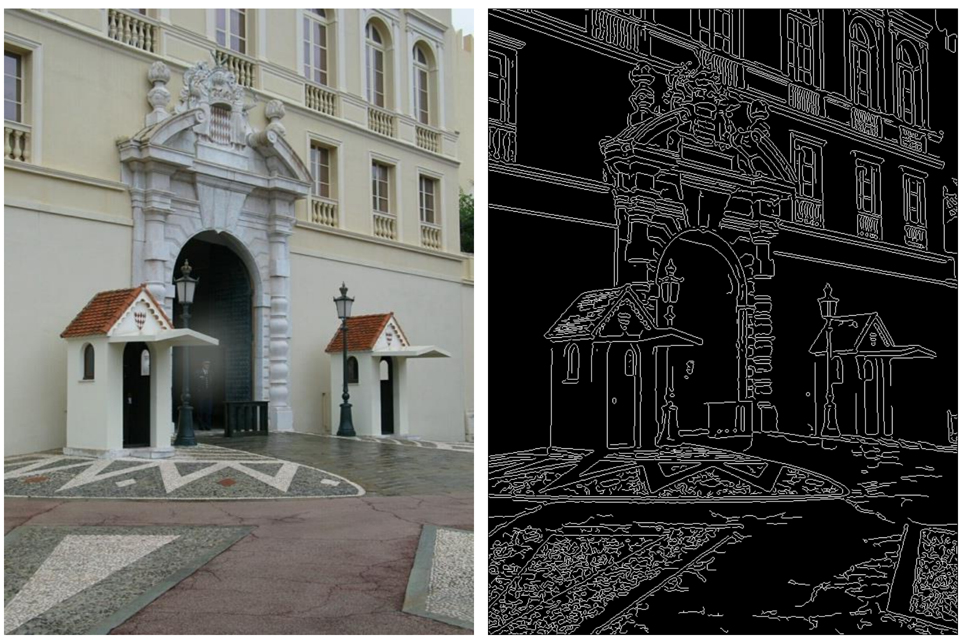
+1	+2	+1
0	0	0
-1	-2	-1

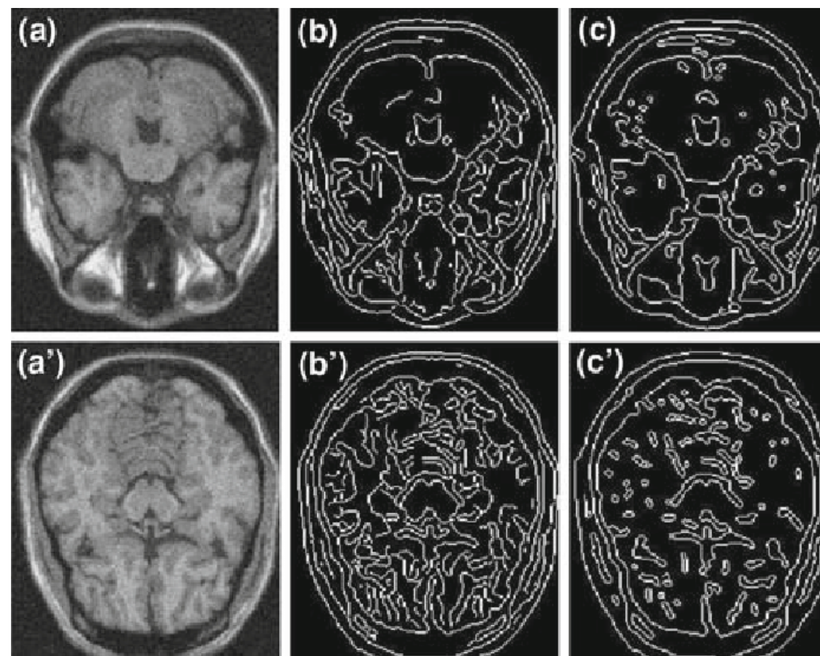
$*$

$\frac{\partial f}{\partial y} =$

Examples

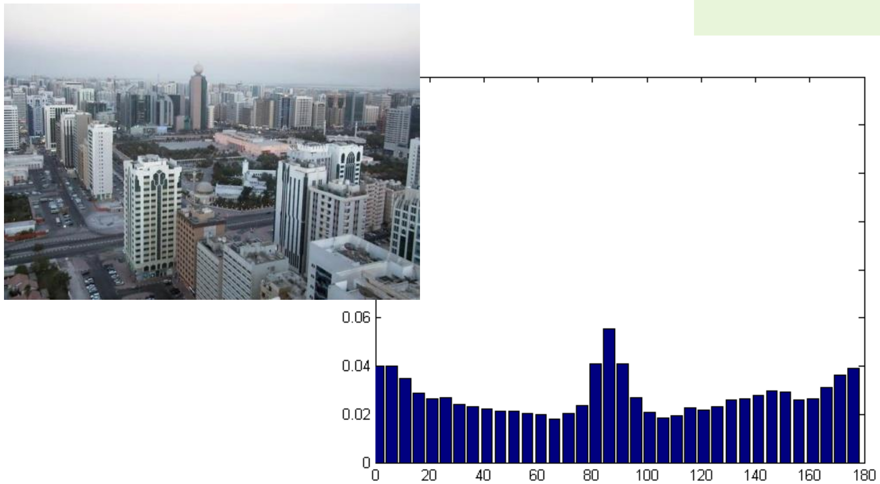


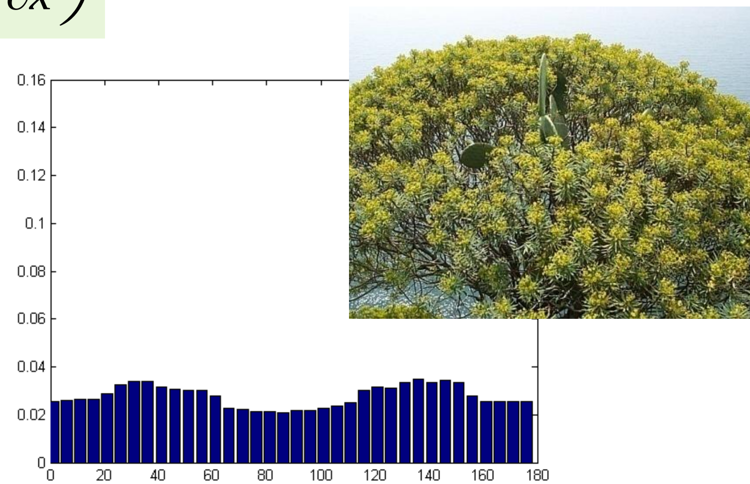


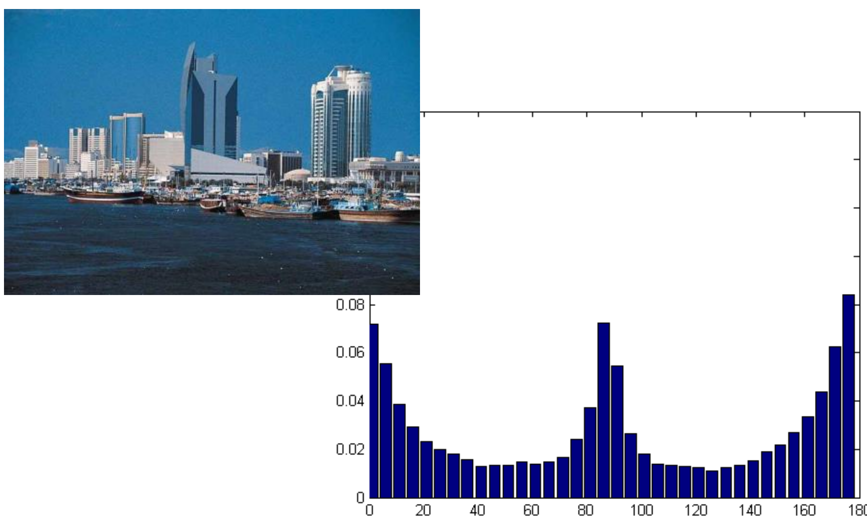


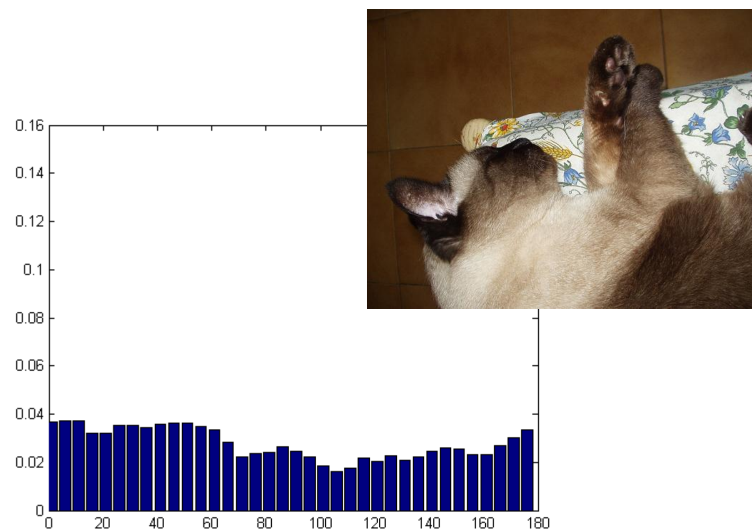
Histogram of Edge Gradient Directions

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$









Matlab: Sobel Edge Detection

```
% Sobel edge detection
A = imread('romina.gif');
fx = [-1 0 1; -2 0 2; -1 0 1]
fy = [1 2 1; 0 0 0; -1 -2 -1]
gx = conv2(double(A),double(fx))/8;
gy = conv2(double(A),double(fy))/8;
mag = sqrt((gx).^2+(gy).^2);
ang = atan(gy./gx);
figure; imagesc(mag); axis off; colormap gray
figure; imagesc(ang); axis off; colormap gray
```

See unit github page for code in Python.

Spatial/Frequency Domain Filtering

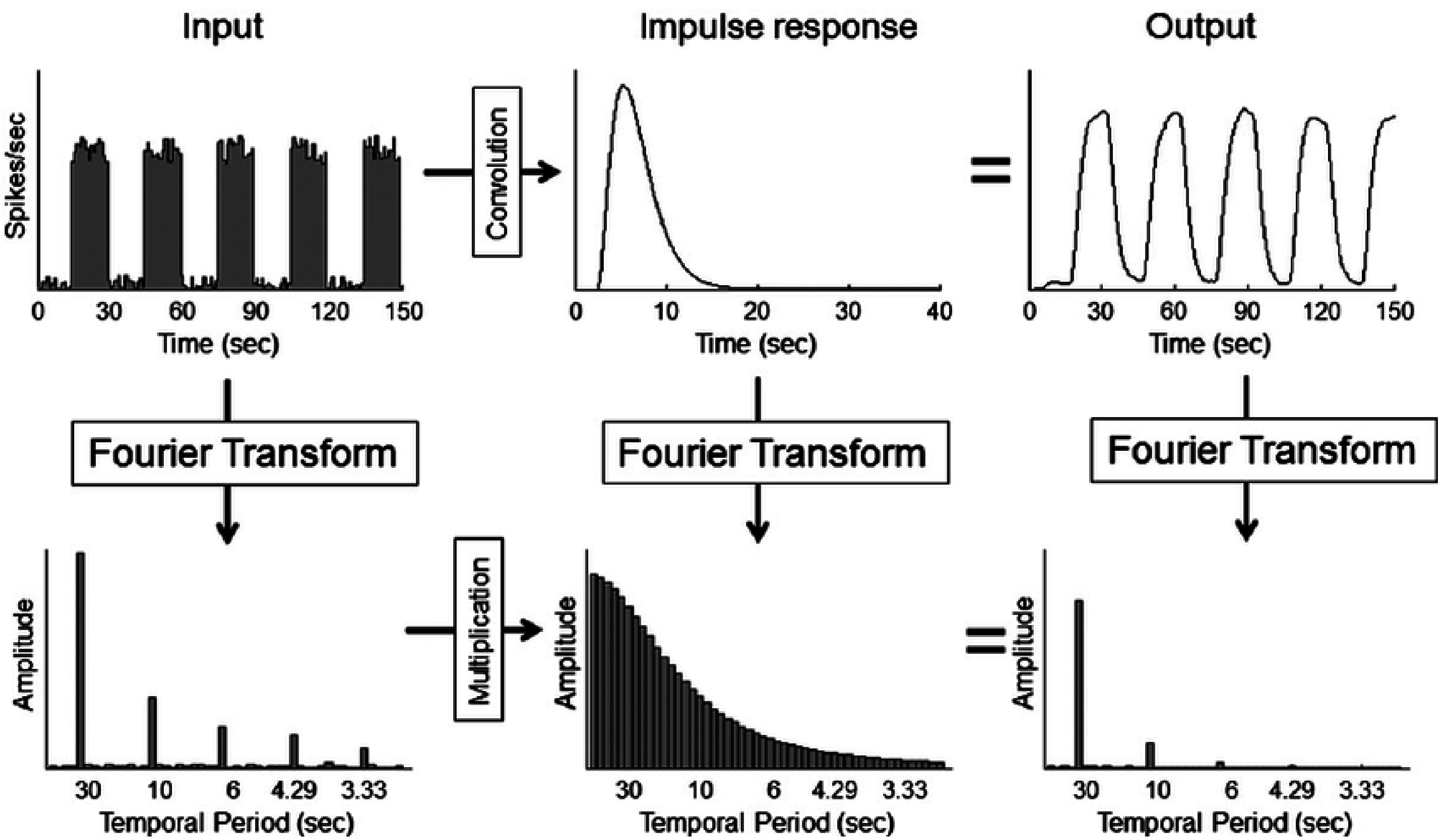
- Convolution Theorem:

Convolution in the spatial domain
is equivalent to
multiplication in the frequency domain
and vice versa

$$g(x) = f(x) * h(x) \quad \Leftrightarrow \quad G(u) = F(u)\, H(u)$$

$$g(x) = f(x)\, h(x) \quad \Leftrightarrow \quad G(u) = F(u) * H(u)$$

Example: Convolution in SD is Multiplication in FD

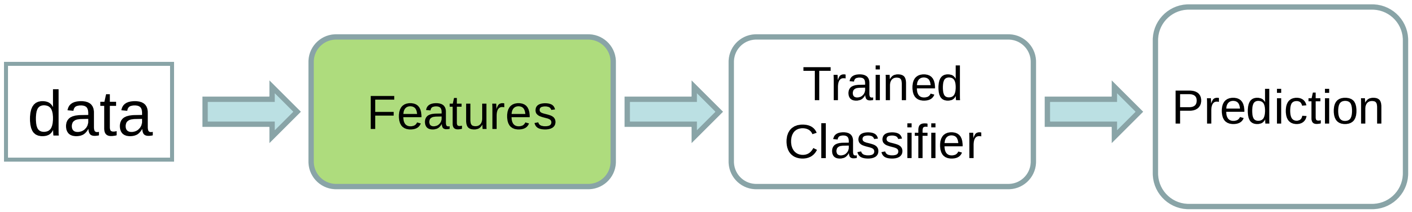


Convolution Exercise

$$P = \begin{pmatrix} -1 & 4 & -1 \\ 4 & 10 & 4 \\ -1 & 4 & -1 \end{pmatrix} \qquad Q = \begin{pmatrix} 5 & 3 & 0 & 0 & 7 \\ 3 & 3 & 2 & 1 & 4 \\ 6 & 5 & 1 & 1 & 6 \\ 6 & 3 & 4 & 1 & 4 \\ 4 & 4 & 5 & 0 & 4 \end{pmatrix}$$

Apply convolution filter P to matrix Q at location (2,3).

Well Done for Lasting This Far!



Feature Selection and Extraction

- Signal basics and Fourier Series
- 1D and 2D Fourier Transform
- Another look at features
- PCA for dimensionality reduction
- Convolutions