Fundamentals of Computer Vision

Unit 4: Linear Filtering

Jorge Bernal

Index

01

1. Signals

02

2. Linear systems. Convolution

03

3. Linear filters

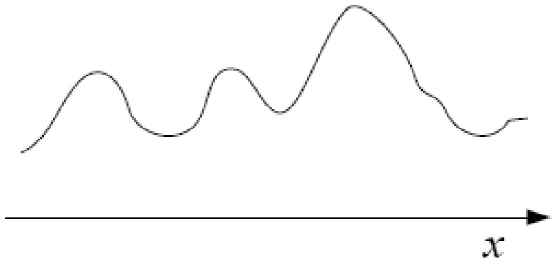
04

4. Fourier Transform

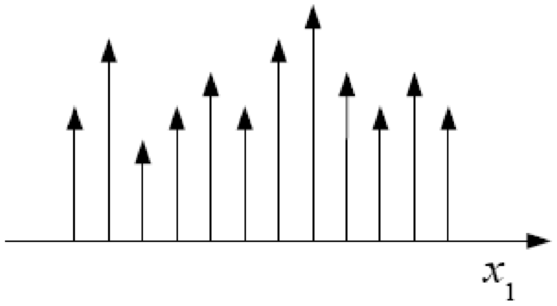
1

Signals

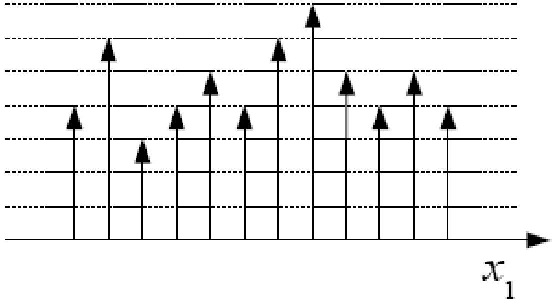
Signals



- Continuous signal



- Discrete space signal $f(x_1)$



- Discrete signal $f(x_1)$ - quantized

2D Signals

Mathematically, what is a discrete 2D signal?

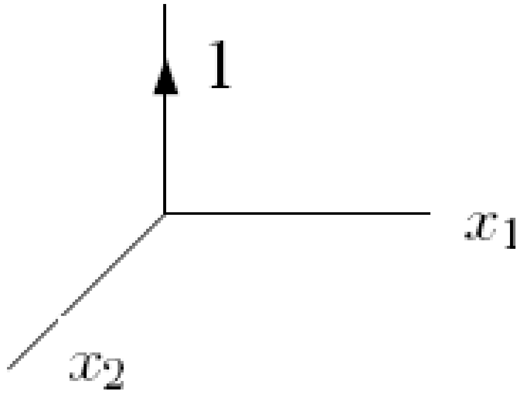
- An infinite sequence, defined at integer coordinates:

$$f(x_1, x_2), x_1, x_2 \in \mathbf{Z}$$

2D Signals

Dirac delta function (impulse unit)

$$\delta(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 = x_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$



2D Signals

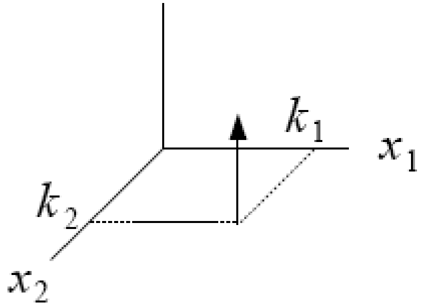
Dirac delta function (impulse unit):

- Any signal can be decomposed as a linear combination (weighted sum) of shifted impulses

$$f(x_1, x_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)\delta(x_1 - k_1, x_2 - k_2)$$

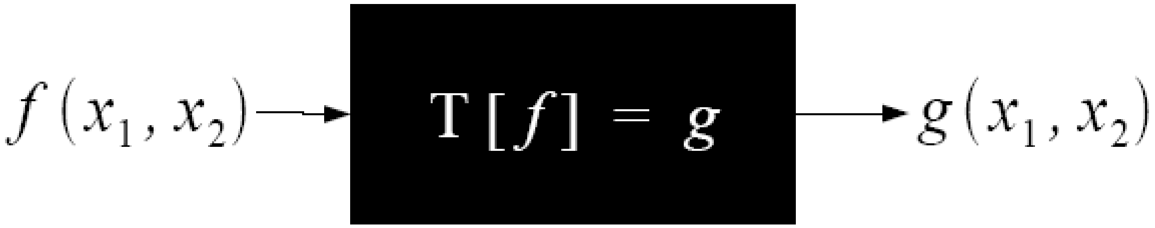
Shifted unit impulses

weights
(values of f at (k_1,k_2))



Systems

What is a system?



$f(x_1, x_2)$

$T[f] = g$

$g(x_1, x_2)$

T can be a lot of things. We are interested in the following features:

1. Simple (= easy to study, characterize and compute)
2. Ability to represent interesting transformations
3. Model real transformations applied to signals

Linear Shift-invariant Systems

A system is **linear** if:

$$T[af(x_1,x_2)+bg(x_1,x_2)] = aT[f(x_1,x_2)]+bT[g(x_1,x_2)]$$

The output for a linear combination of input signals = the same linear combination of the outputs for each of the input signals

A systems is **shift-invariant** if it “does the same anywhere”

$$T[f(x_1,x_2)] = g(x_1,x_2) \Longrightarrow$$

$$T[f(x_1-M,x_2-N)] = g(x_1-M,x_2-N)$$

The output for a shifted input signal = The output of the non-shifted signal shifted in the same way

2

Linear systems. Convolution

Convolution

$$\begin{aligned}
g(x_1,x_2) &= T[f(x_1,x_2)] = \boxed{T}[\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} f(k_1,k_2)\delta(x_1-k_1,x_2-k_2)] \\
&= \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} f(k_1,k_2)T[\delta(x_1-k_1,x_2-k_2)] = \\
&= \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} f(k_1,k_2)h(x_1-k_1,x_2-k_2) = \\
&= f * h\ (x_1,x_2)
\end{aligned}$$

Any given signal can be expressed as a linear combination of Dirac deltas

Convolution

$$
\begin{aligned}
g(x_1, x_2) &= T[f(x_1, x_2)] = T[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)\delta(x_1 - k_1, x_2 - k_2)] \\
&\boxed{=} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) T[\delta(x_1 - k_1, x_2 - k_2)] = \\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) h(x_1 - k_1, x_2 - k_2) = \\
&= f * h \; (x_1, x_2)
\end{aligned}
$$

Linearity

Convolution

$$
\begin{aligned}
g(x_1, x_2) &= T[f(x_1, x_2)] = T[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)\delta(x_1 - k_1, x_2 - k_2)] \\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) T[\delta(x_1 - k_1, x_2 - k_2)] = \\
&\boxed{=} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) h(x_1 - k_1, x_2 - k_2) = \\
&= f * h \ (x_1, x_2)
\end{aligned}
$$

h is defined as the centered unitary impulse response of the system

$$T[\delta(x_1, x_2)] = h(x_1, x_2)$$

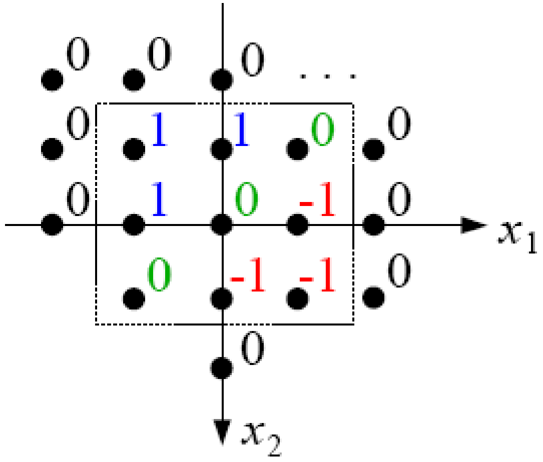
Convolution

$$\begin{aligned}
g(x_1, x_2) &= T[f(x_1, x_2)] = T[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)\delta(x_1 - k_1, x_2 - k_2)] \\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)T[\delta(x_1 - k_1, x_2 - k_2)] = \\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)h(x_1 - k_1, x_2 - k_2) = \\
&\boxed{=} \; f * h \; (x_1, x_2)
\end{aligned}$$

Definition of the convolution operator (*)

Convolution

$h(x_1, x_2)$



1	1	0
1	0	-1
0	-1	-1

$$\delta(x_1, x_2) \longrightarrow \boxed{\mathrm{T}\,[f] = g} \longrightarrow h(x_1, x_2)$$

Convolution

$$\begin{tikzcd} f(x_{1,}x_2) \arrow[r] & \boxed{*h(x_{1,}x_2)} \arrow[r] & g(x_{1,}x_2)=f*h(x_1,x_2) \end{tikzcd}$$

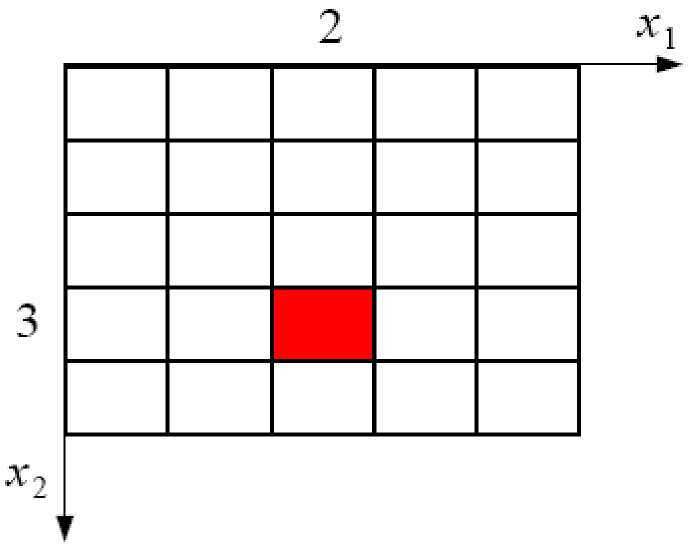
$$g(x_1,x_2) \;=\; \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} f(k_1,k_2)h(x_1-k_1,x_2-k_2)$$

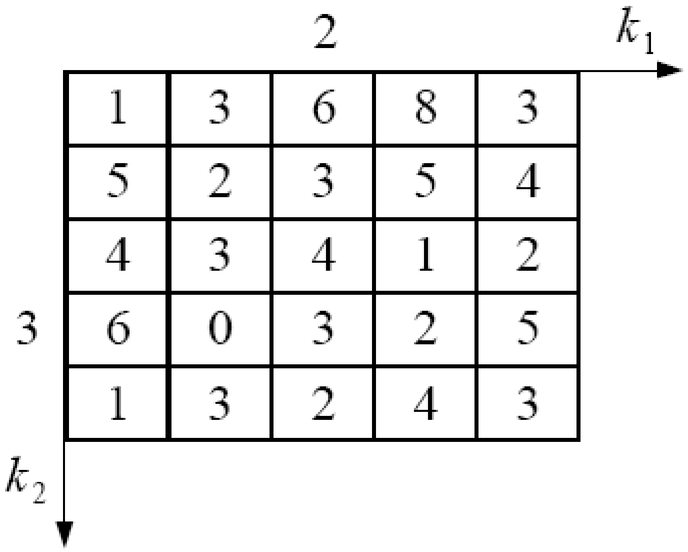
$$\begin{array}{|c|c|c|c|c|}\hline 1 & 3 & 6 & 8 & 3 \\ \hline 5 & 2 & 3 & 5 & 4 \\ \hline 4 & 3 & 4 & 1 & 2 \\ \hline 6 & 0 & 3 & 2 & 5 \\ \hline 1 & 3 & 2 & 4 & 3 \\ \hline \end{array} \;*\; \begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & -1 & -1 \\ \hline \end{array} \;=\; ?$$

Convolution

$$g(x_1, x_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) h(x_1 - k_1, x_2 - k_2)$$

Fix $(x_1,x_2) = (2,3)$ as example. Now, (k_1,k_2) can vary.



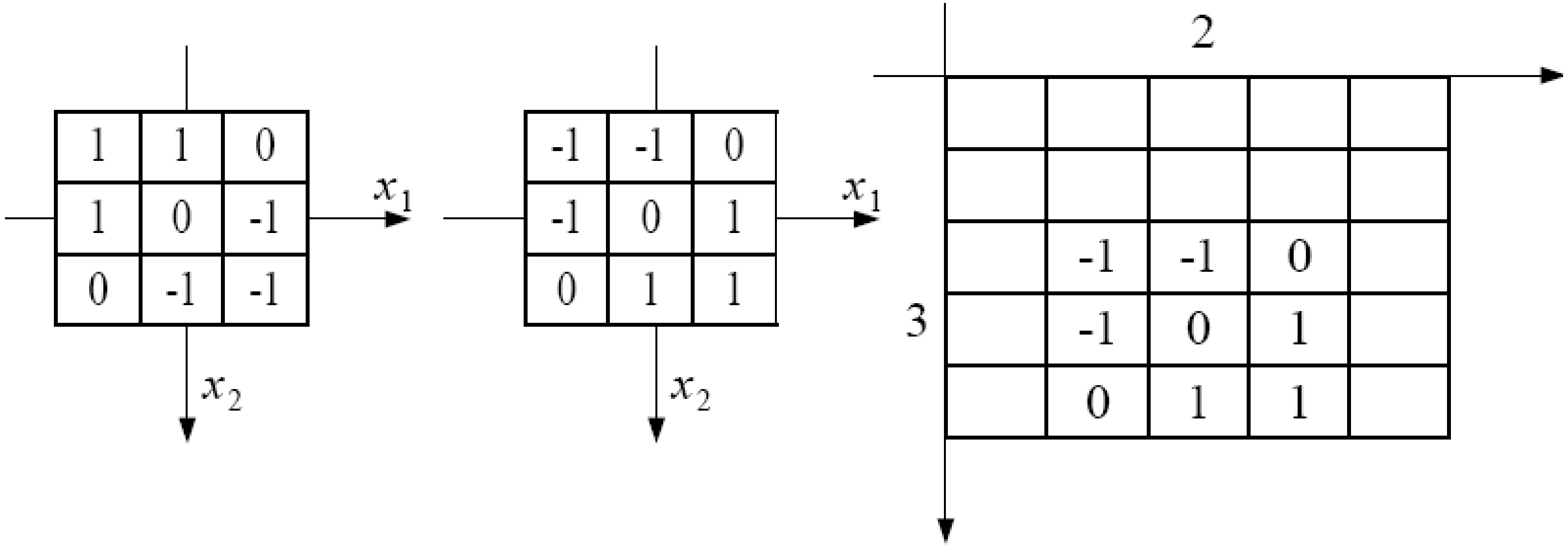


1	3	6	8	3
5	2	3	5	4
4	3	4	1	2
6	0	3	2	5
1	3	2	4	3

Convolution

$$g(x_1, x_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) h(x_1 - k_1, x_2 - k_2)$$

Fix (x_1,x_2) = (2,3) as example. Now, (k_1,k_2) can vary.



1	1	0
1	0	-1
0	-1	-1

$h(k_1, k_2)$

-1	-1	0
-1	0	1
0	1	1

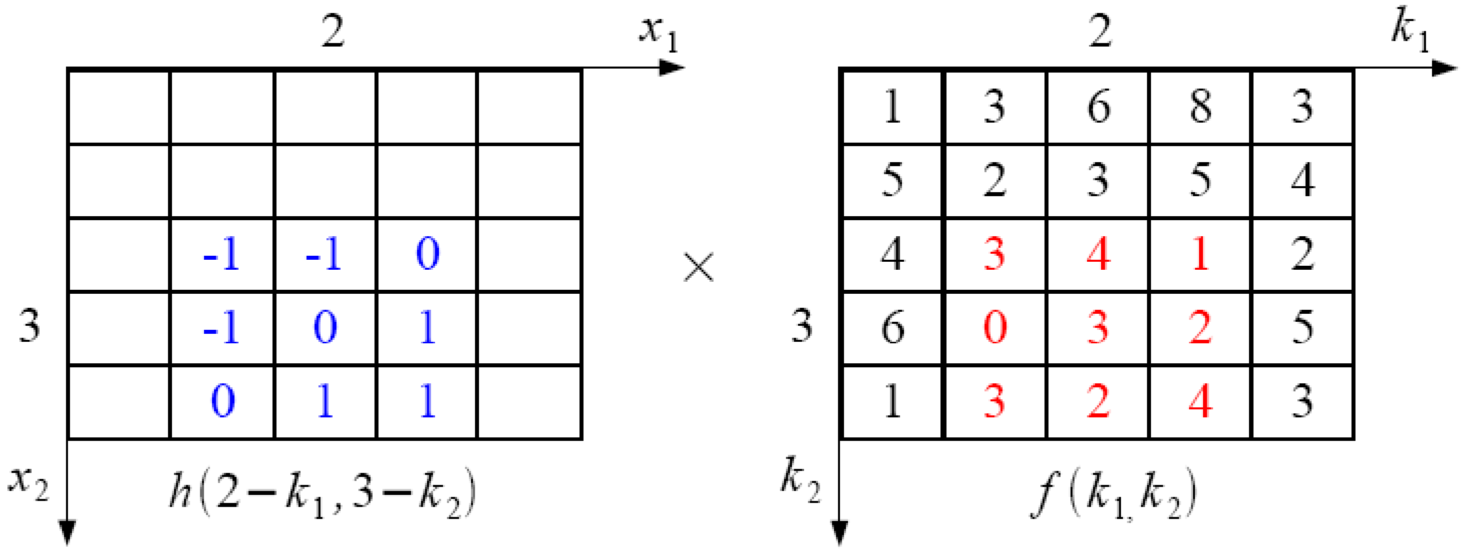
$h(-k_1, -k_2)$

	-1	-1	0	
	-1	0	1	
	0	1	1	

$h(x_1 - k_1, x_2 - k_2)$

Convolution

$$g(x_1, x_2) = \boxed{\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \boxed{f(k_1, k_2)h(x_1 - k_1, x_2 - k_2)}}$$



$h(2-k_1, 3-k_2)$

	-1	-1	0	
	-1	0	1	
	0	1	1	

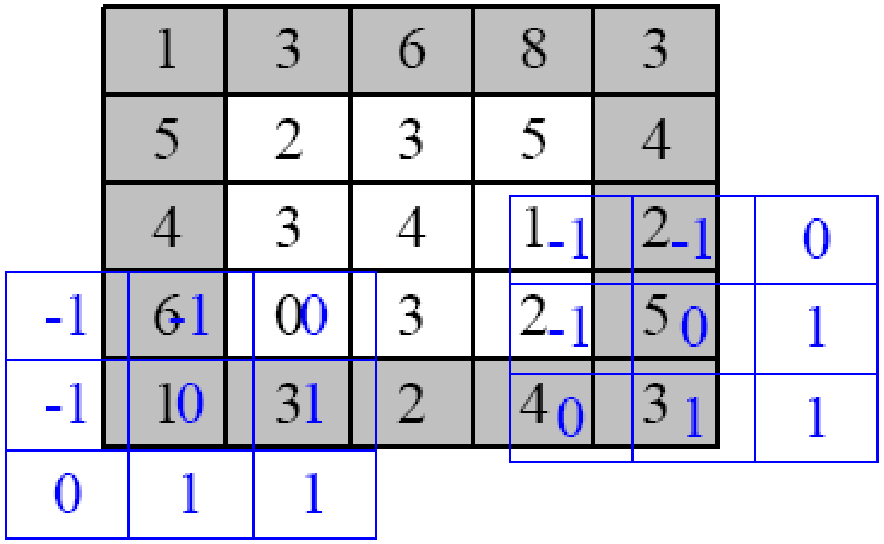
$f(k_1, k_2)$

1	3	6	8	3
5	2	3	5	4
4	3	4	1	2
6	0	3	2	5
1	3	2	4	3

$$\begin{aligned} g(2, 3) = {} & (-1 \times 3) + (-1 \times 4) + (0 \times 1) + \\ & (-1 \times 0) + (0 \times 3) + (1 \times 2) + \\ & (0 \times 3) + (1 \times 2) + (1 \times 4) = 1 \end{aligned}$$

Convolution

What happens at the borders?



$h(4-k_1, 3-k_2)$

$h(0-k_1, 4-k_2)$

Convolution

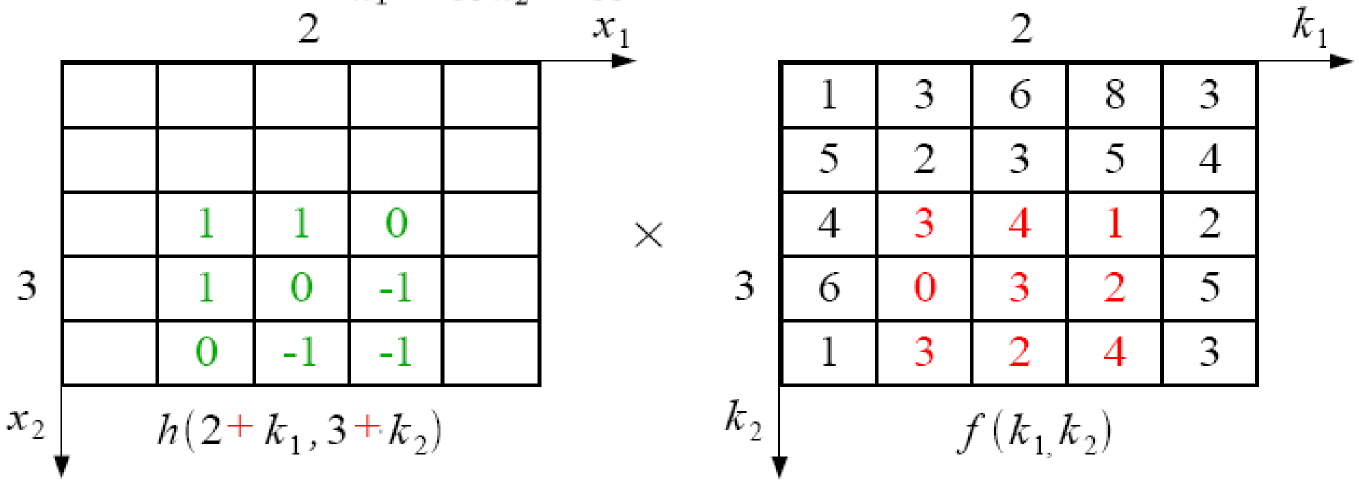
Why convolution operator is so important?

- Allows us to characterize any linear shift-invariant through its impulsive response h.
  - Characterize = compute, have only one property to define it.
- Very simple systems (products and additions).
- Field very well studied (signal processing).
- By changing h we can have several and very different behaviors.
  - Some of them (useful): to finding contours, reducing noise, pattern matching.
- Allow us to model signal degradations, for example: defocused images.
  - Needed if we want to remove these degradations.

Correlation

Similar to convolution but **without reflecting** the h kernel

$$g(x_1,x_2) = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} f(k_1,k_2)h(x_1+k_1,x_2+k_2)$$



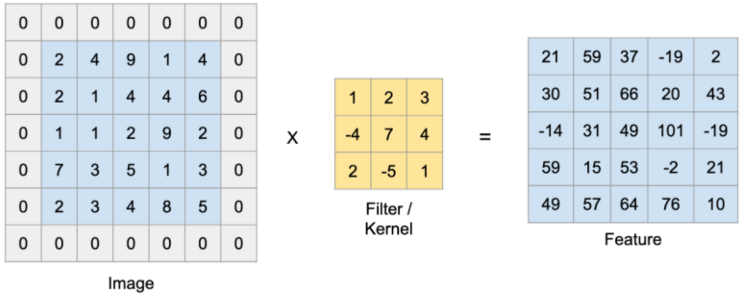
	1	1	0	
	1	0	-1	
	0	-1	-1	

1	3	6	8	3
5	2	3	5	4
4	3	4	1	2
6	0	3	2	5
1	3	2	4	3

$$\begin{aligned} g(2\,,3) = {} & (1\times 3)+(1\times 4)+(0\times 1)+ \\ & (1\times 0)+(0\times 3)+(-1\times 2)+ \\ & (0\times 3)+(-1\times 2)+(-1\times 4) = -1 \end{aligned}$$

Some interesting concepts

- Padding: addition of extra pixels around the boundary
  - Which value can we use: 0 (most common), any value, symmetric, circular



0	0	0	0	0	0	0
0	2	4	9	1	4	0
0	2	1	4	4	6	0
0	1	1	2	9	2	0
0	7	3	5	1	3	0
0	2	3	4	8	5	0
0	0	0	0	0	0	0

Image

x

1	2	3
-4	7	4
2	-5	1

Filter / Kernel

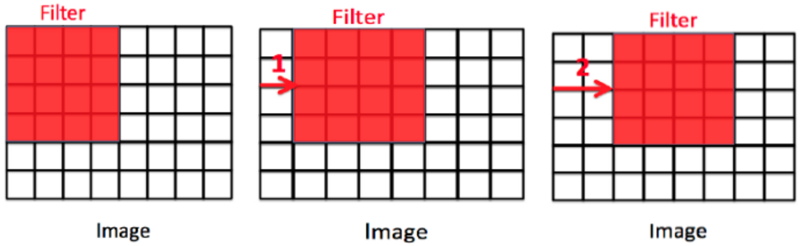
=

21	59	37	-19	2
30	51	66	20	43
-14	31	49	101	-19
59	15	53	-2	21
49	57	64	76	10

Feature

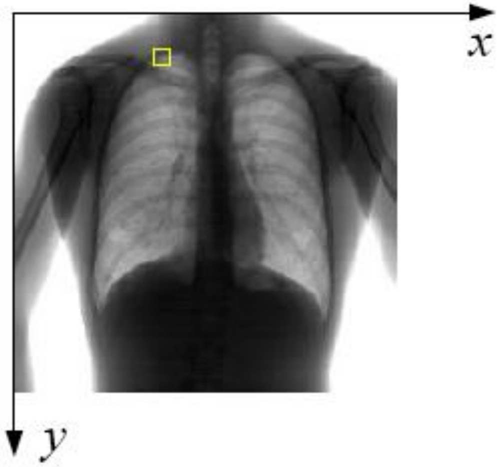
- Output size: same, full, valid

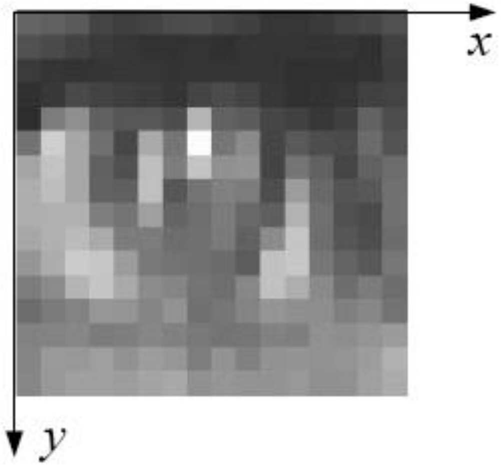
- Stride (used in Deep Learning): skip intermediate locations in a convolution



- À trous (scale, wavelets, DL) : the convolution kernels increases its size adding intermediate zeros.

Images as 2D signals

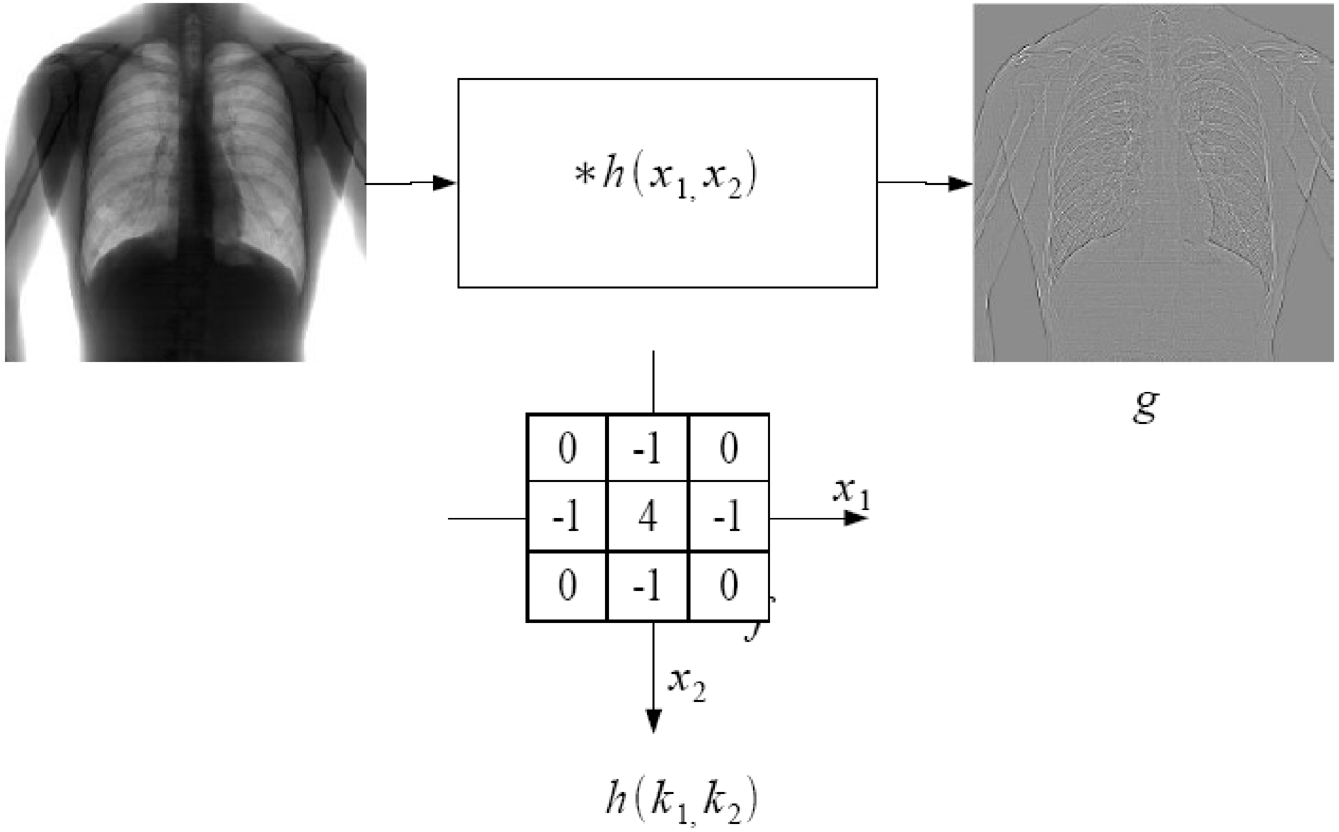




- Images can be seen as digital 2d signals (discrete spacing and quantized values).

- We can think that outside a certain range:
  - image values are zero
  - the image (signal, function) is not defined
  - the image has some periodicity
  - We simply don’t care

Image convolution



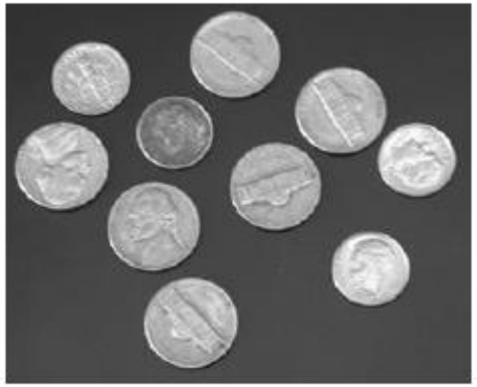
0	-1	0
-1	4	-1
0	-1	0

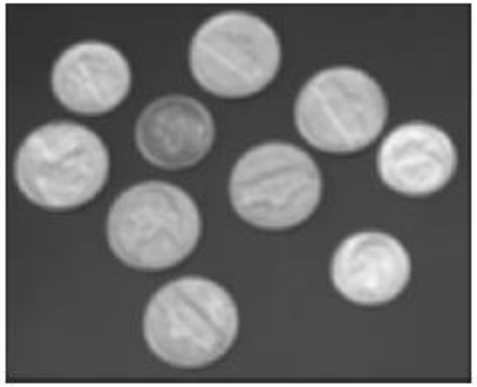
3

Linear filters

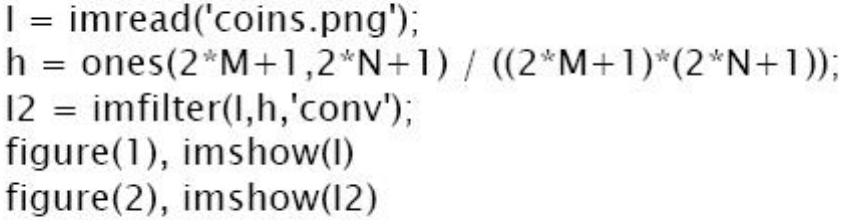
Smoothing

Reduction of the local variations of intensity, often due to acquisition noise





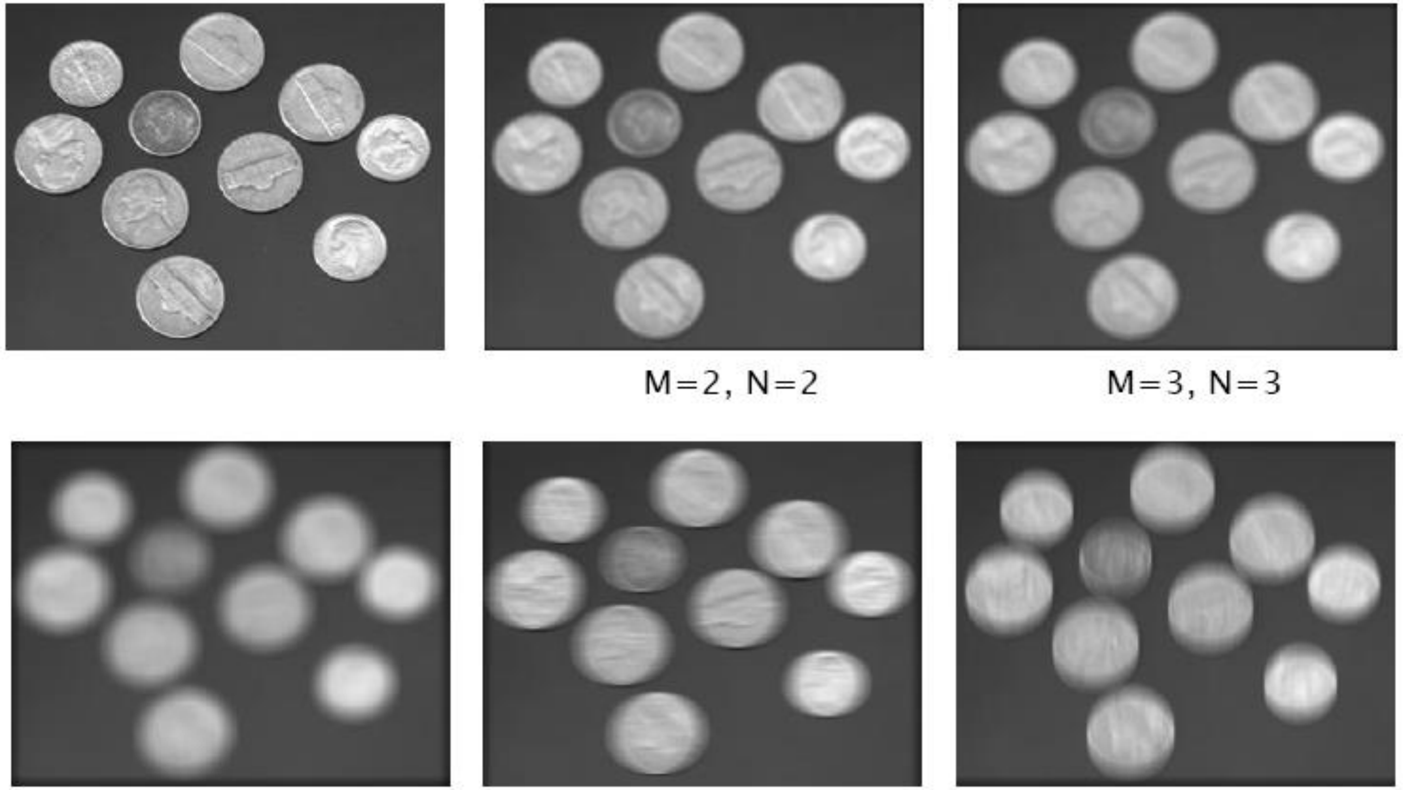
M=2, N=2



```
I = imread('coins.png');
h = ones(2*M+1,2*N+1) / ((2*M+1)*(2*N+1));
I2 = imfilter(I,h,'conv');
figure(1), imshow(I)
figure(2), imshow(I2)
```

Smoothing

Reduction of the local variations of intensity, often due to acquisition noise



M=2, N=2

M=3, N=3

M=7, N=7

M=0, N=7

M=7, N=0

Edges

Mark local variations of intensity when they are motivated by **object contours**

$$
\begin{aligned}
\frac{\partial f(x,y)}{\partial x} &\approx \frac{f(x,y) - f(x - \Delta x, y)}{\Delta x} \\
&\approx \frac{f(x,y) + f(x - \Delta x, y)}{\Delta x} \\
&\approx \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2\Delta x}
\end{aligned}
$$

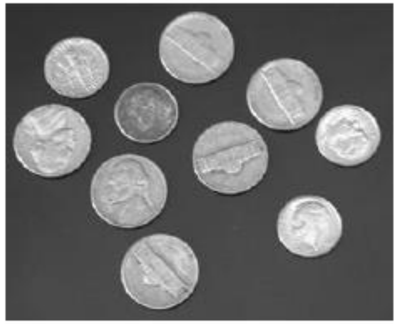
$$\Delta x = 1$$

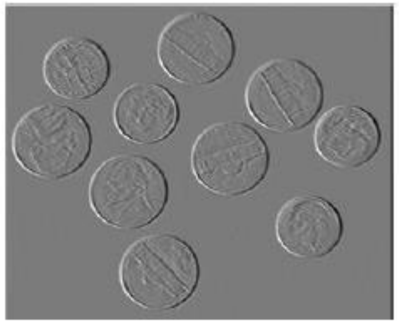
$$
\begin{aligned}
D_x^- &= [\,1_\bullet \;\; -1\,] & \quad f * D_x^- &= f(x,y) - f(x-1,y) \\
D_x^+ &= [\,1 \;\; -1_\bullet\,] & \quad f * D_x^+ &= f(x,y) - f(x-1,y) \\
D_x^s &= \frac{1}{2}\,[\,1 \;\; 0 \;{}_\bullet\, -1\,] & \quad f * D_x^s &= [f(x+1,y) - f(x-1,y)]/2
\end{aligned}
$$

● shows (0,0) origin.

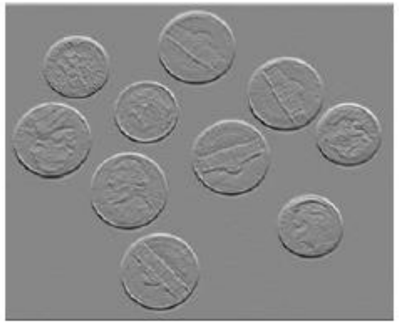
Edges

Mark local variations of intensity when they are motivated by **object contours**



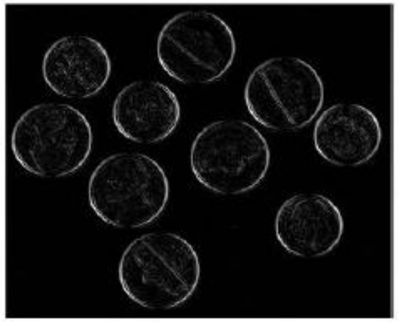


$f * D_x^s$

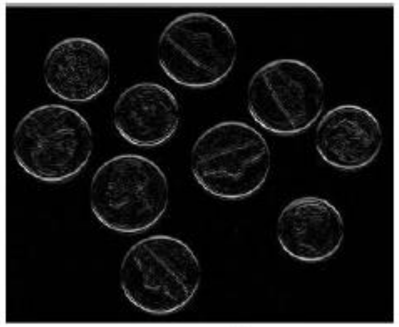


$f * D_y^s$

```
h=0.5*[1  0  -1];
I2=conv2(double(I),h);
figure,imshow(I2,[])
figure,imshow(abs(I2),[])
h=h';
I2=conv2(double(I),h);
figure,imshow(I2,[])
```



$|f * D_x^s|$



$f * D_y^s$

More linear filters

- Gradient

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Magnitude

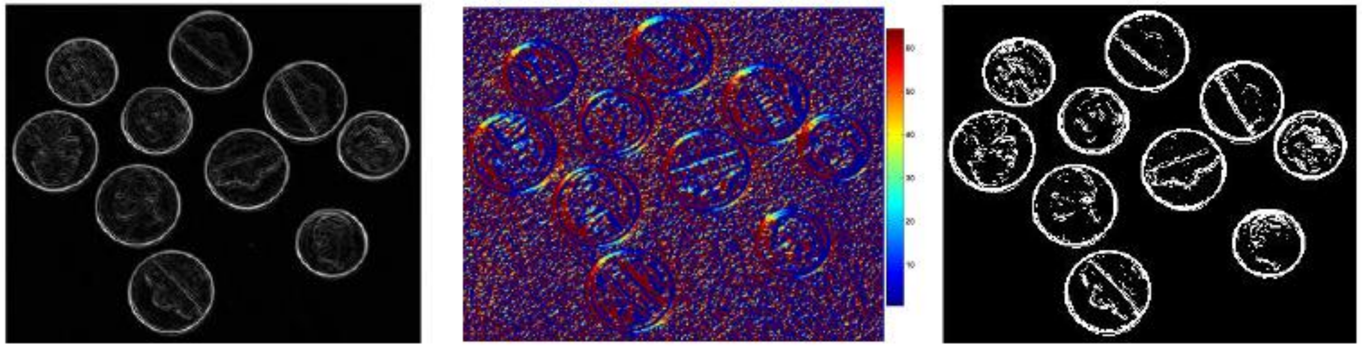
$$|\nabla f|(x,y) = \Big[\Big(\frac{\partial f}{\partial x}(x,y)\Big)^2 + \Big(\frac{\partial f}{\partial y}(x,y)\Big)^2\Big]^{1/2}$$

$$|\nabla f| \approx |D_x^s| + |D_y^s|$$

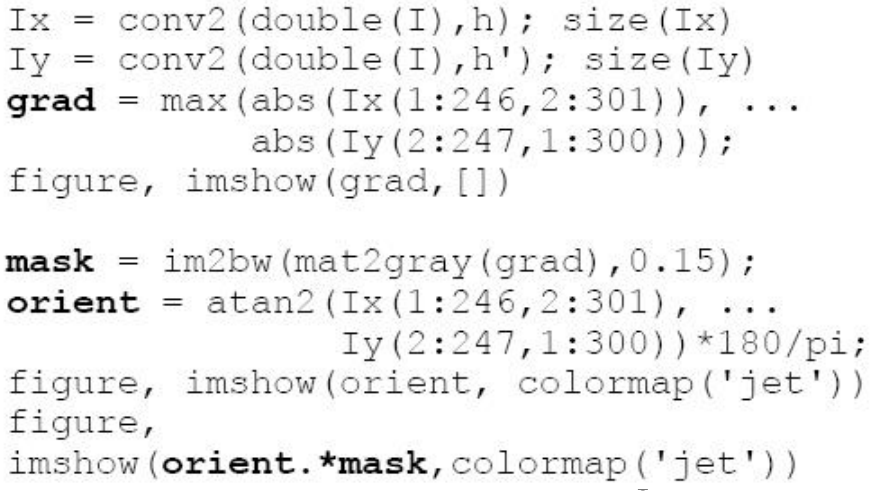
- Orientation

$$\phi(x,y) = \arctan\Big(\frac{\partial f}{\partial y}(x,y) \Big/ \frac{\partial f}{\partial x}(x,y)\Big)$$

More linear filters

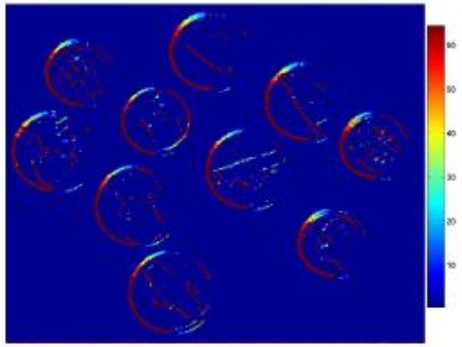


grad **orient** **mask**



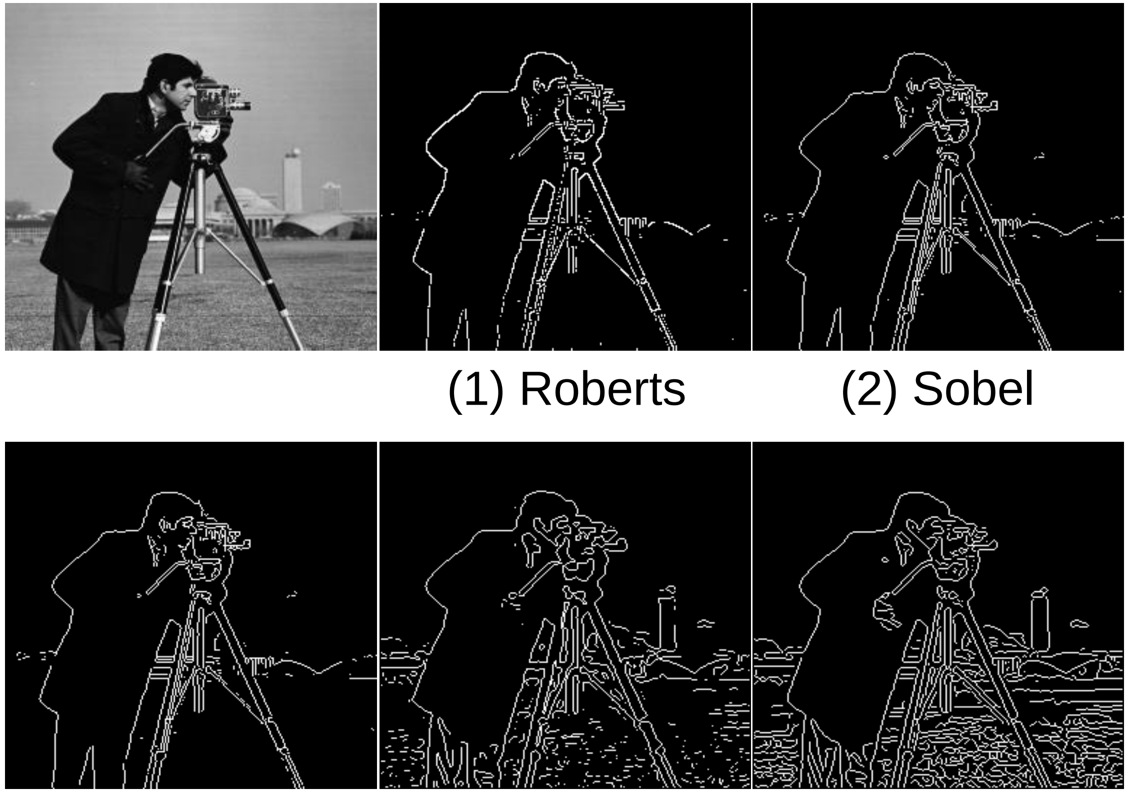
```
Ix = conv2(double(I),h); size(Ix)
Iy = conv2(double(I),h'); size(Iy)
grad = max(abs(Ix(1:246,2:301)), ...
           abs(Iy(2:247,1:300)));
figure, imshow(grad,[])

mask = im2bw(mat2gray(grad),0.15);
orient = atan2(Ix(1:246,2:301), ...
               Iy(2:247,1:300))*180/pi;
figure, imshow(orient, colormap('jet'))
figure,
imshow(orient.*mask,colormap('jet'))
```



orient.*mask

Edge detectors



(1) Roberts

(2) Sobel

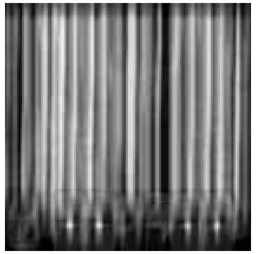
(3) Prewitt

(4) LoG

(5) Canny

Pattern Matching

- Find patterns = find specific subimages, templates or structures in an image → it needs of a definition of a similarity measure
- Correlation is a useful technique to start with this topic.



- Binary images: the result is the number of points that matches the mode
- Grey level images: we need a similarity measure:

$$s(x) = \sum_i [t(i) - f(x+i)]^2 = \sum_i t^2(i) - 2t(x) \circ f(x) + \sum_i f^2(x+i)$$

well suited if $\Sigma f^2(x+i)$ is nearly constant (strong restriction)

Pattern Matching

NCC (Normalized cross-correlation)

$$NCC = \frac{1}{IJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{(A_{(i,j)}-a)(B_{(i,j)}-b)}{\sigma_A\sigma_B}$$

$$a = \frac{1}{IJ}\sum_{j=1}^{J}\sum_{i=1}^{I}A_{(i,j)} \qquad b = \frac{1}{IJ}\sum_{j=1}^{J}\sum_{i=1}^{I}B_{(i,j)}$$

$$\sigma_A = \sqrt{\frac{1}{IJ}\sum_{j=1}^{J}\sum_{i=1}^{I}(A_{(i,j)}-a)^2} \qquad \sigma_B = \sqrt{\frac{1}{IJ}\sum_{j=1}^{J}\sum_{i=1}^{I}(B_{(i,j)}-b)^2}$$

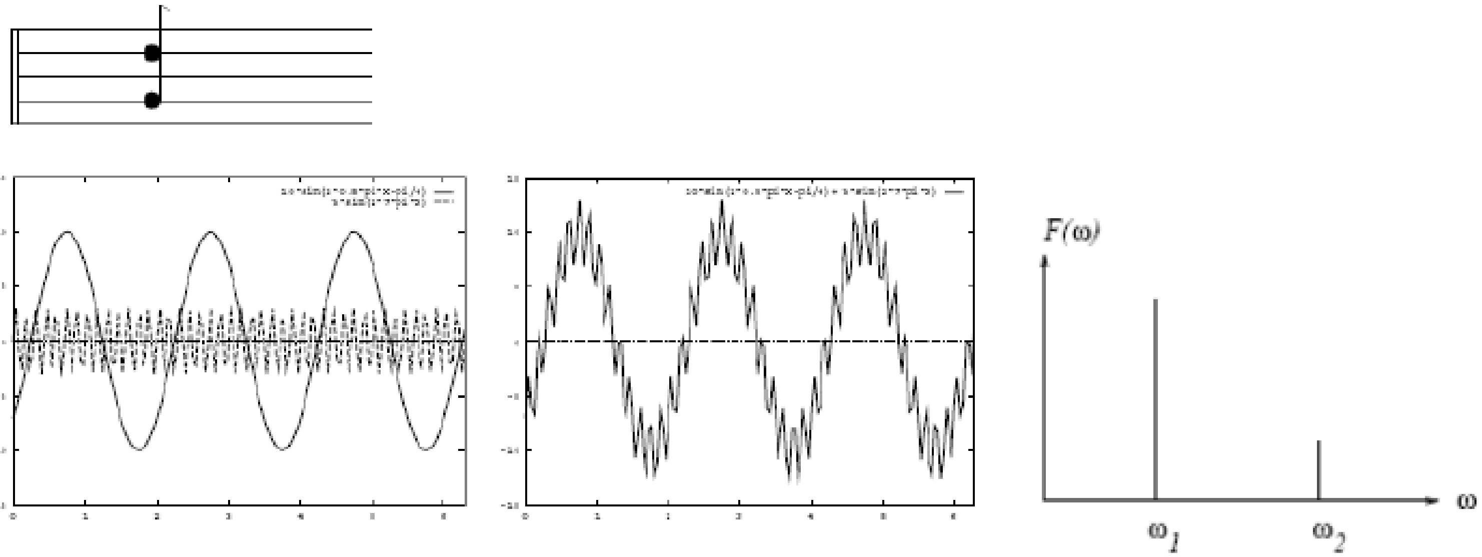
4

Fourier Transform

Motivation

Why Fourier Transform is used in image processing?

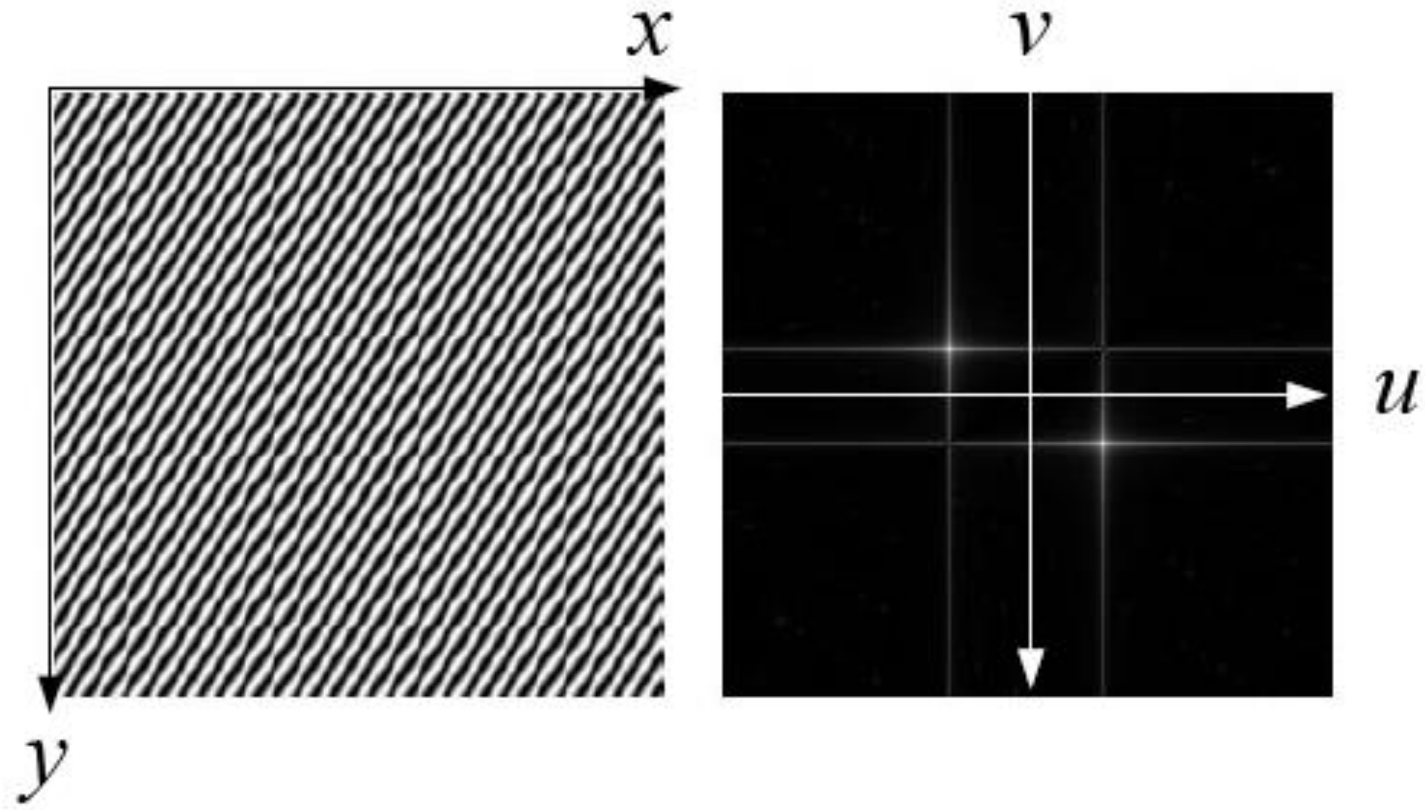
- To change image representation: from spatial domain (x,y) to frequency domain (u,v)



space / time

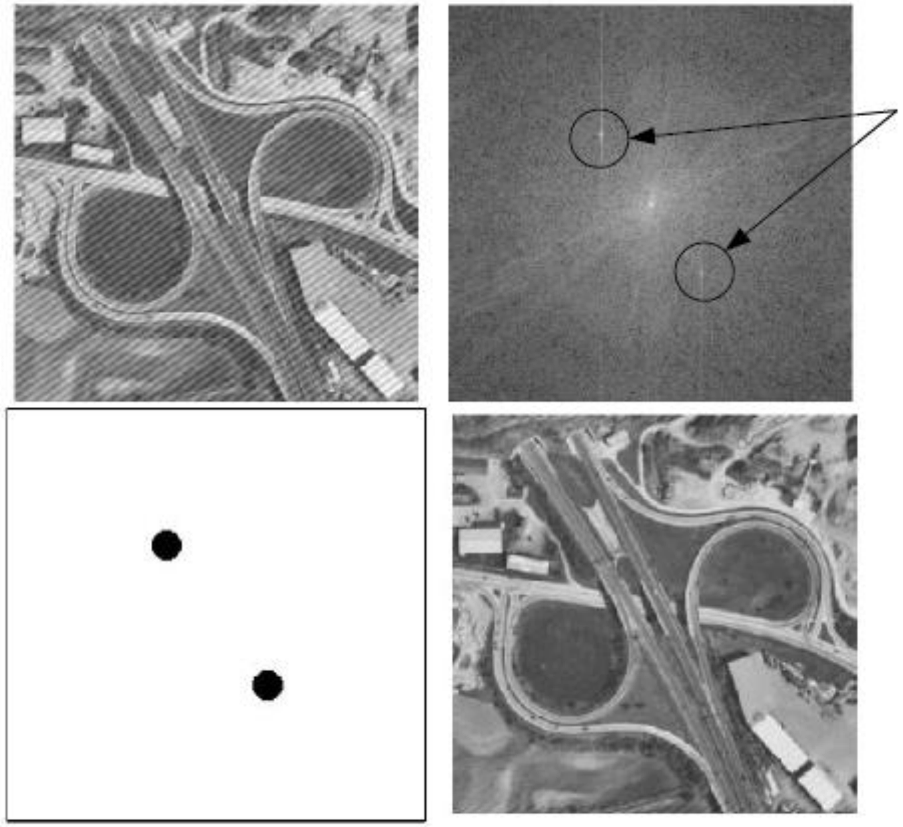
frequency

Motivation



Motivation

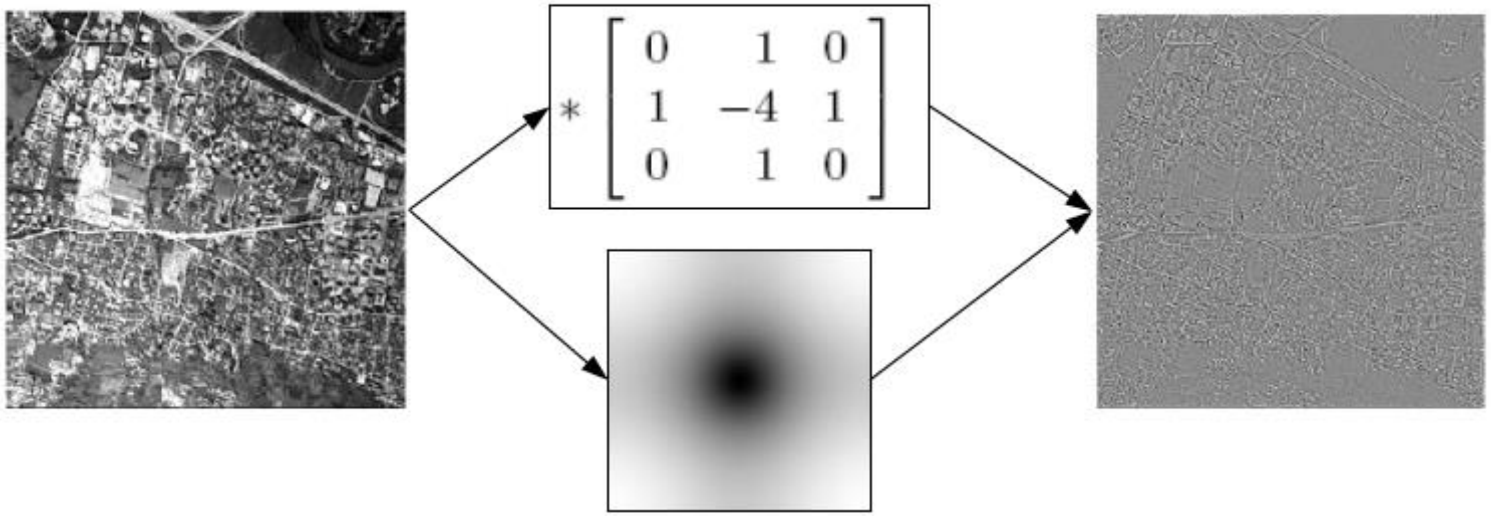
Filtering is different in the frequency and spatial domains



Motivation

- Fourier Transform allows us to obtain an alternative characterization of linear shift-invariant systems

$$f * h\,(x,y) = \mathcal{F}^{-1}\left\{ \mathcal{F}\{f(x,y)\}\boxed{\mathcal{F}\{h(x,y)\}} \right\}$$



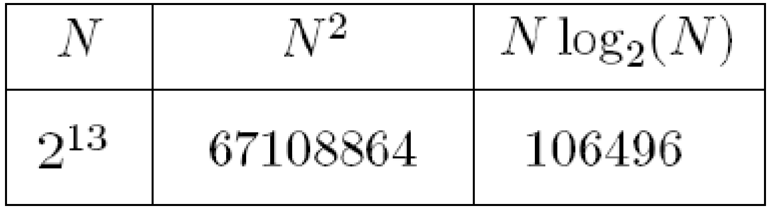
$$* \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Motivation

- Fast computation of convolution and correlations using FFT

$$f(x) \quad x = 0 \ldots N-1$$

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \mathrm{e}^{-j2\pi ux/N}, \quad u = 0, 1 \ldots N-1$$



N	N^2	$N \log_2(N)$
2^{13}	67108864	106496

$$\mathrm{e}^{+j\omega} = \cos(\omega) + \ j \sin(\omega)$$

Definition

- Continuous 2D signals

$$\mathcal{F}[f(x,y)] \quad = \quad F(\omega_1,\omega_2) = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} f(x,y)\mathrm{e}^{-j2\pi(\omega_1 x+\omega_2 y)}dxdy$$

for $x, y, \omega_1, \omega_2 \in \mathbb{R}$.

- Discrete space 2D signals

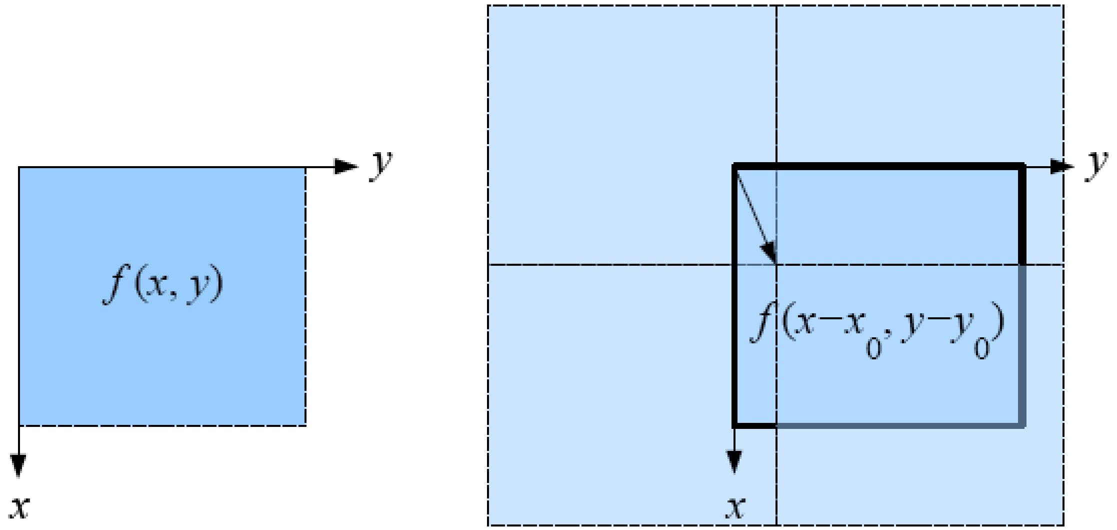
$$\mathcal{F}[f(x_1,x_2)] \quad = \quad F(u,v) = \sum_{x_1=-\infty}^{\infty}\sum_{x_2=-\infty}^{\infty} f(x_1,x_2)\mathrm{e}^{-j2\pi(ux_1+vx_2)}$$

for $-\infty < x_1, x_2 < \infty$ integers, but $u, v \in [-\pi, \pi]$

Properties

- Shift in space domain (x_0, y_0) → phase changes

$$f(x,y) \longleftrightarrow F(u,v)$$
$$f(x-x_0, y-y_0) \longleftrightarrow F(u,v)e^{-j2\pi(ux_0+vy_0)/N}$$



Properties

- **Correlation and convolution in space → product in frequency**

$$\begin{aligned} f * g\ (x,y) &\quad\longleftrightarrow\quad F(u,v)G(u,v) \\ f \circ g\ (x,y) &\quad\longleftrightarrow\quad F(u,v)G(u,v)^* \end{aligned}$$

- Correlation and convolution with module and phase:

$$\begin{aligned} f(x,y) &\quad\longleftrightarrow\quad F(u,v) = |F(u,v)|e^{j\phi_F(u,v)} \\ g(x,y) &\quad\longleftrightarrow\quad G(u,v) = |G(u,v)|e^{\phi_G(u,v)} \end{aligned}$$

$$\begin{aligned} f * g\ (x,y) &\quad\longleftrightarrow\quad F(u,v)G(u,v) = |F(u,v)||G(u,v)|e^{j(\phi_F+\phi_G)} \\ f \circ g\ (x,y) &\quad\longleftrightarrow\quad F(u,v)G(u,v)^* = |F(u,v)||G(u,v)|e^{j(\phi_F-\phi_G)} \end{aligned}$$

Properties

- **Rotation** $f(r, \theta + \theta_0) \longleftrightarrow F(\omega, \phi + \theta_0)$

- **Scale** $f(ax, by) \longleftrightarrow \frac{1}{|ab|} F(u/a, v/b)$

- **Periodicity**
$$\begin{aligned} F(u,v) &= F(u+N, v) = F(u, v+N) \\ &= F(u+N, v+N) \end{aligned}$$

- **Symmetry**
$$\begin{aligned} F(u,v) &= F^*(-u,-v) \\ |F(u,v)| &= |F(-u,-v)| \end{aligned}$$

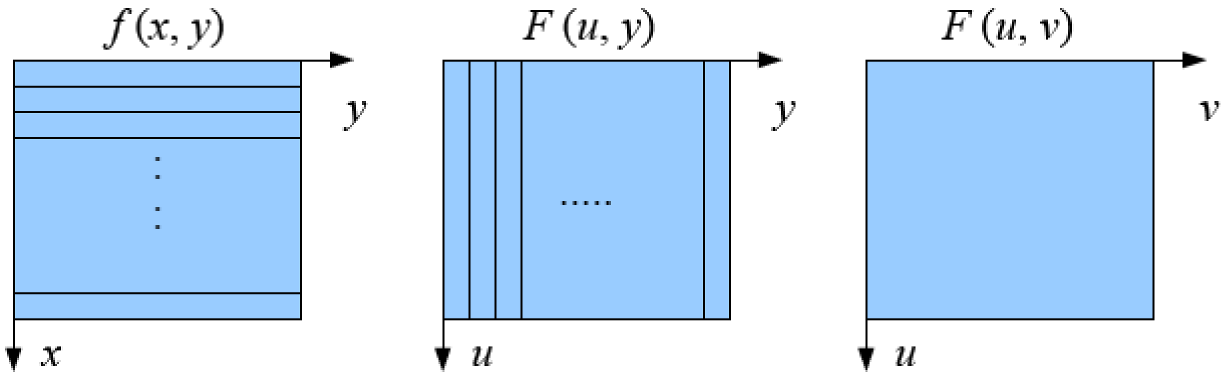
- **Dirac delta** $\delta(x - x_0, y - y_0) \quad \longleftrightarrow \quad e^{-j2\pi(ux_0 + vy_0)}$

Properties

- **Separability:** 2D FT computation starts with 1D FT

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\{\, e^{-j2\pi ux/N}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi vy/N} \,\}$$

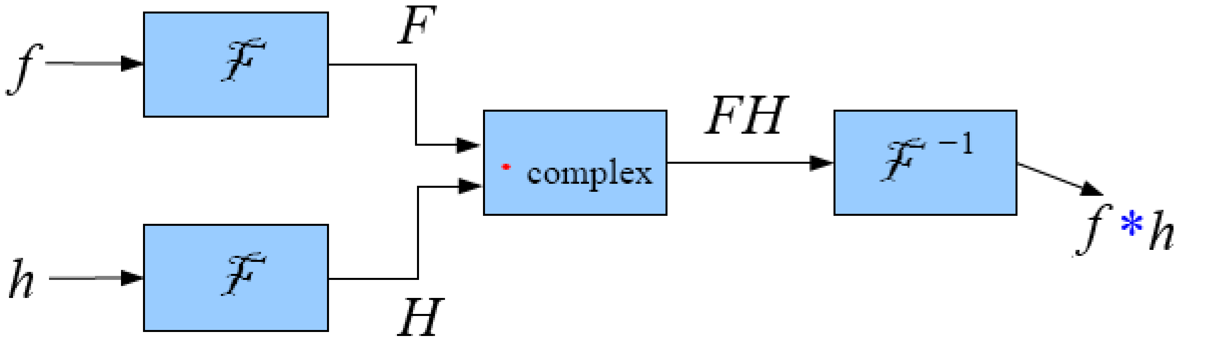
$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\{\, e^{+j2\pi ux/N}\sum_{v=0}^{N-1} f(x,y)e^{+j2\pi vy/N} \,\}$$



Linear Systems: convolution

- As there is only one FT for a given function and that for each FT there is only one source function

$$f * h\,(x, y) = \mathcal{F}^{-1}\{\ \mathcal{F}\{f\} \cdot \mathcal{F}\{h\}\ \}$$



H express how amplitude and phase of *F* change at each *u*,*v*

Linear Systems: correlation

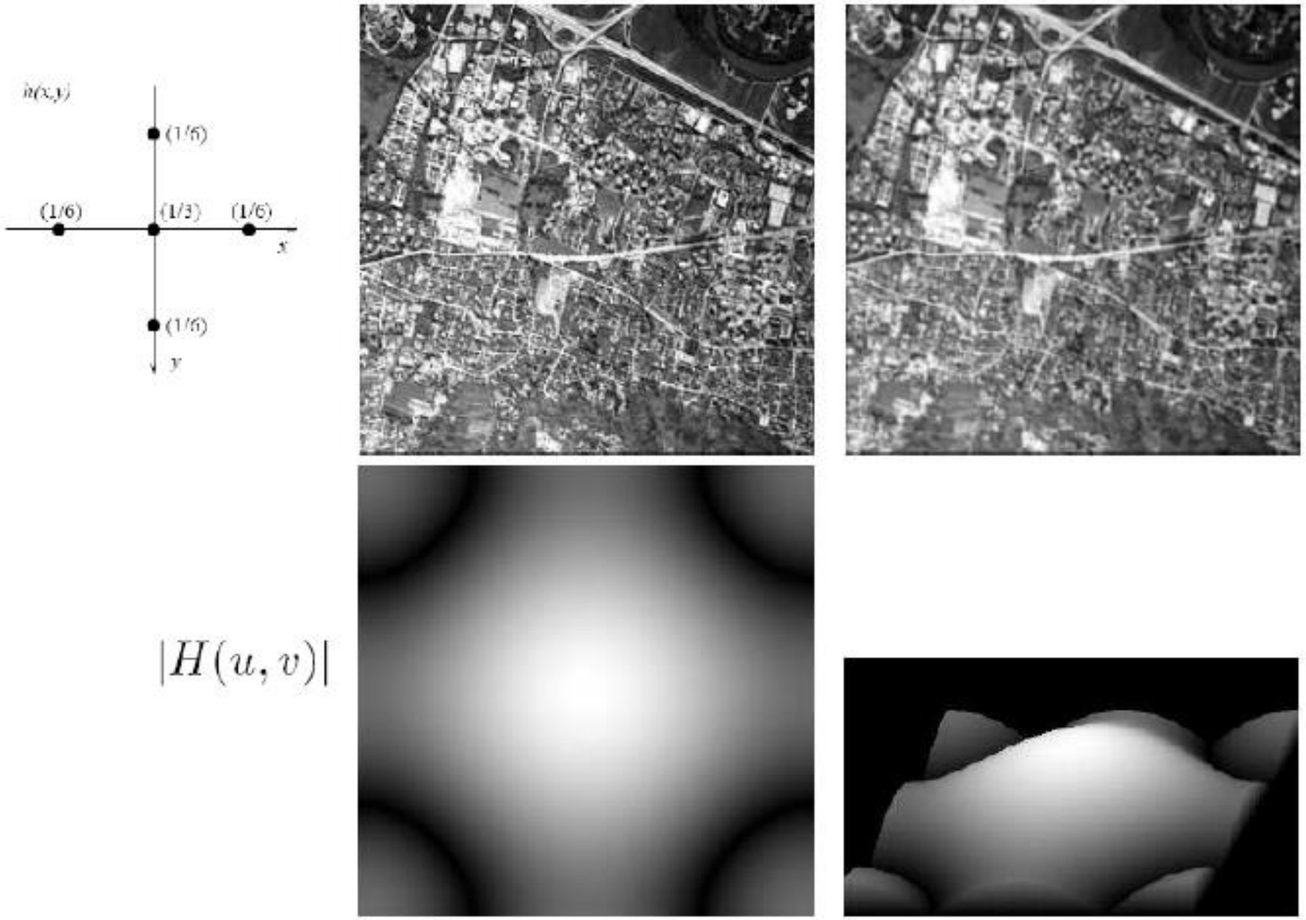
$$f \circ h\,(x, y) = \mathcal{F}^{-1}\{\ \mathcal{F}\{f\} \cdot \mathcal{F}\{h\}^{*}\ \}$$

$$^{*}\text{complex conjugate}: (\mathrm{a}+\mathrm{jb})^{*} = \mathrm{a}-\mathrm{jb}$$

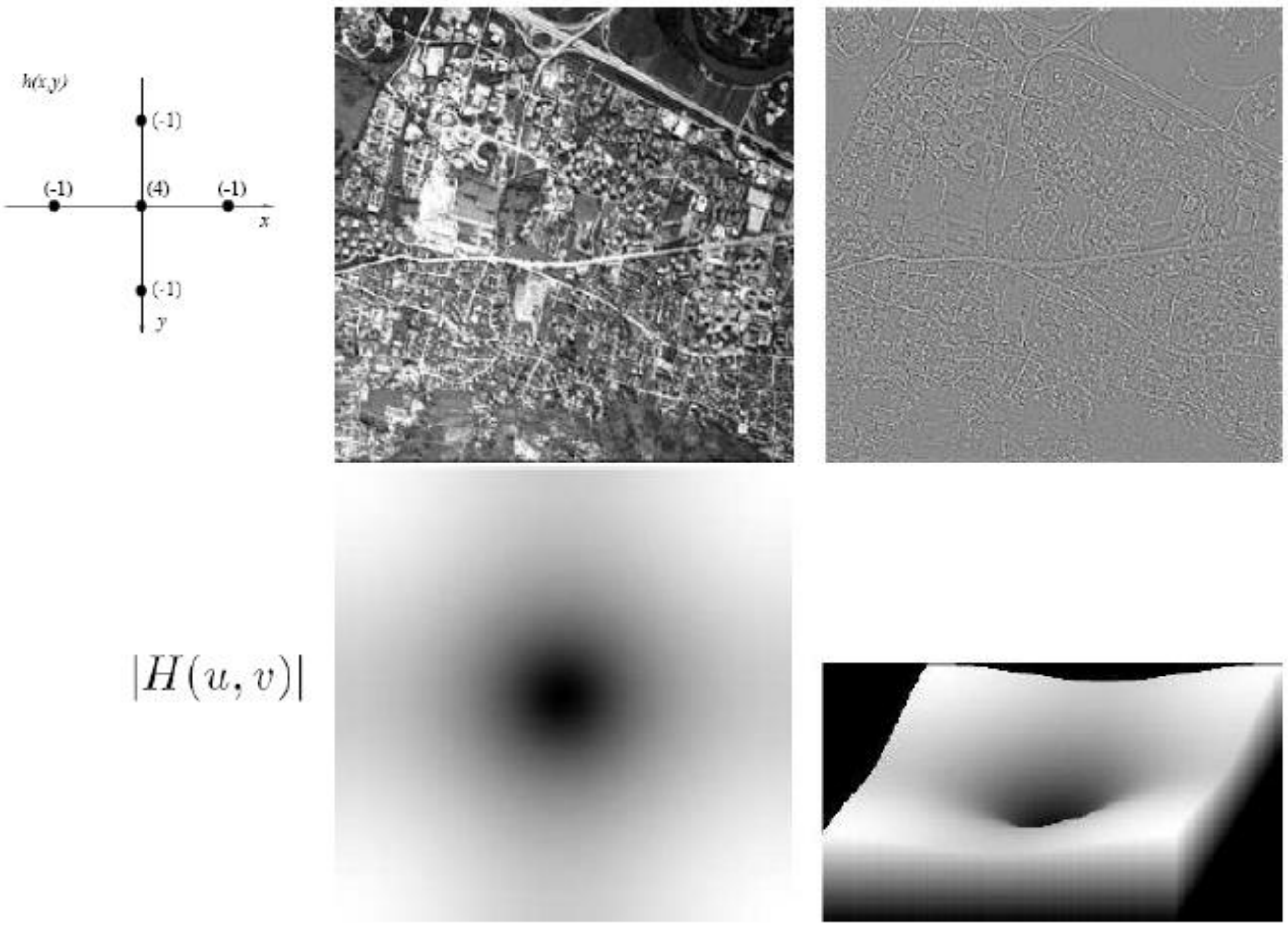
By definition:

$$f(x, y) \circ h\,(x, y) = f(x, y) * h\,(-x, -y)$$

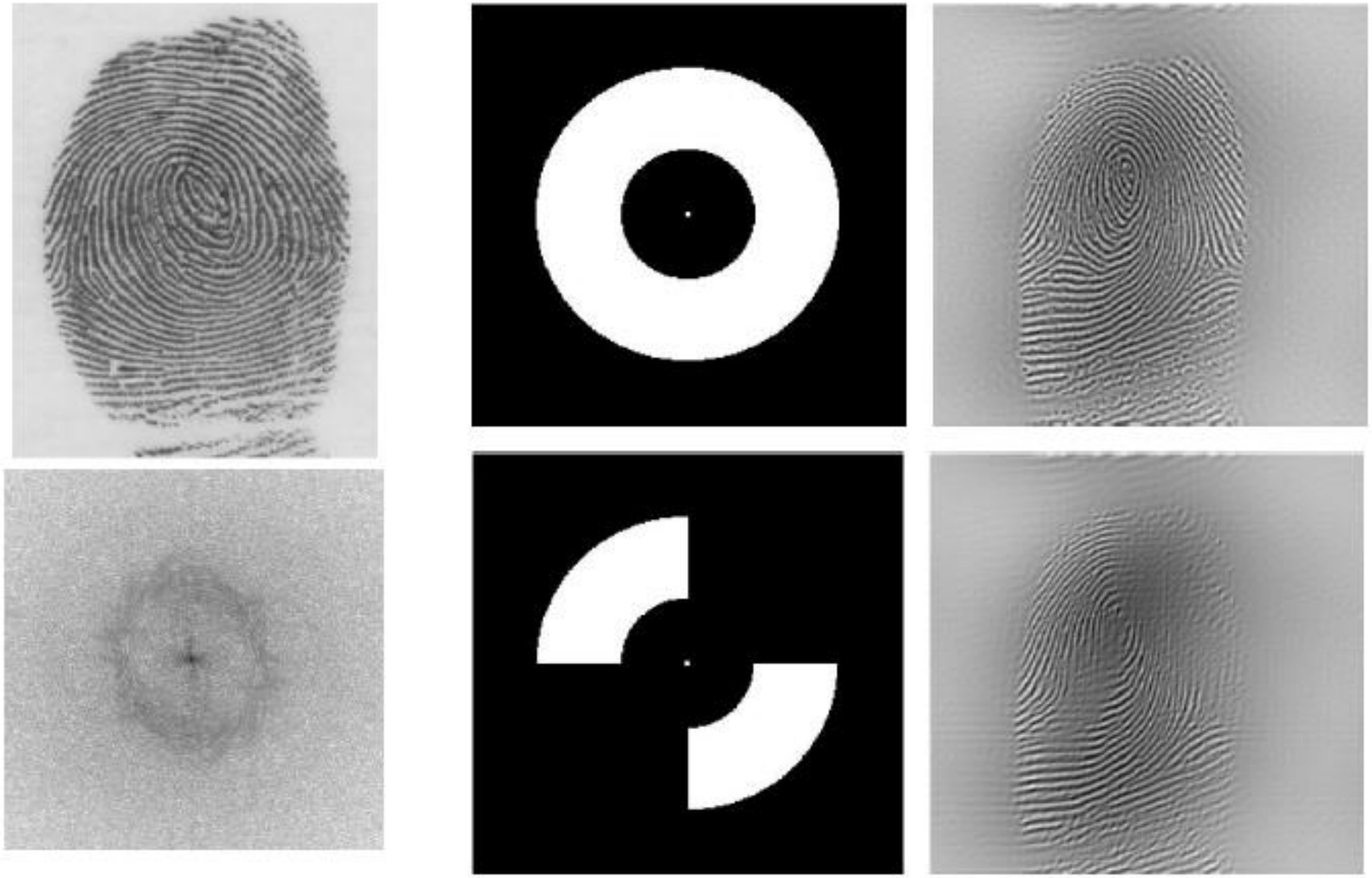
Frequency domain filtering



Frequency domain filtering



Frequency domain filtering



Fast correlation & convolution

- Space domain

$$f * g\ (k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j)h(k-i,l-j) \quad k,l = 0 \ldots M-1$$

M^2N^2 products

- Frequency domain

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

$12M^2 \log M + 4M^2$ products

Fundamentals of Computer Vision

Unit 4: Linear Filtering

Jorge Bernal (Jorge.Bernal@uab.cat)