CMP362/CMPN446: Image Processing and Computer Vision

Lecture 02: Basic Concepts

Mayada Hadhoud
Computer Engineering Department
Cairo University

Agenda

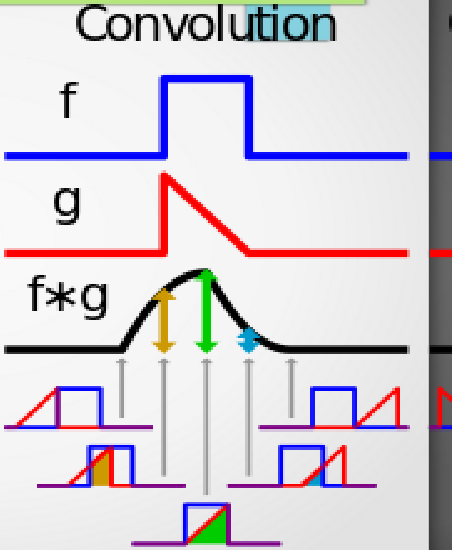
- **Convolution**
- **Fourier Transform**

Basic Concepts - Convolution

- Convolution

  is an integral that expresses the amount of overlap of one function as it is shifted over another function.

  convolution is a filtering operation



Basic Concepts - Convolution

- Convolution of two functions is defined as

$$f(x)*h(x)=\int_{-\infty}^{\infty} f(\alpha)h(x-\alpha)d\alpha$$

  - **In the discrete case is:**

$$f[n]*h[n]=\sum_{m=-\infty}^{\infty} f[m]h[n-m]$$

Basic Concepts - Convolution

- **In the 2D discrete case Convolution** is defined as:

$$f[n_1,n_2]*h[n_1,n_2]=\sum_{m_1=-\infty}^{\infty}\sum_{m_2=-\infty}^{\infty}f[m_1,m_2]h[n_1-m_1,n_2-m_2]$$

$h[n_1,n_2]$ is called a **linear filter**

Basic Concepts - Convolution

- **2D discrete case Convolution** (cont.):
  - Step1:

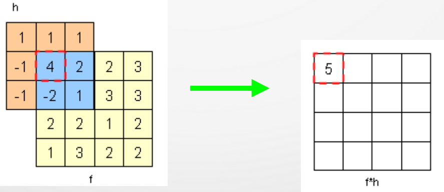
1	1	1
-1	2	1
-1	-1	1

filter

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

image

f



f * h

Basic Concepts - Convolution

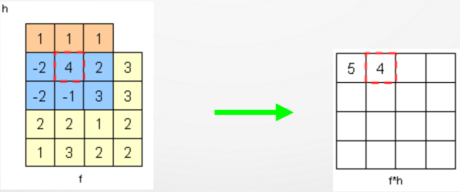
- **2D discrete case Convolution** (cont.):
  - Step2:

1	1	1
-1	2	1
-1	-1	1

filter

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

image



f * h

f

Basic Concepts - Convolution

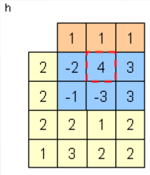
- **2D discrete case Convolution** (cont.):
  - Step3:

1	1	1
-1	2	1
-1	-1	1

filter

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

image



f

5	4	4	

f * h

Basic Concepts - Convolution

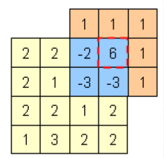
- **2D discrete case Convolution** (cont.):
  - Step4:

1	1	1
-1	2	1
-1	-1	1

filter

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

image



f

5	4	4	-2

f * h

Basic Concepts - Convolution

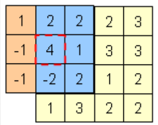
- **2D discrete case Convolution** (cont.):
  - Step5:

1	1	1
-1	2	1
-1	-1	1

filter

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

image



5	4	4	-2
9			

etc. . . . .

f * h

Basic Concepts - Convolution

Convolution: Example

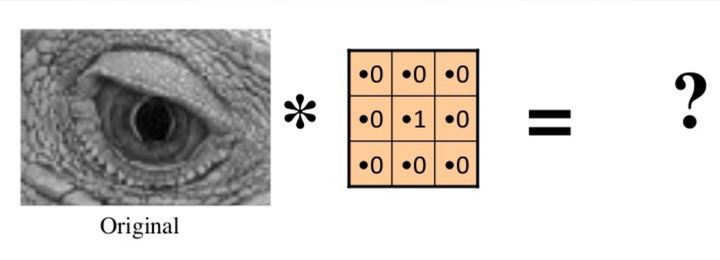
$$* \ \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

Basic Concepts - Convolution

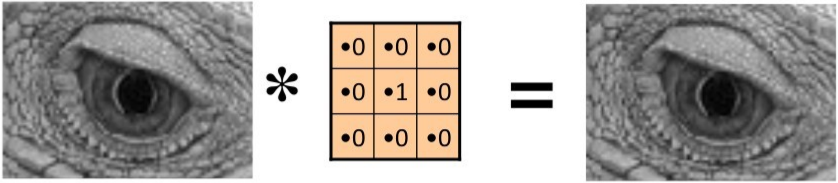
Convolution: Example

$$* \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

Convolution in 2D - examples



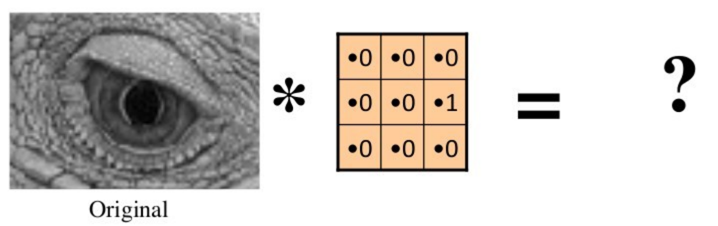
Convolution in 2D - examples



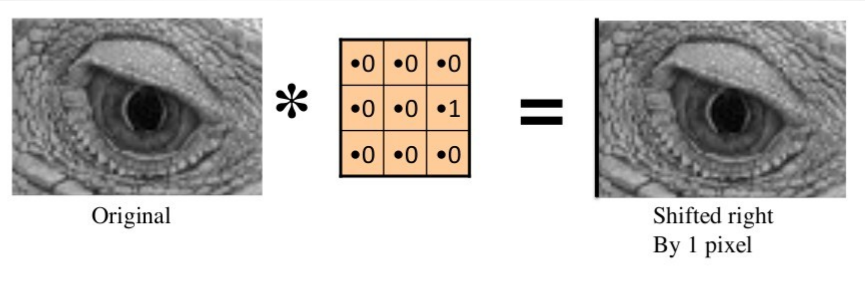
Original

Filtered
(no change)

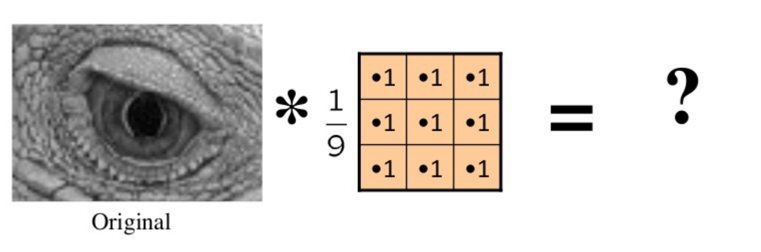
Convolution in 2D - examples



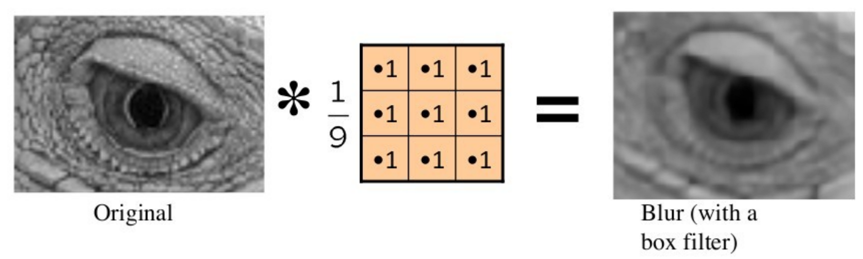
Convolution in 2D - examples



Convolution in 2D - examples



Convolution in 2D - examples



Fourier Transform

- A signal can be represented as **a weighted sum of sinusoids.**

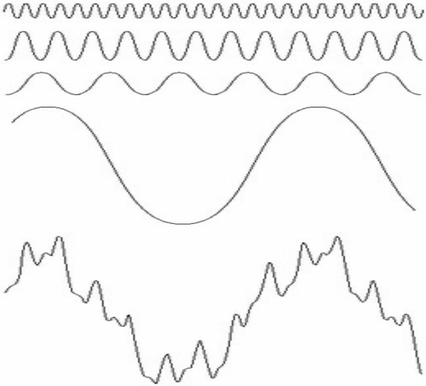
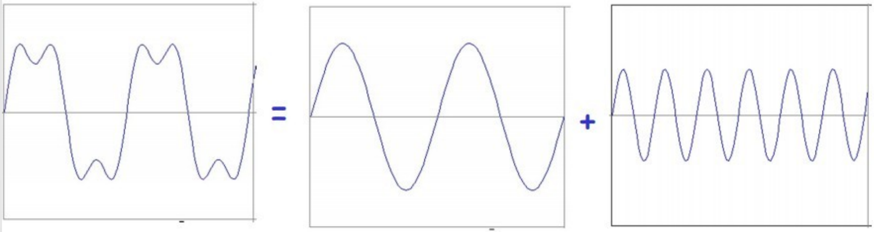
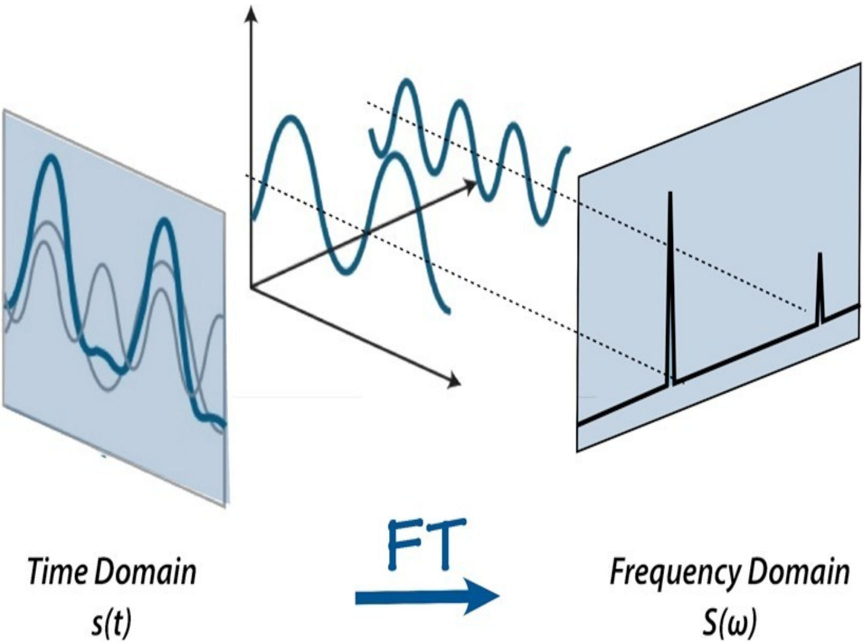


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Example



$$f(x) = \sin x + \frac{1}{3} \sin 3x + \ldots$$



Time Domain
s(t)
FT
Frequency Domain
S(ω)

Fourier Transform

- The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components.
- **Fourier Transform convert the image from Spatial Domain to Frequency Domain**
- In the Fourier domain image, **each point represents a particular frequency contained in the spatial domain image**.

Discrete Fourier Transform

- 2D Discrete Fourier Transform (DFT)

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j)\, e^{-\iota 2\pi\left(\frac{ki}{N} + \frac{lj}{N}\right)}$$

- **Inverse DFT**

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l)\, e^{\iota 2\pi\left(\frac{ka}{N} + \frac{lb}{N}\right)}$$

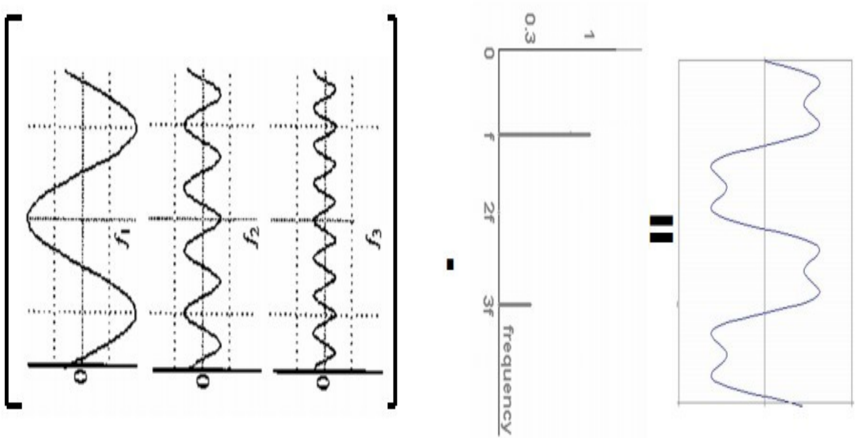
1- The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image

2- f(a,b) is the image in the spatial domain and the exponential term is the basis function corresponding to each point F(k,l) in the Fourier space. The equation can be interpreted as: the value of each pointF(k,l) is obtained by multiplying the spatial image with the corresponding base function and summing the result.

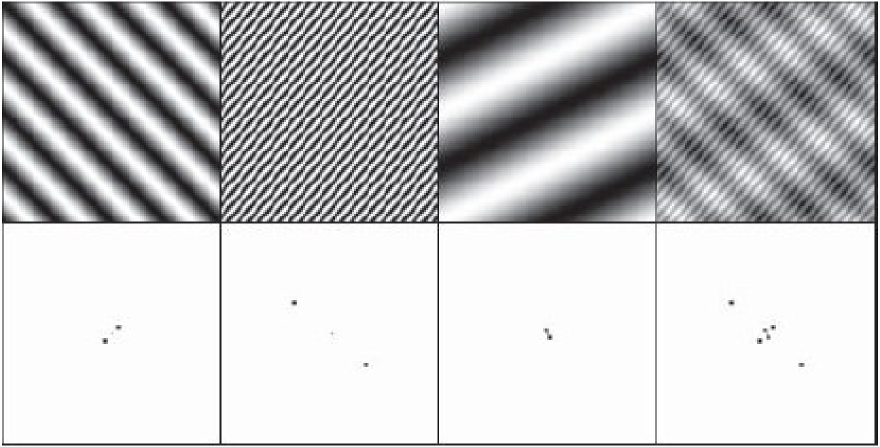
3- The basis functions are sine and cosine waves with increasing frequencies, i.e. F(0,0) represents the DC-component of the image which corresponds to the average brightness and F(N-1,N-1) represents the highest frequency.

4- The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with
the real and imaginary part or with magnitude and phase. In image processing, often only the magnitude of the Fourier Transform is displayed, as it contains most of the information of the geometric structure of the spatial domain image. However, if we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.

Inverse Fourier Transform

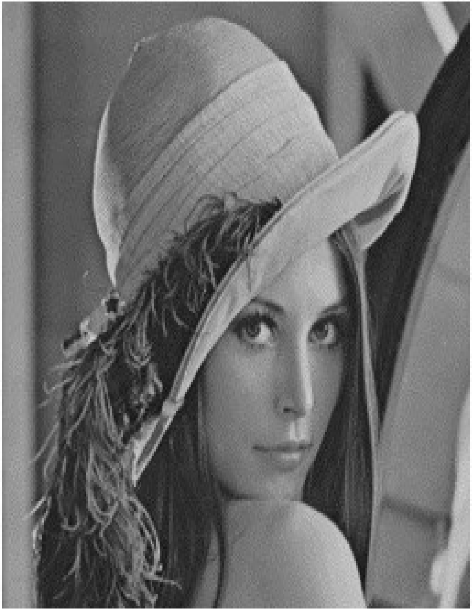
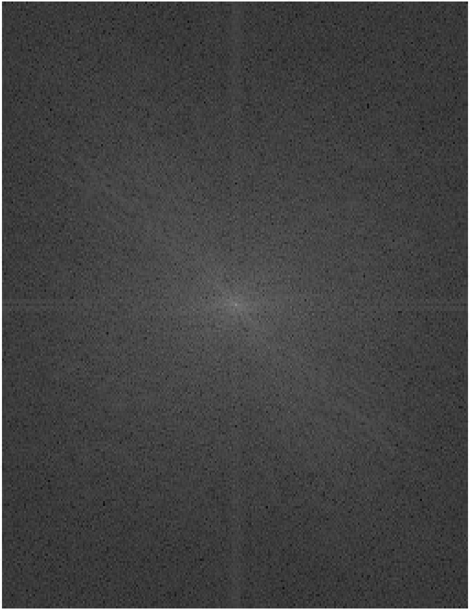


2D DFT – Power Spectra



Four sinusoidal patterns, their frequency transforms, and their sum. The frequency increases with radius ρ, and the orientation depends on the angle θ. Origin is in the center of the image.

- Taking the Fourier transform of an image converts the **straightforward information** in the **spatial domain** into a **scrambled form** in the **frequency domain**. In short, don't expect the Fourier transform to help you understand the information encoded in images.

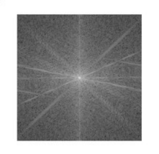



Why we use DFT?

- In images **increasing frequency** is associated with **more abrupt transitions in brightness or color**.
- **Noise** is usually embedded in the **high end of the spectrum**, so low-pass filtering can be used **for noise reduction**.
- For **image compression**, because the high frequency components are usually just noise.
- **Convolution**, a fundamental image processing operation, can be done much faster by using the Fourier transform.

Fourier Transform: DFT- Filtering

F(u,v)

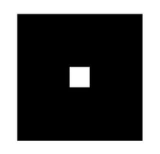


DFT

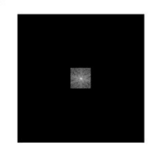
f(x,y)

H(u,v)

LPF



H(u,v)F(u,v)



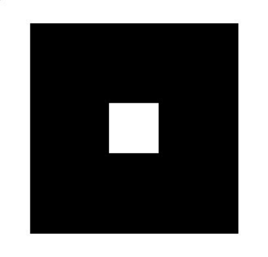
Inverse

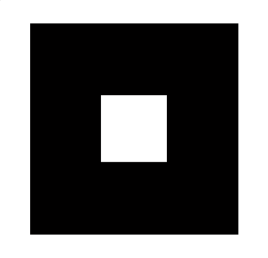
DFT

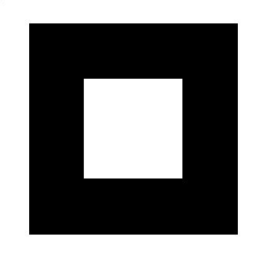
Frequency domain

Spatial domain

Fourier Transform: DFT- Filtering





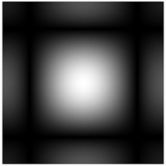


61x61 81x81 121x121

Fourier Transform: DFT- Filtering

h

$$* \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad =$$

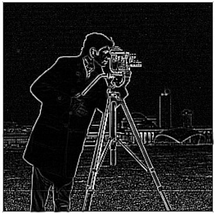


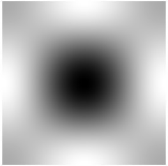
DFT(h)

Fourier Transform: DFT- Filtering

h

$$* \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$





DFT(h)

eliminating high frequencies blurs the image.
Eliminating low frequencies gives you edges.
And enhancing high frequencies while keeping the low frequencies sharpens the image.