Linear and Non-linear Filtering

Arthur Leclaire

TELECOM Paris
IP PARIS

Athens Week 2025

Goals of this course

- Understand the effect of a convolution (linear filtering)
- Analyze convolution operators in the spectral domain
- Discover basic non-linear filters (mathematical morphology)
- Perform simple image analysis tasks with these filters

Outline

Linear Filtering

Linear Filtering in Fourier Domain

Mathematical Morphology

Convolution

Let $u : \Omega \to \mathbf{R}$ be a graylevel image defined on $\Omega = [0 : M-1] \times [0 : N-1]$.

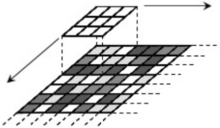
Let $k : \omega \to \mathbf{R}$ be a **kernel** defined on a small domain $\omega \subset \mathbb{Z}^2$.

Often, k will be defined on a small square $\omega = [-s, s]^2$.

Definition

The convolution $k * u$ of the image u with kernel k is defined by

$$k * u(x, y) = \sum_{(x', y') \in \omega} k(x', y') u(x - x', y - y')$$



This operation is also called **linear filtering with filter** k.

Convolution

Let $u : \Omega \to \mathbf{R}$ be a graylevel image defined on $\Omega = [0 : M-1] \times [0 : N-1]$.

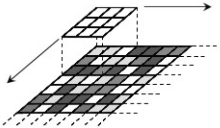
Let $k : \omega \to \mathbf{R}$ be a **kernel** defined on a small domain $\omega \subset \mathbb{Z}^2$.

Often, k will be defined on a small square $\omega = [-s, s]^2$.

Definition

The convolution $k * u$ of the image u with kernel k is defined by

$$k * u(x, y) = \sum_{(x', y') \in \omega} k(x', y') u(x - x', y - y')$$



This operation is also called **linear filtering with filter** k.

NB: If

- ω is a small neighborhood of $(0, 0)$
- $k : \omega \to \mathbf{R}_+$ is such that $\sum_{(x,y)\in\omega} k(x, y) = 1$

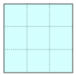
then $k * u(x, y)$ is a kind of average of values of u around pixel (x, y).

Convolution Example

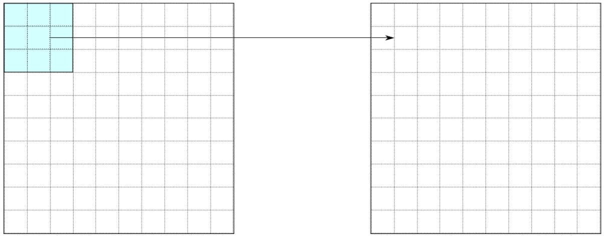
$$k * u(x,y) = \sum_{(-x',-y')\in\omega} k(-x',-y')u(x+x',y+y') = \sum_{(-x',-y')\in\omega} \tilde{k}(x',y')u(x+x',y+y')$$

where $\tilde{k}$ is defined by $\tilde{k}(x,y) = k(-x,-y)$.

Example: with a kernel k defined on a 3×3 square $\omega = [-1,1]^2$:



Reflected Kernel $\tilde{k}$

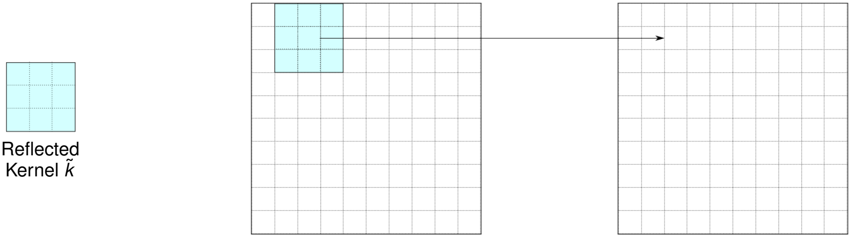


Convolution Example

$$k * u(x,y) = \sum_{(-x',-y')\in\omega} k(-x',-y')u(x+x',y+y') = \sum_{(-x',-y')\in\omega} \tilde{k}(x',y')u(x+x',y+y')$$

where $\tilde{k}$ is defined by $\tilde{k}(x,y) = k(-x,-y)$.

Example: with a kernel k defined on a 3×3 square $\omega = [-1,1]^2$:



Exercise

Compute the convolution $k * u$ with the following kernel and image

1	1	-1

k

-1	1	1

$\tilde{k}$

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

u

Exercise

Compute the convolution $k * u$ with the following kernel and image

1	1	-1

k

-1	1	1

$\tilde{k}$

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

u

$k * u$

Exercise

Compute the convolution $k * u$ with the following kernel and image

1	1	-1

k

-1	1	1

$\tilde{k}$

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

u

	3	4	
	1	1	
	2	4	

$k * u$

Exercise

Compute the convolution $k * u$ with the following kernel and image

1	1	-1

k

-1	1	1

$\tilde{k}$

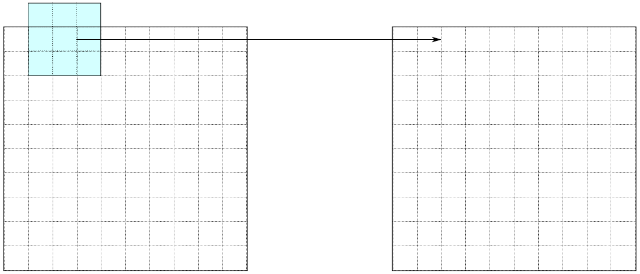
0	1	2	3
3	2	2	1
0	0	2	2

u

?	3	4	?
?	1	1	?
?	2	4	?

$k * u$

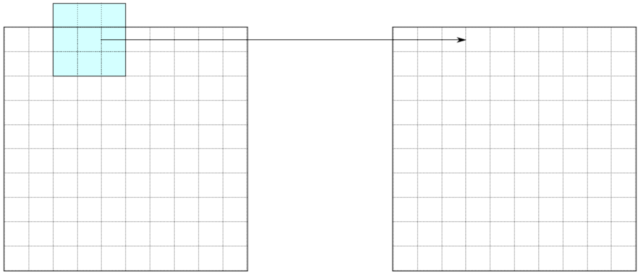
Border condition



When (x, y) is at the border of Ω, we should adopt a **border condition** to compute $u(x - x', y - y')$, for example:

- Zero-padding (extend image domain with 0 values)
- Periodic extension (recopy image in both directions)

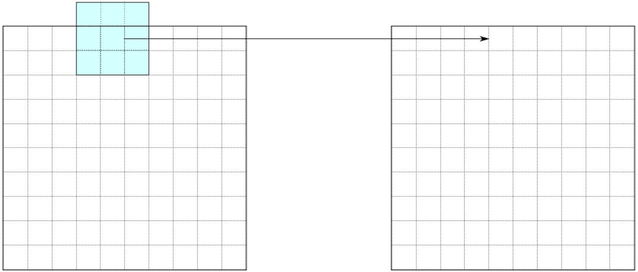
Border condition



When (x, y) is at the border of Ω, we should adopt a **border condition** to compute $u(x - x', y - y')$, for example:

- Zero-padding (extend image domain with 0 values)
- Periodic extension (recopy image in both directions)

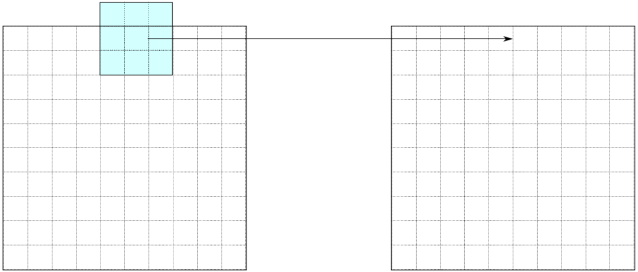
Border condition



When (x, y) is at the border of Ω, we should adopt a **border condition** to compute $u(x - x', y - y')$, for example:

- Zero-padding (extend image domain with 0 values)
- Periodic extension (recopy image in both directions)

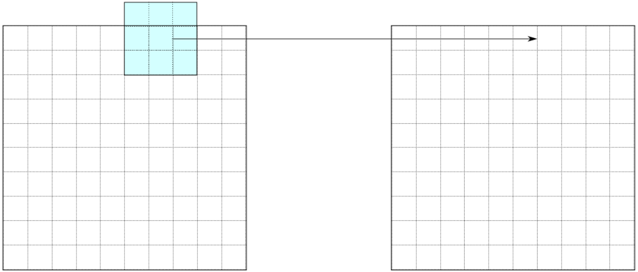
Border condition



When (x, y) is at the border of Ω, we should adopt a **border condition** to compute $u(x - x', y - y')$, for example:

- Zero-padding (extend image domain with 0 values)
- Periodic extension (recopy image in both directions)

Border condition



When (x, y) is at the border of Ω, we should adopt a **border condition** to compute $u(x - x', y - y')$, for example:

- Zero-padding (extend image domain with 0 values)
- Periodic extension (recopy image in both directions)

Back to the Exercice

Compute the convolution $k * u$ by using **zero-padding** as boundary condition:

1/3	1/3	1/3

k

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

u

Back to the Exercice

Compute the convolution $k * u$ by using **zero-padding** as boundary condition:

1/3	1/3	1/3

k

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

u

	3	4	
	1	1	
	2	4	

$k * u$

Back to the Exercice

Compute the convolution $k * u$ by using **zero-padding** as boundary condition:

1/3	1/3	1/3

k

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

u

1	3	4	1
5	1	1	-1
0	2	4	0

$k * u$

Discrete Derivatives

The discrete derivatives of u are images $\partial_1 u, \partial_2 u$ defined by

$$\begin{cases} \partial_1 u(x,y) = u(x+1,y) - u(x,y) \\ \partial_2 u(x,y) = u(x,y+1) - u(x,y) \end{cases} .$$

We also define the gradient of u which is a "vector-field image" $\nabla u : \Omega \to \mathbf{R}^2$ with

$$\nabla u(x,y) = (\partial_1 u(x,y), \partial_2 u(x,y)).$$

One can remark that partial derivatives are given by discrete convolutions:

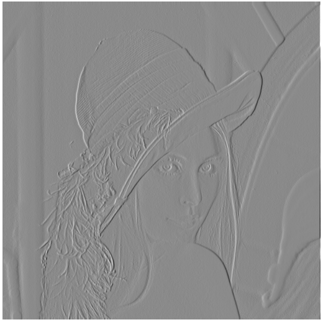
For example, $\partial_1 u = k_1 * u$ with k_1 which has only two non-zero values:

$$\begin{cases} k_1(0,0) = -1 \ , \\ k_1(-1,0) = 1 \ . \end{cases}$$

Example of Discrete Derivatives

u

$\partial_1 u = k_1 * u$

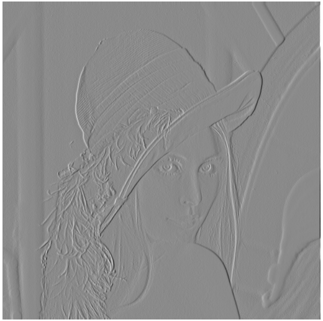


$\partial_2 u = k_2 * u$

Example of Discrete Derivatives

u

$\partial_1 u = k_1 * u$



$\partial_2 u = k_2 * u$

NB: With Python convention for indexing,

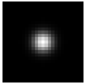
- $\partial_1 u$ responds more to horizontal edges
- $\partial_2 u$ responds more to vertical edges

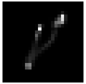
Blurring Operators

- A spatially uniform blur can be seen as a convolution

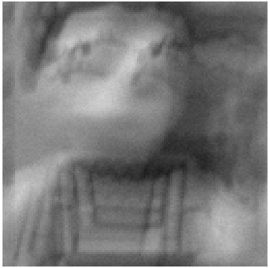
$$Au = k * u$$

- Depending on k we may have different kinds of blur.


Isotropic Blur


Motion Blur


Original u


Blurred image $k * u$

Outline

Discrete Fourier Transform (DFT)

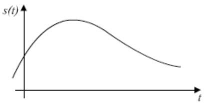
Definition

The discrete Fourier transform (DFT) of the image $u : \Omega \to \mathbb{C}$ is defined by

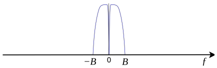
$$\forall (\xi, \zeta) \in \mathbb{Z}^2, \quad \hat{u}(\xi, \zeta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} u(x, y) \exp\left(-2i\pi\left(\frac{\xi x}{M} + \frac{\zeta y}{N}\right)\right) .$$

Such an image is implicitly extended by (M, N)-periodicity as $u : \mathbb{Z}^2 \to \mathbb{C}$.

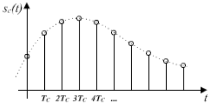
One can see that $\hat{u}$ is also (M, N)-periodic: $\hat{u}(\xi + kM, \zeta + \ell N) = \hat{u}(\xi, \zeta), \quad \forall k, \ell \in \mathbb{Z}$.



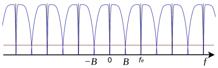
Continuous s



$\hat{s}$



Discrete s



$\hat{s}$

DFT Properties

- The DFT is linear.
- Consider the translated image $v(x, y) = u(x - a, y - b)$. Then

$$\hat{v}(\xi, \zeta) = \exp\left(-2i\pi\left(\frac{\xi a}{M} + \frac{\zeta b}{N}\right)\right)\hat{u}(\xi, \zeta).$$

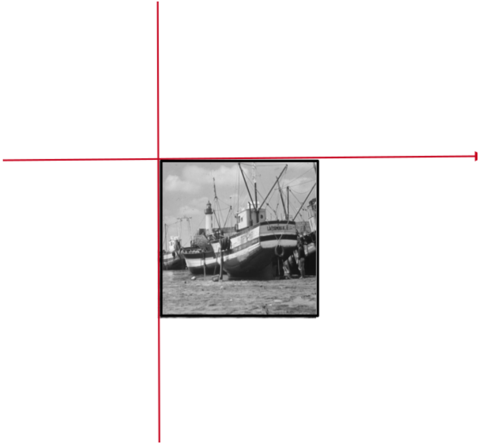
- If u is real-valued, then $\quad \forall(\xi, \zeta) \in \mathbb{Z}^2, \quad \hat{u}(-\xi, -\zeta) = \overline{\hat{u}(\xi, \zeta)}$.

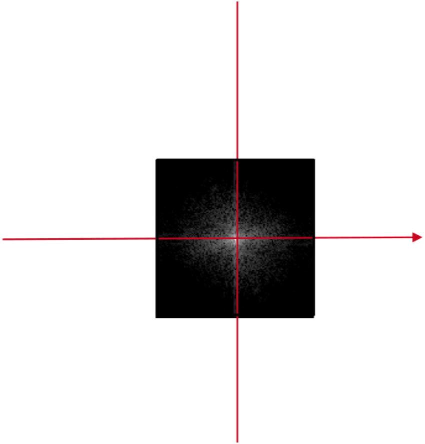
  In this case, we have that $|\hat{u}|$ is an even function.

NB: For this reason, DFT are usually displayed on the centered $M \times N$ domain, approximately:

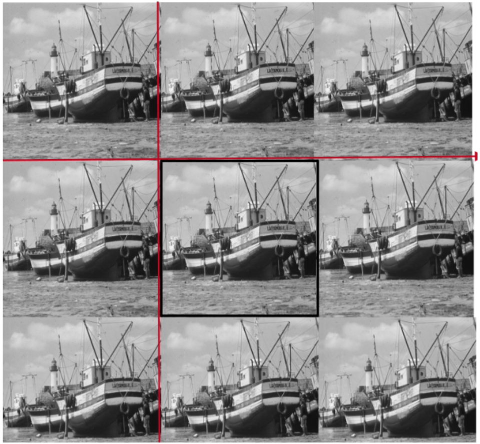
$$\Theta = \left[-\frac{M}{2} : \frac{M}{2}\right] \times \left[-\frac{N}{2} : \frac{N}{2}\right].$$

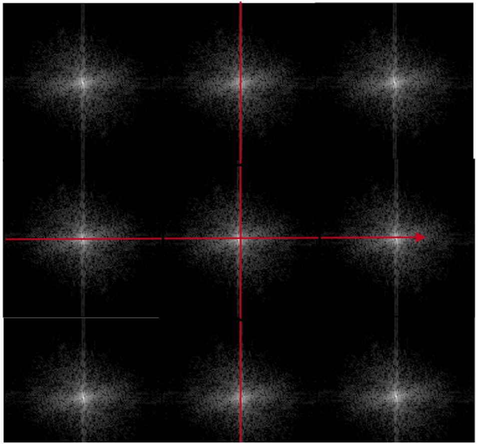
DFT and Periodicity





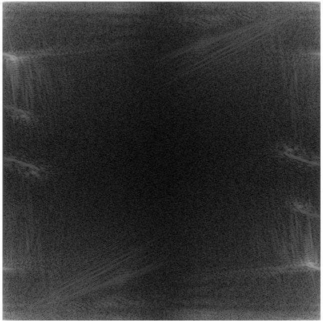
DFT and Periodicity



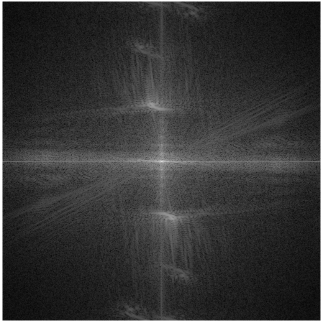


Spectrum of Natural Images

`u`

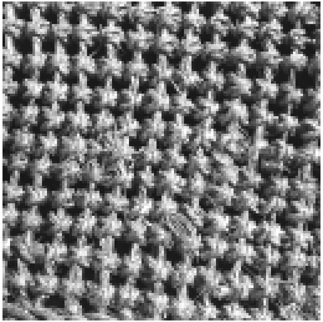


`fft2(u)`

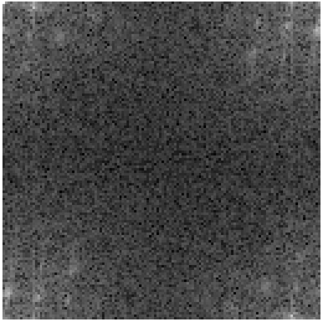


`fftshift(fft2(u))`

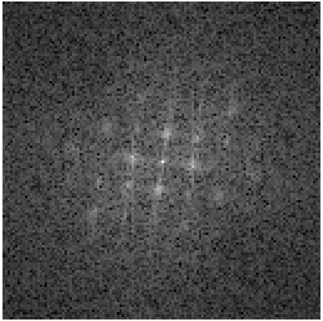
Spectrum of Natural Images



`u`



`fft2(u)`



`fftshift(fft2(u))`

Inverse Discrete Fourier Transform

Definition

The inverse discrete Fourier transform (iDFT) of $v : \Omega \to \mathbb{C}$ is defined by

$$\forall (x, y) \in \mathbb{Z}^2, \quad \check{v}(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\zeta=0}^{N-1} v(\xi, \zeta) \exp\left(2i\pi\left(\frac{\xi x}{M} + \frac{\zeta y}{N}\right)\right) .$$

Theorem

For any $u : \Omega \to \mathbb{C}$ (extended by (M, N)-periodicity), we have $\check{\hat{u}} = u$, that is,

$$\forall (x, y) \in \mathbb{Z}^2, \quad u(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\zeta=0}^{N-1} \hat{u}(\xi, \zeta) \exp\left(2i\pi\left(\frac{\xi x}{M} + \frac{\zeta y}{N}\right)\right) .$$

IMPORTANT: There exists an algorithm, called the **Fast Fourier Transform (FFT)** that allows to compute the DFT (or iDFT) of u with $\mathcal{O}(MN \log(MN))$ operations.

Convolution in Fourier Domain

Theorem

Consider $u, v : \Omega \to \mathbb{C}$ and the convolution $u * v$ computed with periodic boundary conditions. Then $\widehat{u * v} = \hat{u}\hat{v}$, that is,

$$\forall(\xi, \zeta) \in \mathbb{Z}^2, \quad \widehat{u * v}(\xi, \zeta) = \hat{u}(\xi, \zeta)\hat{v}(\xi, \zeta).$$

In other words, the **DFT transforms a convolution into a product**.

Important Consequence: The convolution $u * v$ can be computed with $\mathcal{O}(MN \log MN)$ operations:

$$u * v = \mathrm{DFT}^{-1}\Big(\mathrm{DFT}(u) \cdot \mathrm{DFT}(v)\Big).$$

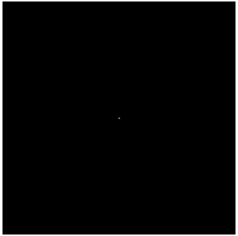
Corollaire

For $u, v : \Omega \to \mathbb{C}$, we also have $\widehat{uv} = \frac{1}{MN}\hat{u} * \hat{v}$.

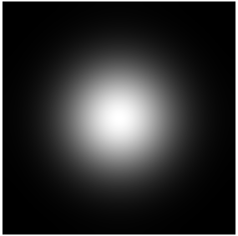
Gaussian Blur

u

$g_1 * u$

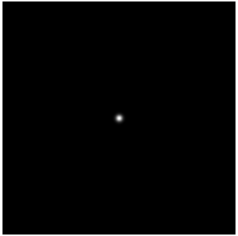


g_1

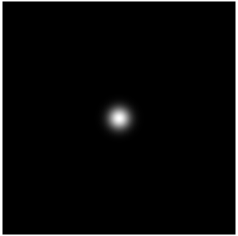


$|\hat{g}_1|$

$g_5 * u$

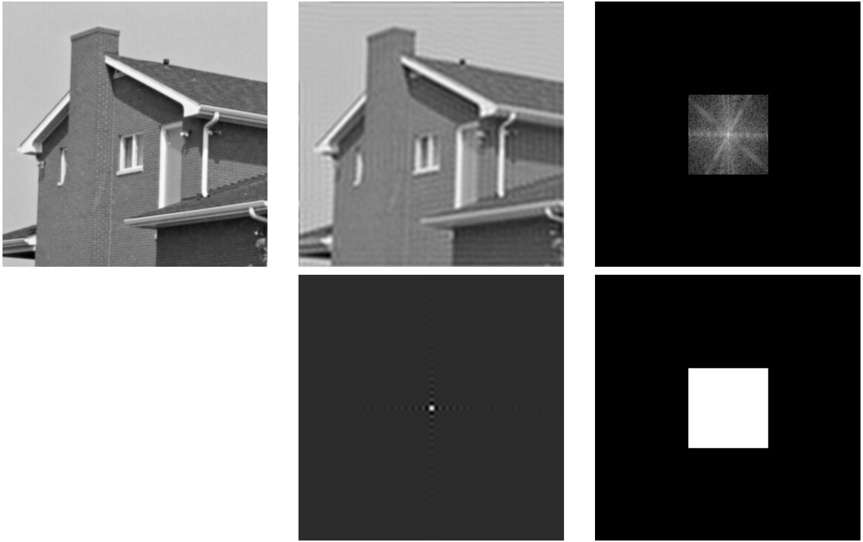


g_5

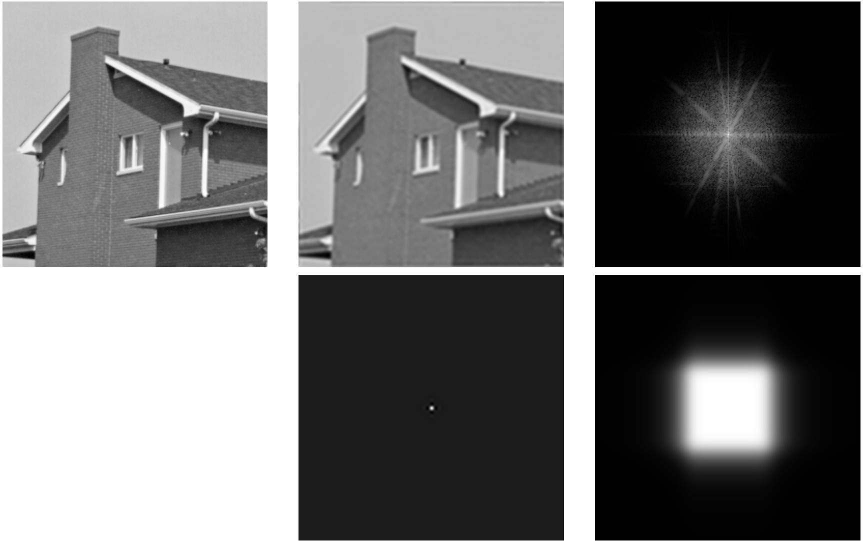


$|\hat{g}_5|$

Anti-Aliasing Filters



Anti-Aliasing Filters



Different Kinds of Filtering

Consider the spectral domain

$$\Theta = [-\tfrac{M}{2}, \tfrac{M}{2}] \times [-\tfrac{N}{2}, \tfrac{N}{2}].$$

- The **low frequencies** correspond to $(\xi, \zeta) \in \Theta$ located near $(0, 0)$.
- The **high frequencies** correspond to $(\xi, \zeta) \in \Theta$ with $\max(|\xi|, |\zeta|)$ large.

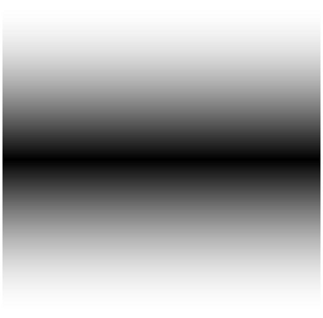
In practice, there are different kinds of linear filtering operators depending on $\hat{k}$:

- **Low-pass filtering** can be used for smoothing or removing noise (see later).
- **High-pass** filtering can be used for contour extraction.
- **Band-pass filtering** can be used for fine analysis (e.g. texture extraction).

Derivatives seen as high-pass filter

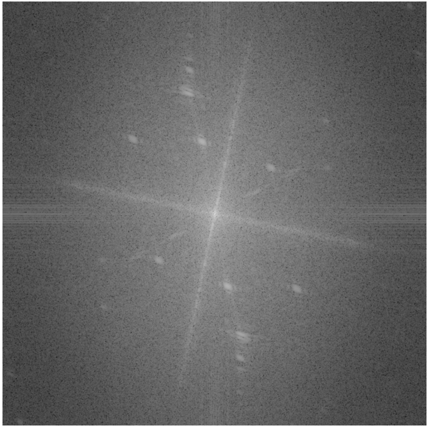
u

$\partial_1 u = k_1 * u$

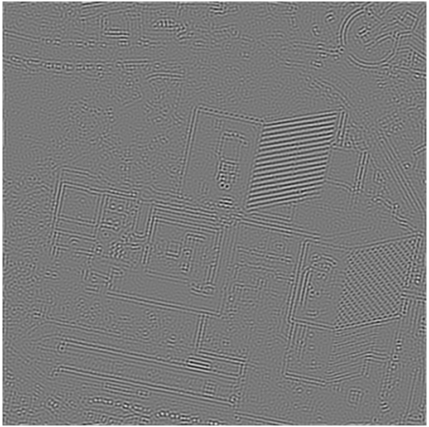


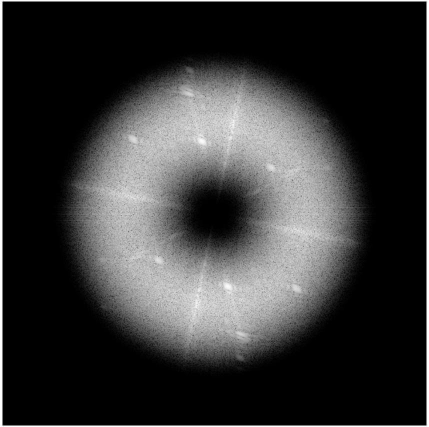
$|\hat{k}_1|$

Example of Bandpass Filtering



Example of Bandpass Filtering





Outline

Linear Filtering

Linear Filtering in Fourier Domain

Mathematical Morphology

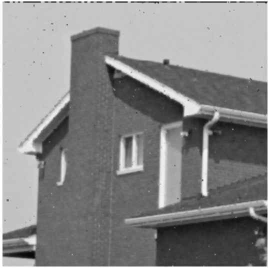
A Simple Non-linear Filter: the Median filter

For the median filter, the output value at pixel **x** is the median of neighbooring pixels:

$$u(\mathbf{x}) = \text{Median}\Big(\ u(\mathbf{x}+\mathbf{a})\ ,\ \mathbf{a} \in \omega\Big),$$

where $\omega = [-1, 1]^2$.

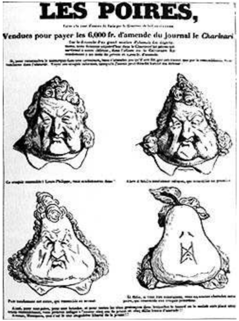
The median filter is useful to remove sparse noise, like “salt and pepper” noise.



Mathematical Morphology

- Linear filtering (with convolution) relies on a linear structure on the image space $(+)$
- Mathematical filters will rely on an order between image values $(\leq)$
- This is interesting because in natural images, edges often correspond to "occlusion":
  that is, a 2D image is obtained by a projection of a 3D scene where objects occlude each other.
- Also, we will define operators that do progressive simplification of images.

x

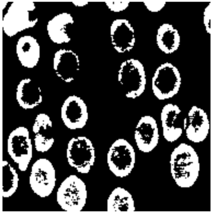


Binary v.s. Continuous Image

- First, we will work with binary image $u : \Omega \to \{0, 1\}$ (0 = False, 1 = True).
- A binary image u encodes a set

$$X_u = \{ \mathbf{x} \in \Omega \mid u(\mathbf{x}) = 1 \}.$$

- If $\vee$ is the "OR" operator (applied pixel by pixel), then $X_u \cup X_v = X_{u \vee v}$.
- If $\wedge$ is the "AND" operator (applied pixel by pixel), then $X_u \cap X_v = X_{u \wedge v}$.

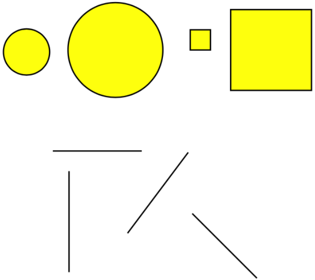


Binary image

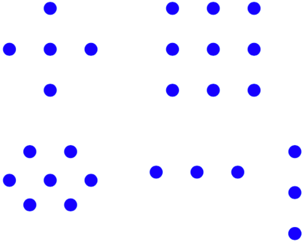
Graylevel image

Structuring Element

- We will work either on **discrete** domain $\Omega \subset \mathbb{Z}^2$ or **continuous** domain $\Omega \subset \mathbf{R}^2$.
- Morphological filters will rely on a **structuring element**, denoted as $B \subset \mathbb{Z}^2$ or $\mathbf{R}^2$.
- It is analogous to the ω neighborhood of $(0, 0)$ used for convolutions.



Continous



Discrete

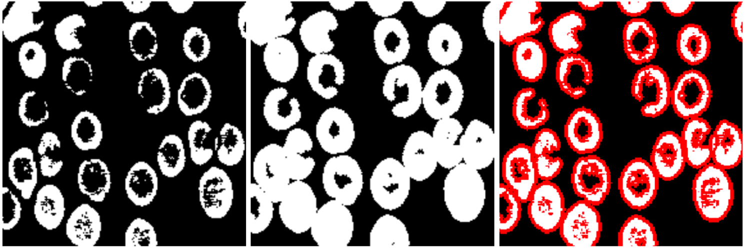
Binary Dilation

Warning: Ω is the ambient space (i.e. all sets are written as subsets of Ω).

Definition

The binary dilation of $X \subset \Omega$ by B is defined by

$$D(X, B) = X \oplus B = \{ \mathbf{x} + \mathbf{b} \mid \mathbf{x} \in X, \mathbf{b} \in B \} = \cup_{x \in X}(\mathbf{x} + B).$$



Left: X. Middle: Dilation $D(X, B)$ by a disk B. Right: Difference between original and dilation.

Binary Erosion

Warning: Ω is the ambient space (i.e. all sets are written as subsets of Ω).

Definition

The binary erosion of $X \subset \Omega$ by B is defined by

$$E(X, B) = X \ominus B = \{ \mathbf{x} \in \Omega \mid \mathbf{x} + B \subset X \}.$$

Left: X. Middle: Erosion $E(X, B)$ by a disk B. Right: Difference between original and erosion.

Properties of Binary Dilation and Erosion

- Dilation and Erosion are both non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad D(X,B) \subset D(Y,B) \quad \text{and} \quad E(X,B) \subset E(Y,B).$$

Properties of Binary Dilation and Erosion

- Dilation and Erosion are both non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad D(X, B) \subset D(Y, B) \quad \text{and} \quad E(X, B) \subset E(Y, B).$$

- Dilation commutes with union, but not with intersection:

$$D(X \cup Y, B) = D(X, B) \cup D(Y, B) \quad , \quad D(X \cap Y, B) \subset D(X, B) \cap D(Y, B).$$

Properties of Binary Dilation and Erosion

- Dilation and Erosion are both non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad D(X,B) \subset D(Y,B) \quad \text{and} \quad E(X,B) \subset E(Y,B).$$

- Dilation commutes with union, but not with intersection:

$$D(X \cup Y, B) = D(X,B) \cup D(Y,B) \quad , \quad D(X \cap Y, B) \subset D(X,B) \cap D(Y,B).$$

- Erosion commutes with intersection, but not with union:

$$E(X \cap Y, B) = E(X,B) \cap E(Y,B) \quad , \quad E(X \cup Y, B) \supset E(X,B) \cup E(Y,B).$$

Properties of Binary Dilation and Erosion

- Dilation and Erosion are both non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad D(X,B) \subset D(Y,B) \quad \text{and} \quad E(X,B) \subset E(Y,B).$$

- Dilation commutes with union, but not with intersection:

$$D(X \cup Y, B) = D(X,B) \cup D(Y,B) \quad , \quad D(X \cap Y, B) \subset D(X,B) \cap D(Y,B).$$

- Erosion commutes with intersection, but not with union:

$$E(X \cap Y, B) = E(X,B) \cap E(Y,B) \quad , \quad E(X \cup Y, B) \supset E(X,B) \cup E(Y,B).$$

- Iteration property:

$$D(D(X,B),B') = D(X, B \oplus B') \quad , \quad E(E(X,B),B') = E(X, B \oplus B').$$

Properties of Binary Dilation and Erosion

- Dilation and Erosion are both non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad D(X,B) \subset D(Y,B) \quad \text{and} \quad E(X,B) \subset E(Y,B).$$

- Dilation commutes with union, but not with intersection:

$$D(X \cup Y, B) = D(X,B) \cup D(Y,B) \quad , \quad D(X \cap Y, B) \subset D(X,B) \cap D(Y,B).$$

- Erosion commutes with intersection, but not with union:

$$E(X \cap Y, B) = E(X,B) \cap E(Y,B) \quad , \quad E(X \cup Y, B) \supset E(X,B) \cup E(Y,B).$$

- Iteration property:

$$D(D(X,B),B') = D(X, B \oplus B') \quad , \quad E(E(X,B),B') = E(X, B \oplus B').$$

- If $0 \in B$, then $B \subset D(X,B)$ (extensive), and $E(X,B) \subset B$ (anti-extensive).

Properties of Binary Dilation and Erosion

- Dilation and Erosion are both non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad D(X,B) \subset D(Y,B) \quad \text{and} \quad E(X,B) \subset E(Y,B).$$

- Dilation commutes with union, but not with intersection:

$$D(X \cup Y, B) = D(X,B) \cup D(Y,B) \quad , \quad D(X \cap Y, B) \subset D(X,B) \cap D(Y,B).$$

- Erosion commutes with intersection, but not with union:

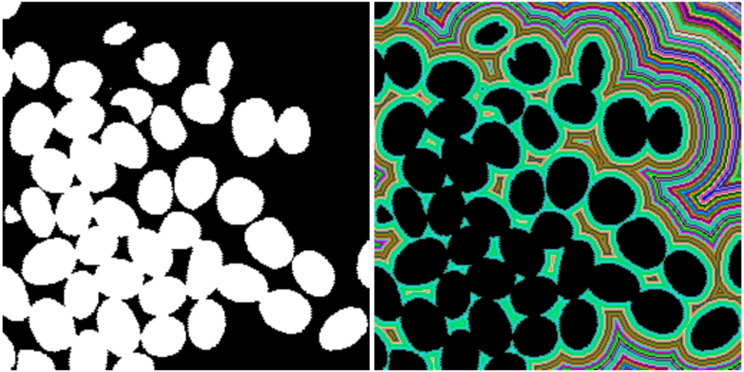
$$E(X \cap Y, B) = E(X,B) \cap E(Y,B) \quad , \quad E(X \cup Y, B) \supset E(X,B) \cup E(Y,B).$$

- Iteration property:

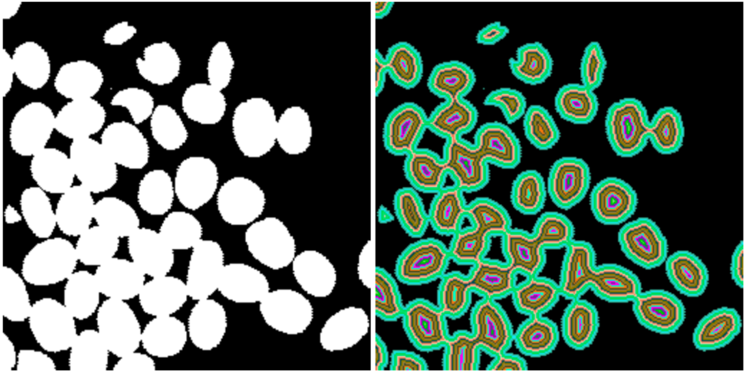
$$D(D(X,B),B') = D(X, B \oplus B') \quad , \quad E(E(X,B),B') = E(X, B \oplus B').$$

- If $0 \in B$, then $B \subset D(X,B)$ (extensive), and $E(X,B) \subset B$ (anti-extensive).
- $E(X,B)^c = D(X^c, B)$

Relation with Distances



Relation with Distances

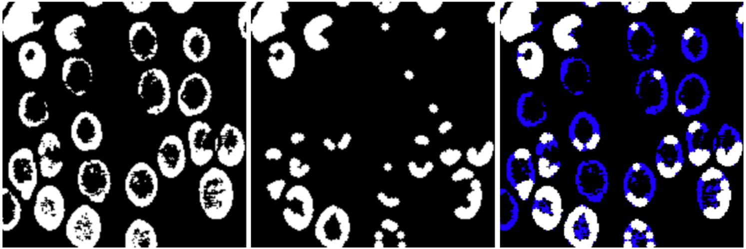


Binary Opening

Definition

The opening of $X \subset \Omega$ by B is defined by

$$X_B = D(E(X, B), B).$$



Left: X. Middle: Opening $D(X, B)$ by a disk B. Right: Difference between original and opening.

Binary Closure

Definition

The binary closure of $X \subset \Omega$ by B is defined by

$$X^B = E(D(X, B), B).$$

Left: X. Middle: Closure $E(X, B)$ by a disk B. Right: Difference between original and closure.

Properties of Binary Opening and Closure

- Binary opening and closure are non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad X_B \subset Y_B \ \text{ and } \ X^B \subset Y^B.$$

Properties of Binary Opening and Closure

- Binary opening and closure are non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad X_B \subset Y_B \quad \text{and} \quad X^B \subset Y^B.$$

- Binary closure is extensive $X \subset X^B$.

Properties of Binary Opening and Closure

- Binary opening and closure are non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad X_B \subset Y_B \quad \text{and} \quad X^B \subset Y^B.$$

- Binary closure is extensive $X \subset X^B$.
- Binary opening is anti-extensive $X_B \subset X$.

Properties of Binary Opening and Closure

- Binary opening and closure are non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad X_B \subset Y_B \quad \text{and} \quad X^B \subset Y^B.$$

- Binary closure is extensive $X \subset X^B$.
- Binary opening is anti-extensive $X_B \subset X$.
- Binary opening and closure are idempotent:

$$(X_B)_B = X_B \quad , \quad (X^B)^B = X^B.$$

Properties of Binary Opening and Closure

- Binary opening and closure are non-decreasing with respect to X:

$$X \subset Y \quad \Rightarrow \quad X_B \subset Y_B \quad \text{and} \quad X^B \subset Y^B.$$

- Binary closure is extensive $X \subset X^B$.
- Binary opening is anti-extensive $X_B \subset X$.
- Binary opening and closure are idempotent:

$$(X_B)_B = X_B \quad , \quad (X^B)^B = X^B.$$

- $(X^B)^c = (X^c)_B$

From Binary to Grayscale Morphology

- The order between sets $(\subset, \supset)$ corresponds to order on indicator functions $(\leq, \geq)$.
- The union/intersection correspond to max/min on indicator functions.
- We will thus generalize morphological filters to graylevel images with sup/inf operations.

Dilation

Definition

The dilation of $u : \Omega \to \mathbf{R}$ by B is an image $D(u, B)$ defined by

$$\forall \mathbf{x} \in \Omega, \quad D(u, B)(\mathbf{x}) = \sup_{\mathbf{b} \in B | \mathbf{x}+\mathbf{b} \in \Omega} u(\mathbf{x} + \mathbf{b}).$$

Dilation by a disk B

Erosion

Definition

The erosion of $u : \Omega \to \mathbf{R}$ by B is an image $E(u, B)$ defined by

$$\forall \mathbf{x} \in \Omega, \quad E(u, B)(\mathbf{x}) = \inf_{\mathbf{b} \in B | \mathbf{x} + \mathbf{b} \in \Omega} u(\mathbf{x} + \mathbf{b}).$$

Erosion by a disk B

Opening

Definition

The opening of $u : \Omega \to \mathbf{R}$ by B is the image $u_B = D(E(u, B))$.

Opening by a disk B

Closure

Definition

The closure of $u : \Omega \to \mathbf{R}$ by B is the image $u^B = E(D(u, B))$.

Closure by a disk B

Mathematical Properties

- The properties of dilation, erosion, opening, closure extends to the grayscale case.
- We have the iteration property:

$$D(D(u,B),B') = D(u, B \oplus B') \quad , \quad E(E(u,B),B') = E(u, B \oplus B').$$

- Also, opening and closure are still idempotent:

$$(u_B)_B = u_B$$

$$(u^B)^B = u^B$$

NB: The iteration property holds in the discrete AND continuous case.
In the discrete case, $B \oplus B'$ is a discrete sum (but discrete disks may not look like disks...).

Iterated Filters

- Let us iterate opening-closure $u \mapsto (u_B)^B$ with larger and larger squares B.
- This allows to compute successive morphological sketches of the image.

Iterated Filters

- Let us iterate opening-closure $u \mapsto (u_B)^B$ with larger and larger squares B.
- This allows to compute successive morphological sketches of the image.

Iterated Filters

- Let us iterate opening-closure $u \mapsto (u_B)^B$ with larger and larger squares B.
- This allows to compute successive morphological sketches of the image.

Iterated Filters

- Let us iterate opening-closure $u \mapsto (u_B)^B$ with larger and larger squares B.
- This allows to compute successive morphological sketches of the image.

Iterated Filters

- Let us iterate opening-closure $u \mapsto (u_B)^B$ with larger and larger squares B.
- This allows to compute successive morphological sketches of the image.

Morphological Gradient

The morphological gradient of u w.r.t. B is defined as $D(u, B) - E(u, B)$.

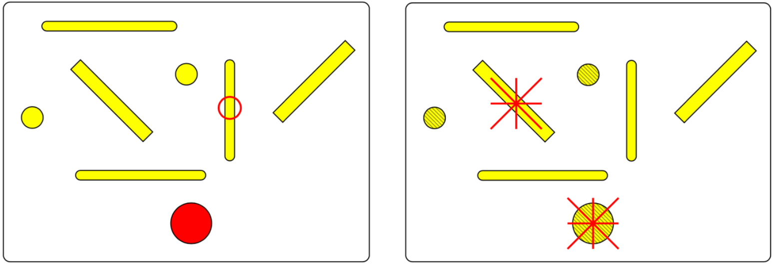
Top-hat Transform

- The top-hat transform is defined as $u - u_B$.
- It can be used to highlight edges or salient details.

u $\quad$ u_B $\quad$ $u - u_B$

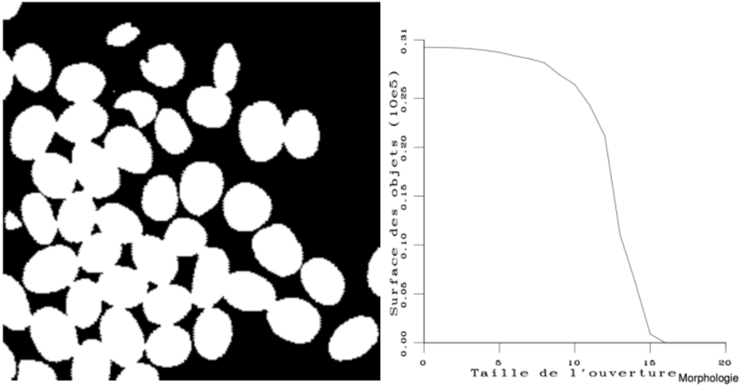
Choice of Structuring Element

- Of course, the type of highlighted details depends on the choice of structuring element.



Granulometry

- Summing binary erosions with larger balls allows to perform granulometry.
- This allows to extract "characteristic scales" of objects in the image.



Conclusion

- You've discovered linear and non-linear filtering operators.
- Linear filtering is a convolution by a kernel k.
- Linear filtering can be computed in Fourier domain with the DFT.
- Linear filtering can model many imaging operators (derivatives, blur, sharpening filters, ...)
- Morphological operators perform non-linear filtering which allows to better preserve the edges.
- Linear filters and Morphological filters can be used for image analysis.
  (detection of edges or simple shapes)

Credits for illustrations: Isabelle Bloch, Christophe Kervazo, Alasdair Newson

THANK YOU FOR YOUR ATTENTION!