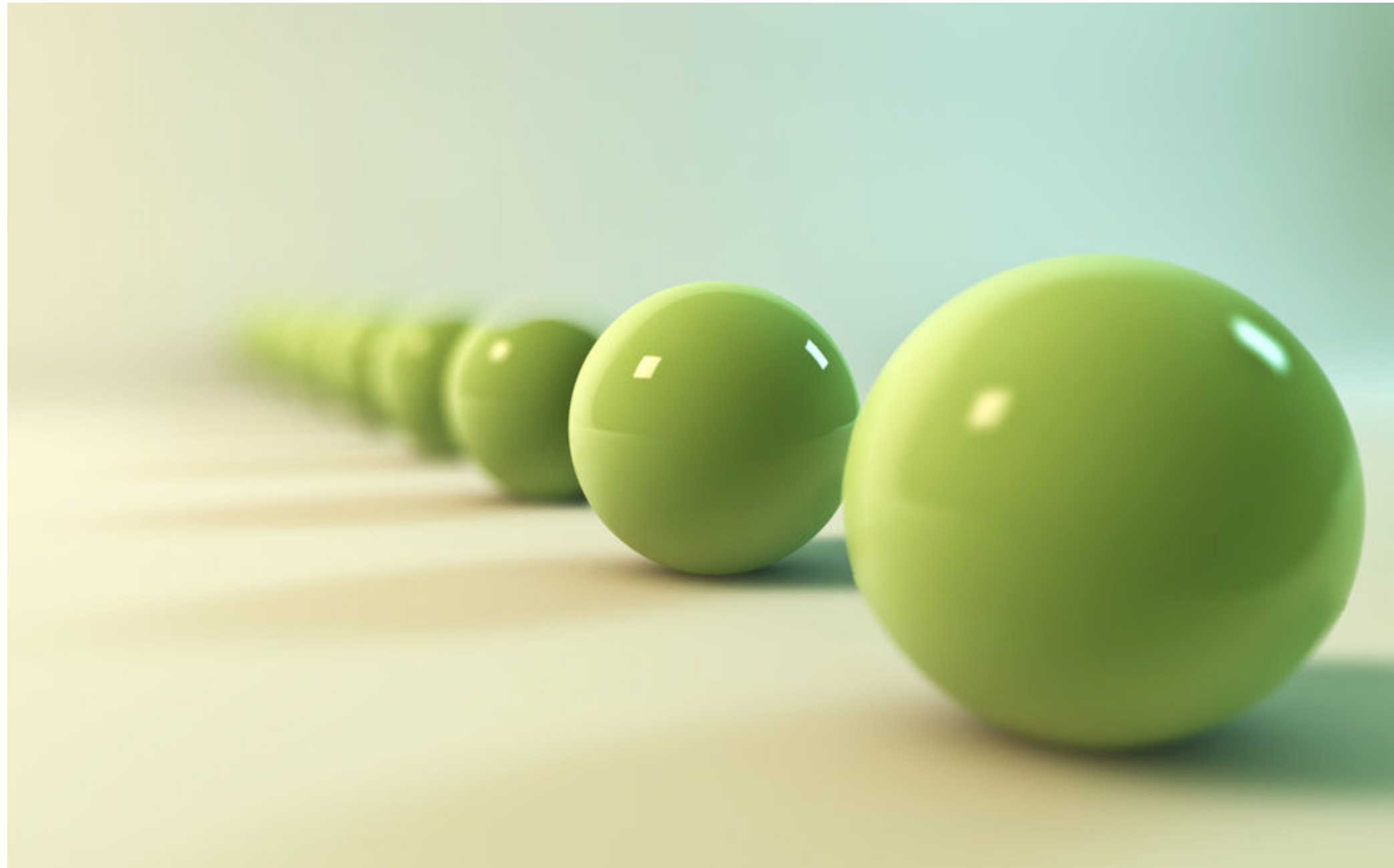


Filtrage dans le domaine spatial



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

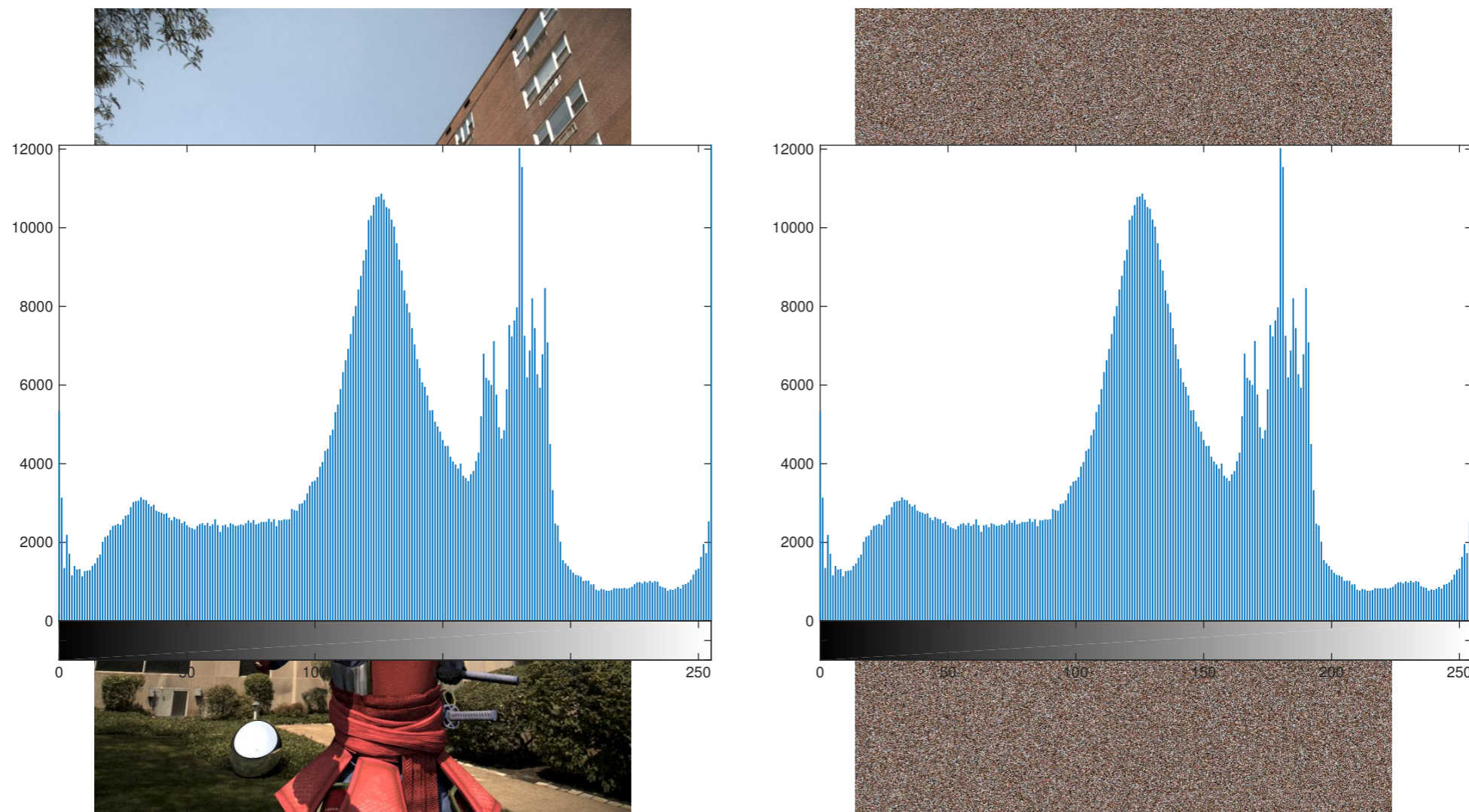
Limites des fonctions sur les pixels

Qu'arrive-t-il à l'histogramme d'une image lorsque les pixels sont ré-ordonnés?



Limites des fonctions sur les pixels

Qu'arrive-t-il à l'histogramme d'une image lorsque les pixels sont ré-ordonnés?

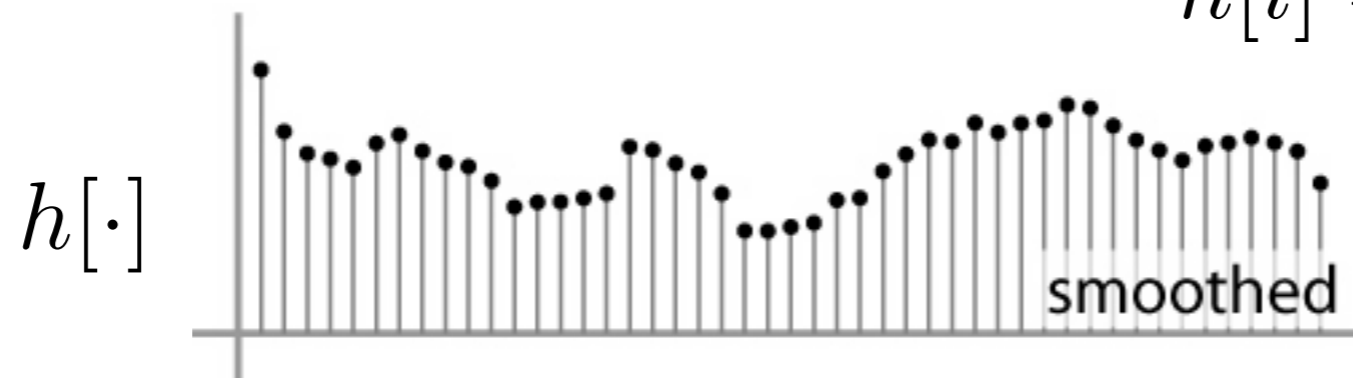
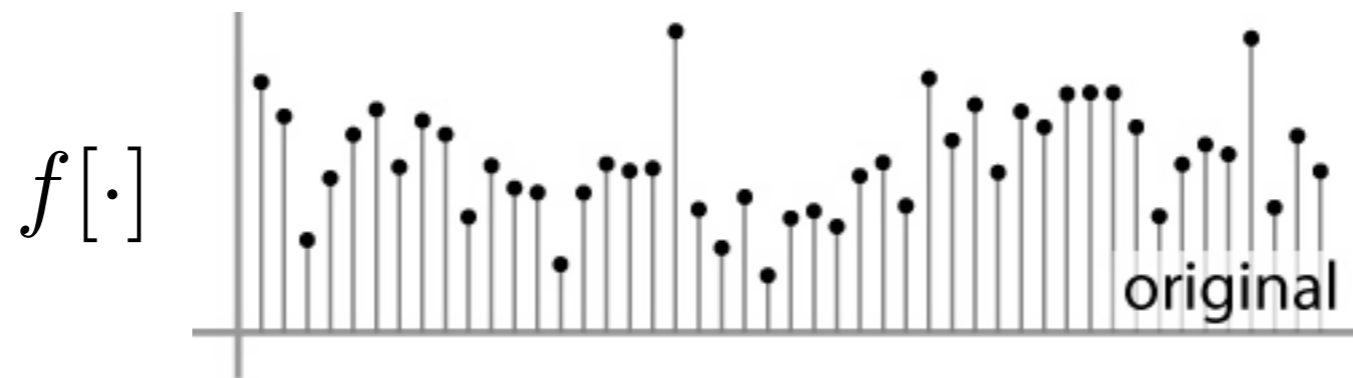


Filtrage d'images

- Filtrage: fonction d'un pixel et de ses voisins
- Très important!
 - Modifier l'image
 - Réduire le bruit, re-dimensionner, contraste, etc.
 - Extraire de l'information
 - Texture, arêtes, points d'intérêt, etc.
 - Detecter des formes
 - «template matching»

Moyenne temporelle (1D)

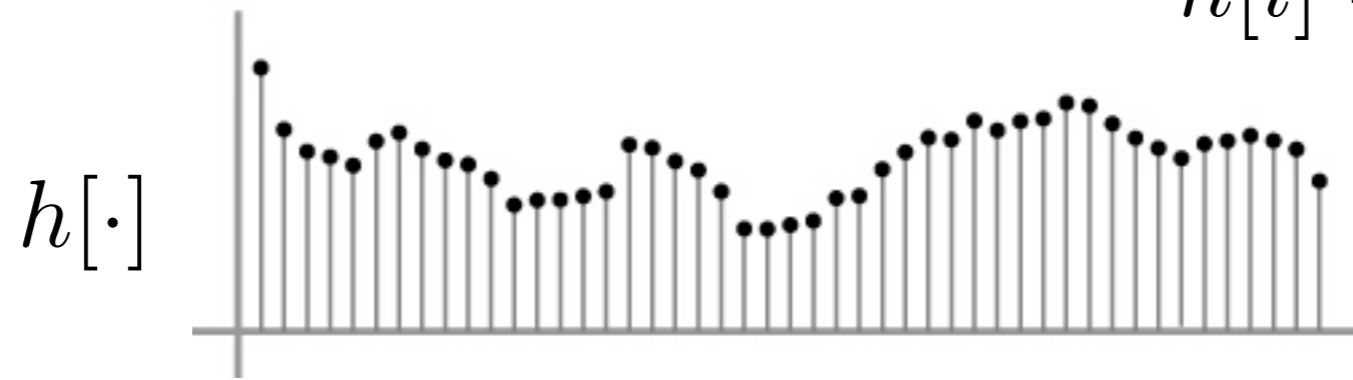
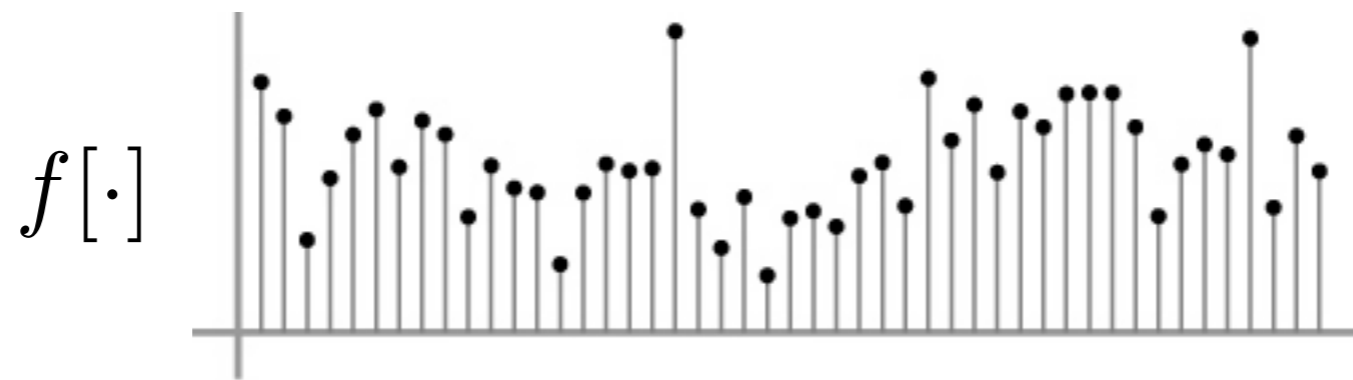
- idée: définir une fonction qui calcule la moyenne sur un intervalle
- déplacer l'intervalle au fil du temps, et calculer le résultat



$$h[i] = \sum_{u=-k}^k g[u] f[i + u]$$

Moyenne temporelle (1D)

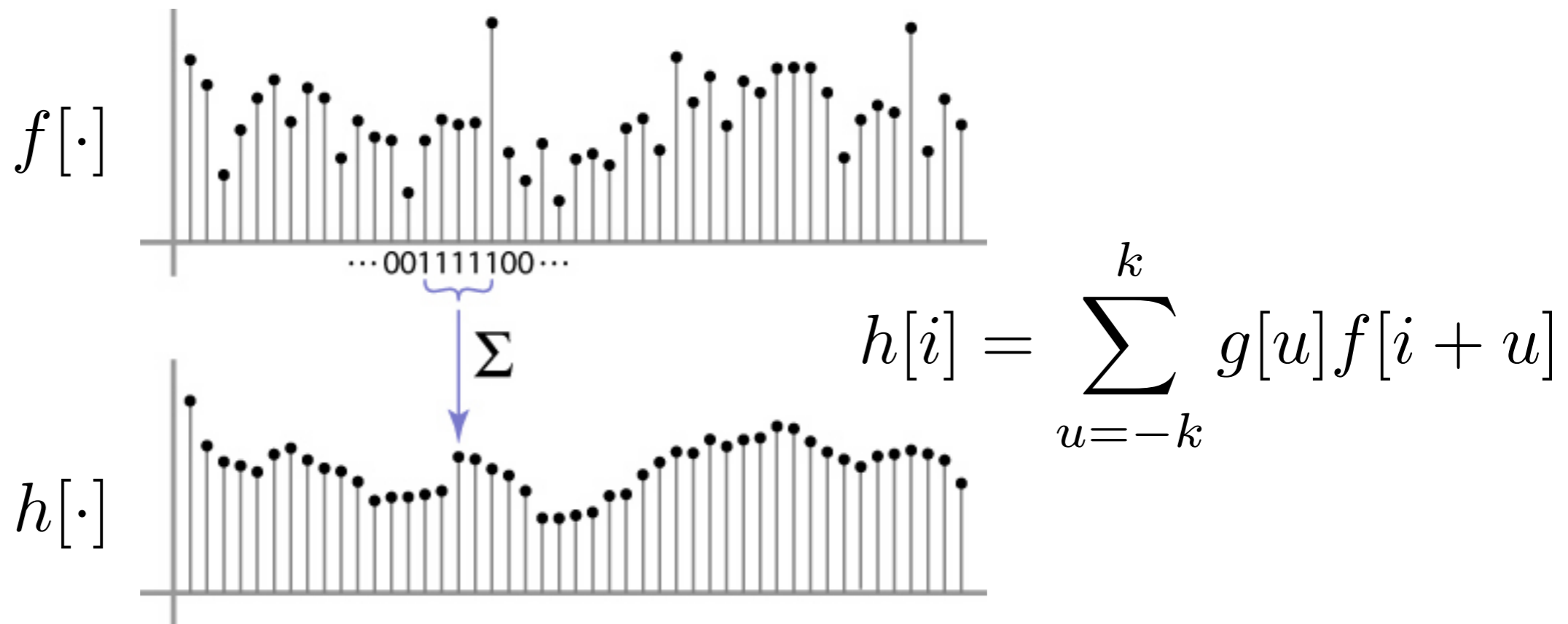
- Quel «filtre» g doit-on utiliser pour calculer la moyenne sur 5 points?
- $g = [\dots, 0, 1, 1, 1, 1, 1, 0, \dots] / 5$



$$h[i] = \sum_{u=-k}^k g[u] f[i + u]$$

Moyenne temporelle (1D)

- Quel «filtre» g doit-on utiliser pour calculer la moyenne sur 5 points?
- $g = [\dots, 0, 1, 1, 1, 1, 1, 0, \dots] / 5$



Corrélation croisée (en 2D)

- Nous avons:
 - une image f
 - un filtre g , de taille $(2k+1) \times (2k+1)$

- h est le résultat de

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

- Cette opération est la corrélation croisée («cross-correlation»)
- Produit scalaire entre le filtre g et l'image f à chaque emplacement sur l'image

$g[\cdot, \cdot]$

9 | 1

1	1	1
1	1	1
1	1	1

Filterage

$$g[\cdot, \cdot] \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$f[\cdot, \cdot]$$

$$h[\cdot, \cdot]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

		-							

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

Filtrage

$$g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f[\cdot, \cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[\cdot, \cdot]$

	0	10							

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

Filtrage

$$g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f[\cdot, \cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[\cdot, \cdot]$

	0	10	20						

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

Filtrage

$$g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f[\cdot, \cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[\cdot, \cdot]$

	0	10	20	30					

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

Filtrage

$$g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f[\cdot, \cdot]$

$h[\cdot, \cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30					
				?					

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

Filtrage

$$g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f[\cdot, \cdot]$

$h[\cdot, \cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0

	0	10	20	30					

Qu'est-ce qui se passe, intuitivement?

0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

				80					

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

Filtrage

$$g[\cdot, \cdot] \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

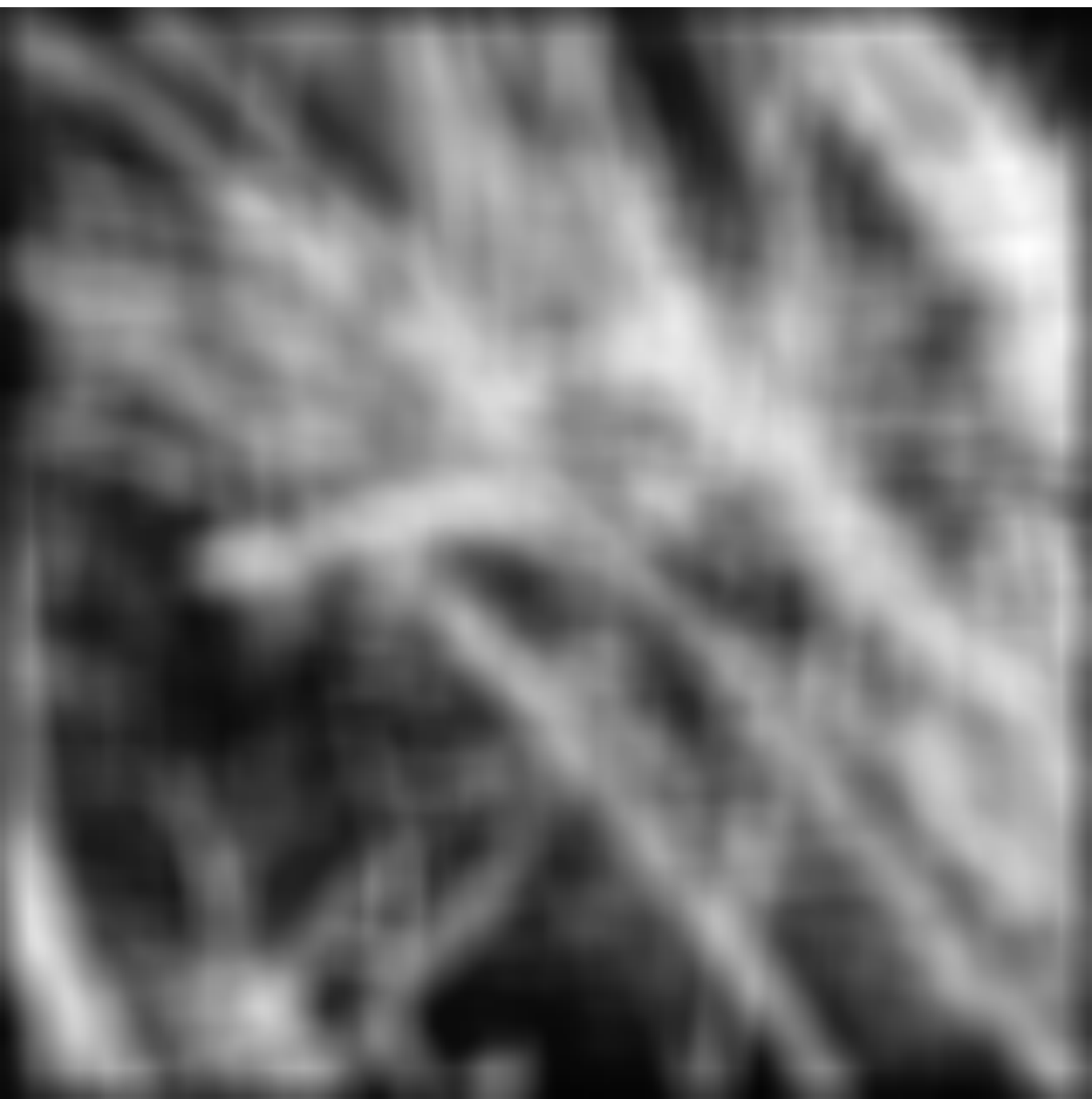
Filtrage “en boîte” (box filter)

- Remplace chaque pixel par la moyenne de son voisinage
- On adoucit l'image (enlève les hautes fréquences)

$$\frac{1}{9} g[\cdot, \cdot]$$

1	1	1
1	1	1
1	1	1

“Atténuer” l’image avec le filtre boîte



Petite pratique avec les filtres linéaires

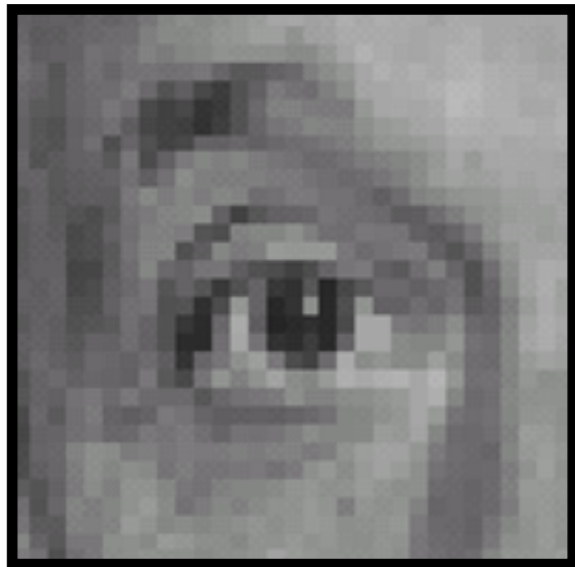


Image originale

0	0	0
0	1	0
0	0	0

?

Petite pratique avec les filtres linéaires

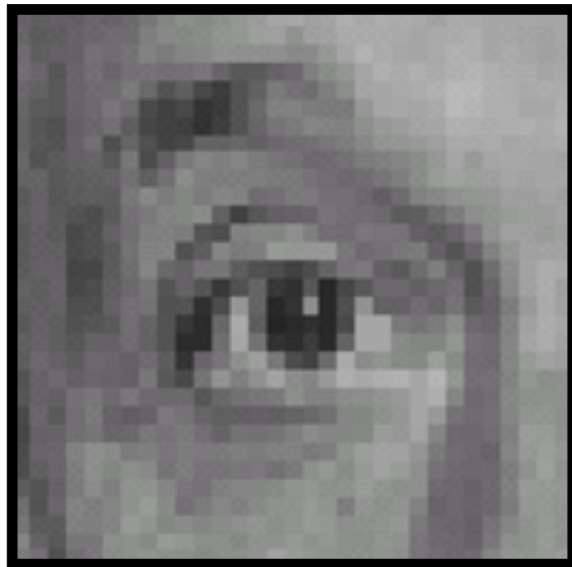
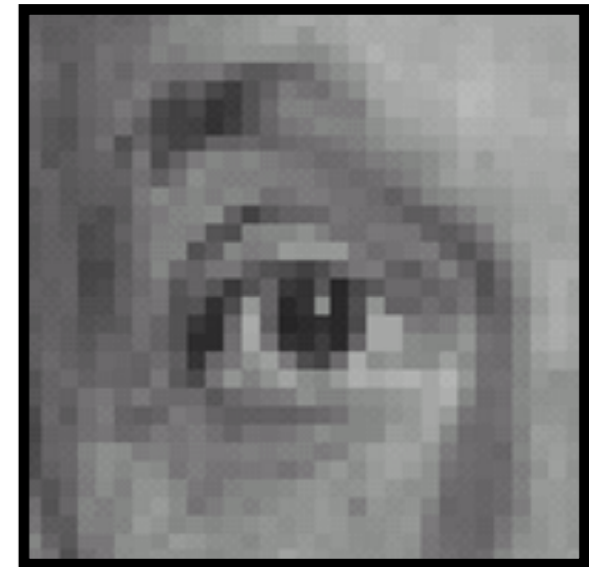


Image originale

0	0	0
0	1	0
0	0	0



Résultat
(identique!)

Petite pratique avec les filtres linéaires

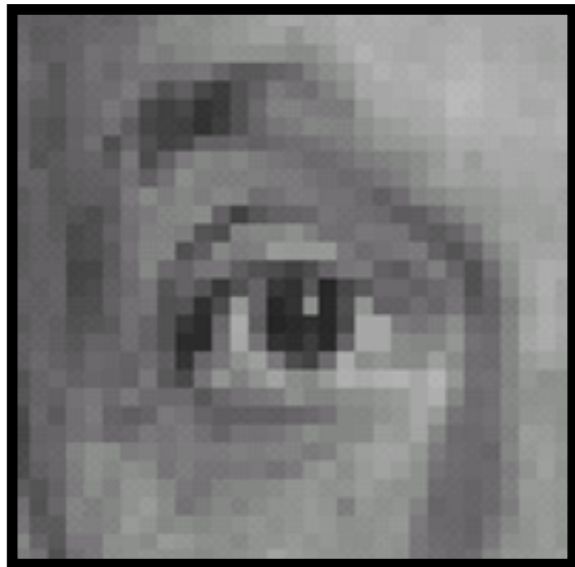


Image originale

0	0	0
0	0	1
0	0	0

?

Petite pratique avec les filtres linéaires

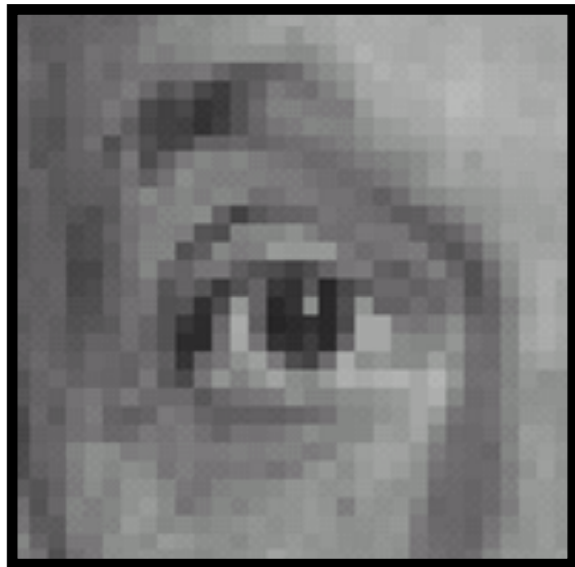
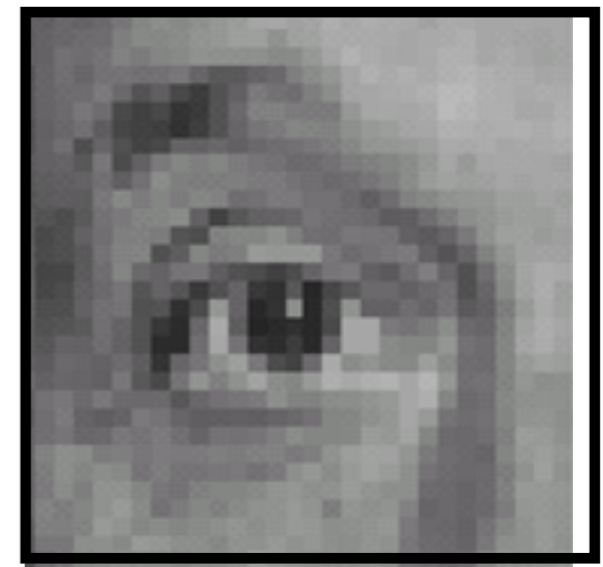


Image originale

0	0	0
0	0	1
0	0	0



À gauche de 1 pixel

Petite pratique avec les filtres linéaires

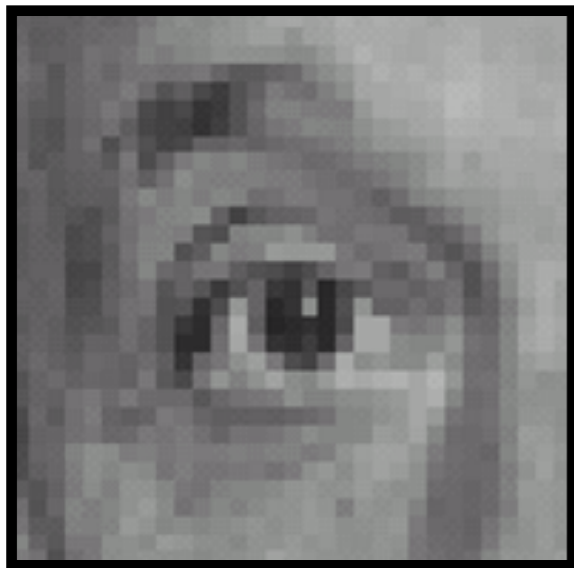


Image originale

0	0	0
0	2	0
0	0	0

-

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

?

Petite pratique avec les filtres linéaires

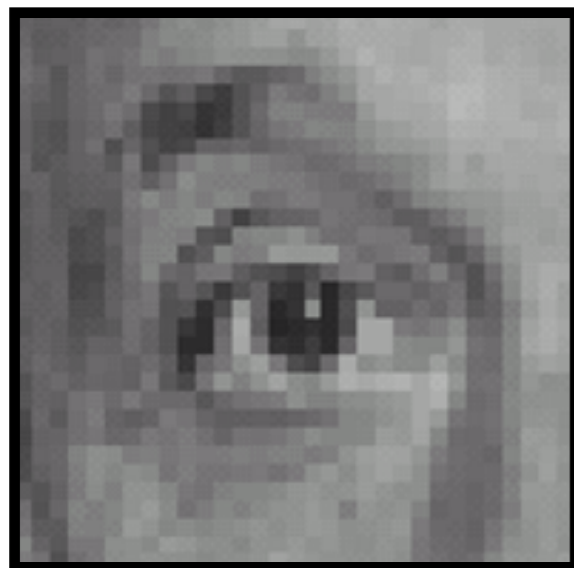


Image originale

0	0	0
0	2	0
0	0	0

-

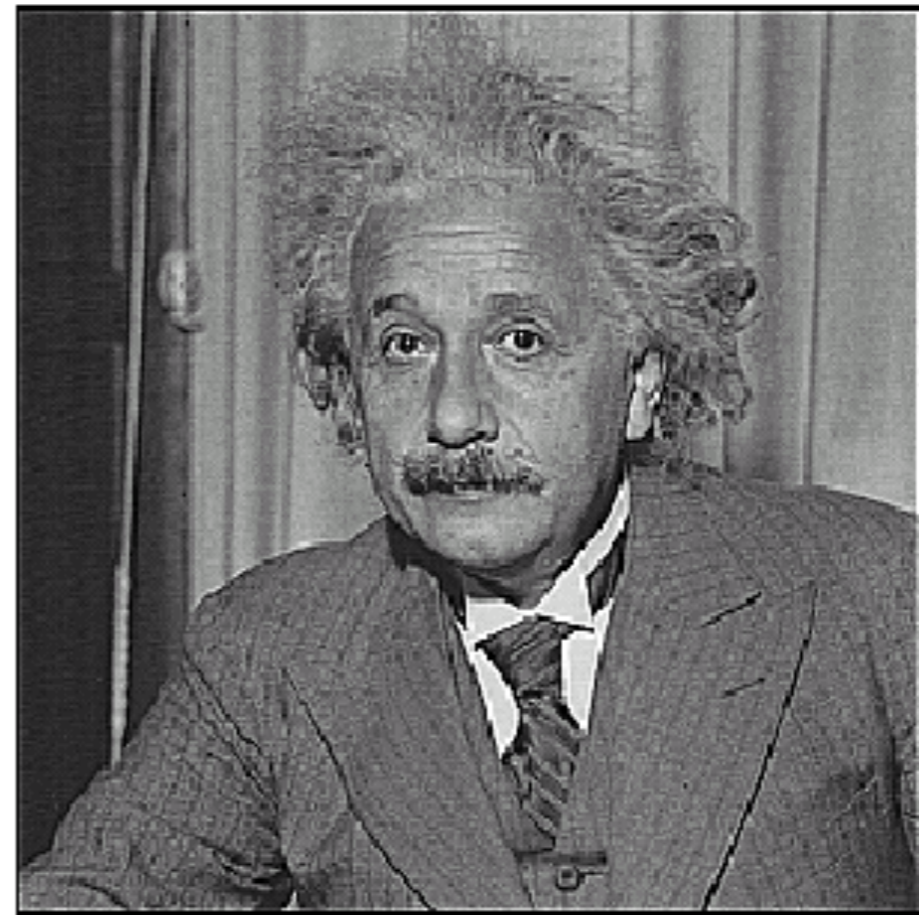
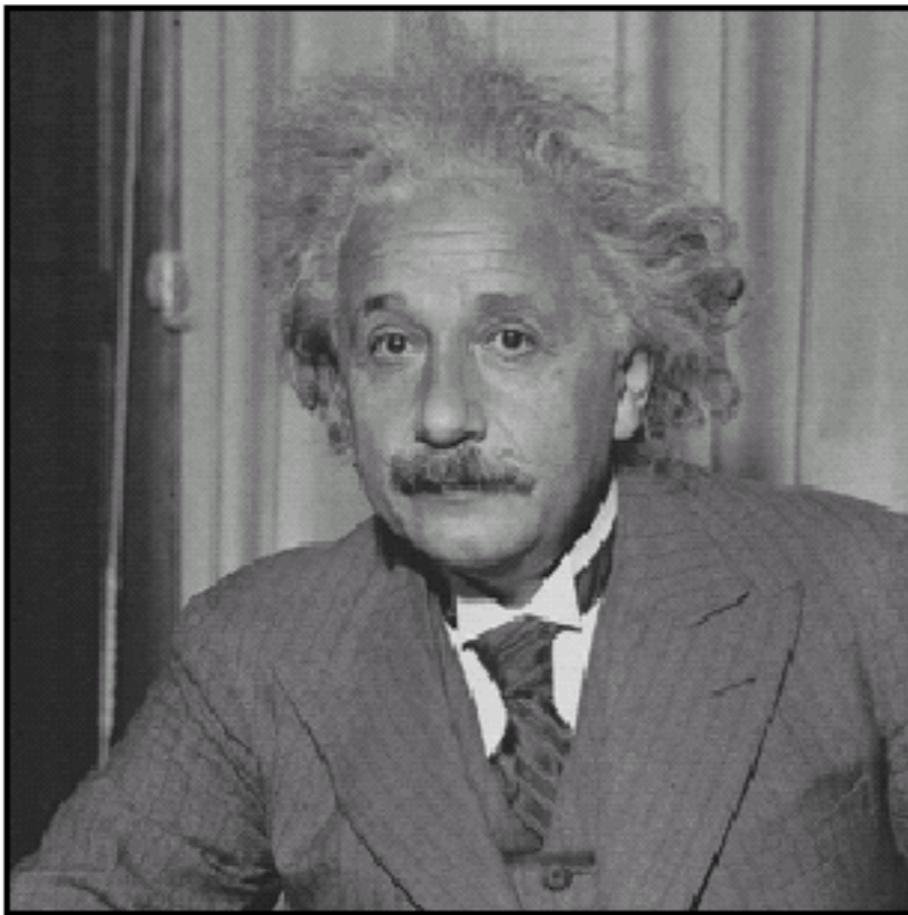
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

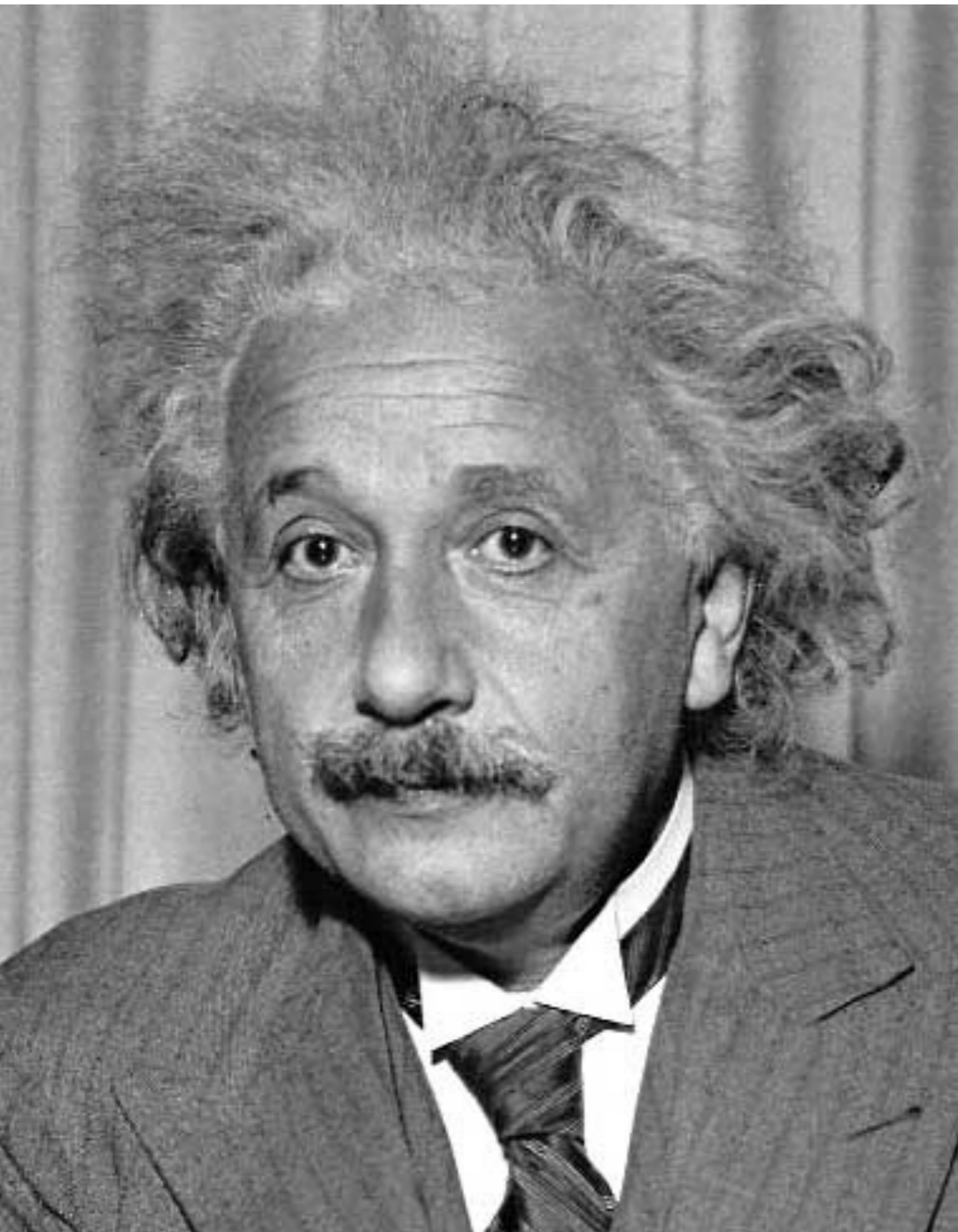


Accentue les différences par rapport à la moyenne

Accentuation “sharpening”

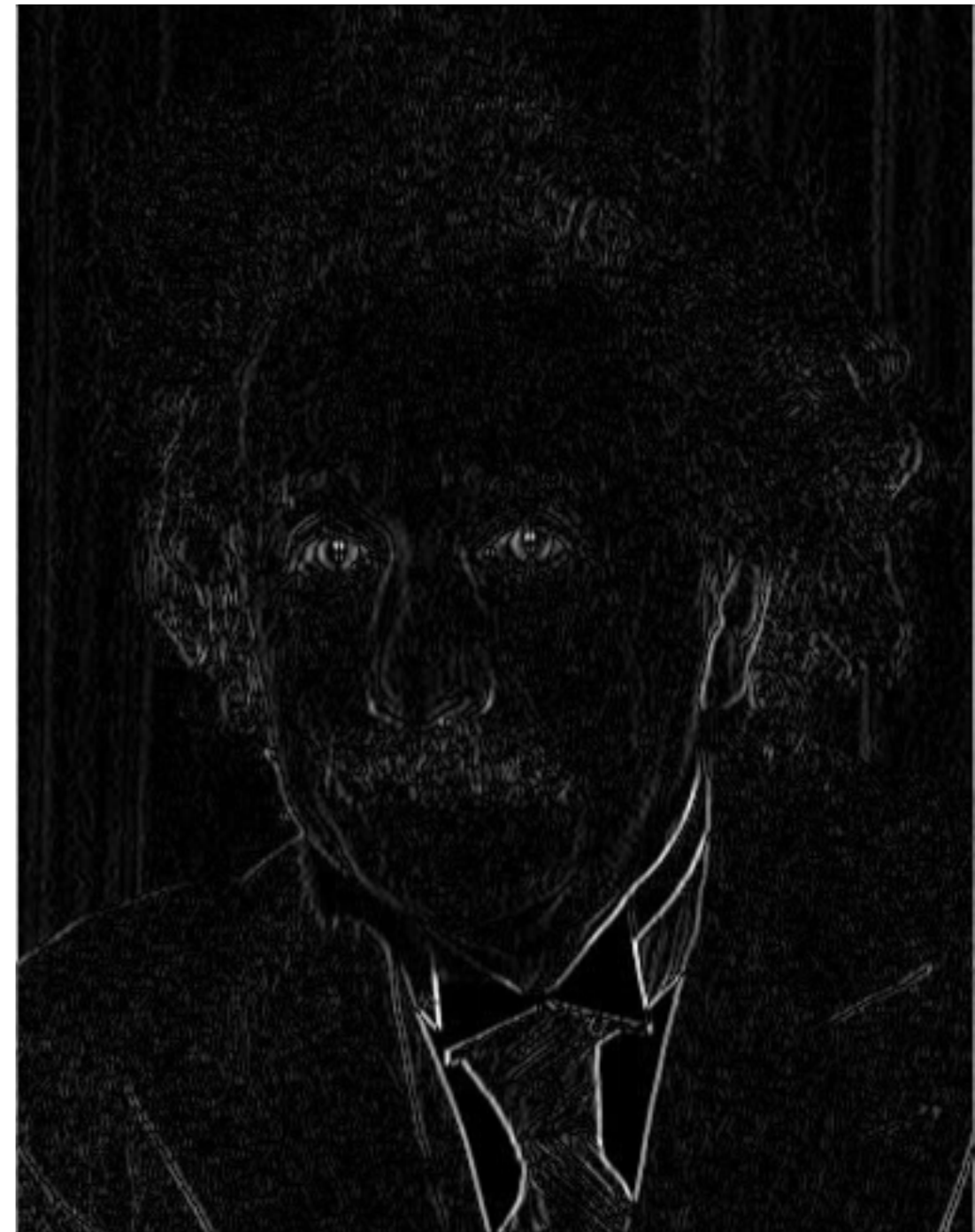


Autres filtres



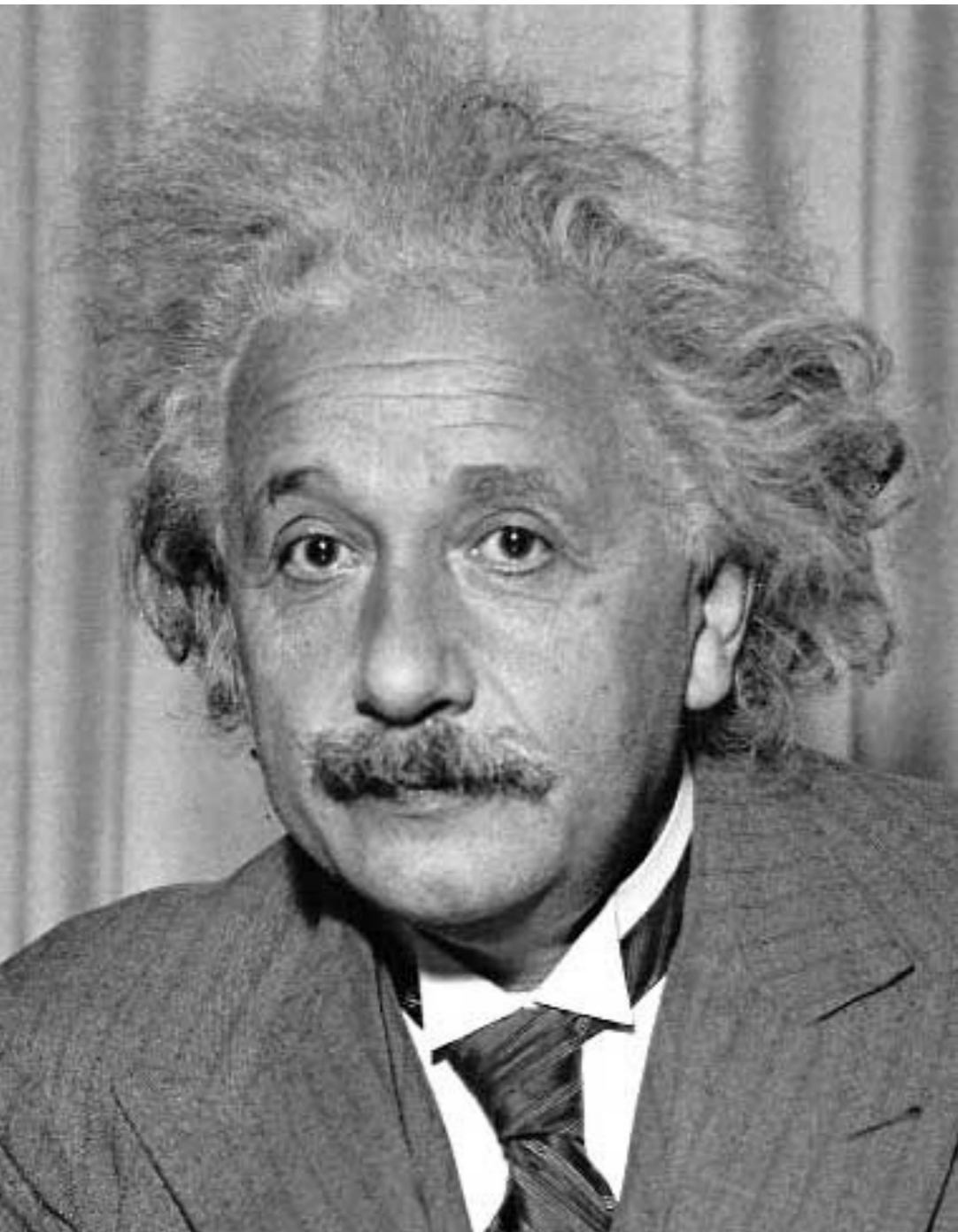
1	0	-1
2	0	-2
1	0	-1

Sobel



Arêtes verticales
(valeur absolue)

Autres filtres



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

Sobel



Arêtes horizontales
(valeur absolue)

Démo

Filtrage

Corrélation vs. convolution

- La **corrélacion croisée** («cross-correlation») est une mesure de similarité entre deux signaux

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i + u, j + v]$$

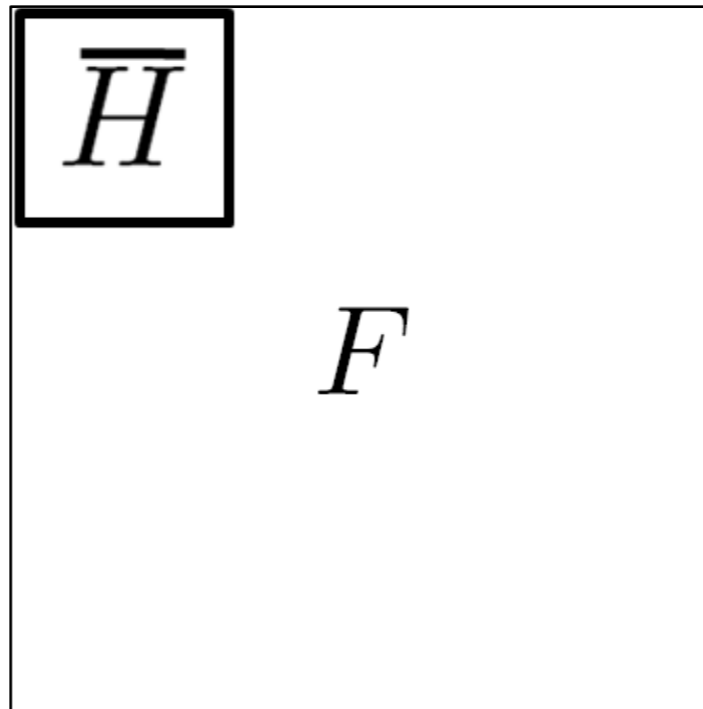
- La **convolution** est un opérateur qui «applique» un signal (filtre) à un autre (e.g. image). Mathématiquement, cela équivaut à une corrélation croisée où le filtre est inversé (horizontalement et verticalement) avant d'être appliqué à l'image:

$$h[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k g[u, v] f[i - u, j - v]$$

- On l'écrit:

$$h = g * f$$

Convolution



Propriété des filtres linéaires

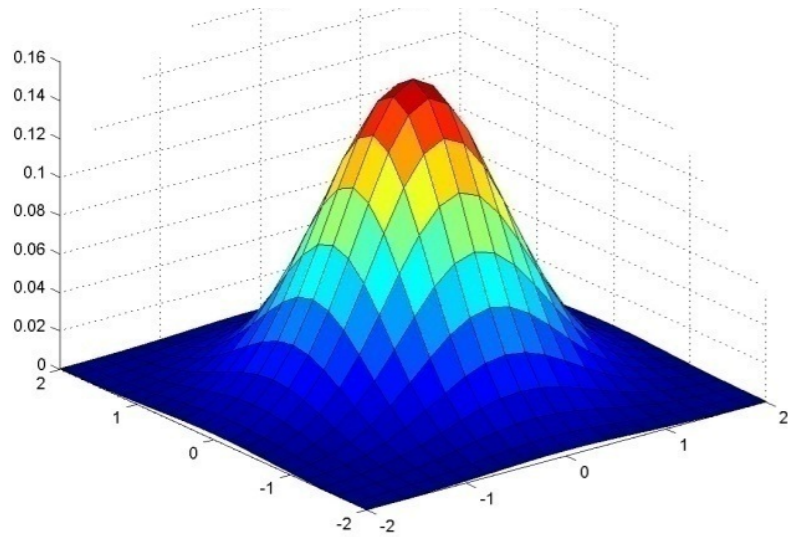
- Linéarité (!):
 - $\text{filtre}(f1 + f2) = \text{filtre}(f1) + \text{filtre}(f2)$
- Invariance aux déplacements: ne dépend pas de l'emplacement du pixel
 - $\text{filtre}(\text{shift}(f)) = \text{shift}(\text{filtre}(f))$

Propriétés des filtres linéaires

- Commutatif: $a * b = b * a$
 - On peut donc traiter l'image comme un filtre
- Associatif: $a * (b * c) = (a * b) * c$
 - Souvent, on applique plusieurs filtres: $((a * b_1) * b_2) * b_3$
 - Équivalent à filtrer l'image avec le "filtre du filtre": $a * (b_1 * b_2 * b_3)$
- Distribution: $a * (b + c) = (a * b) + (a * c)$
- Factoriser les scalaires: $ka * b = a * kb = k(a * b)$
- Identité $e = [0, 0, 1, 0, 0]$, $a * e = a$

Filtre important: gaussien

- Pondere les contributions des voisins en fonction de leur distance

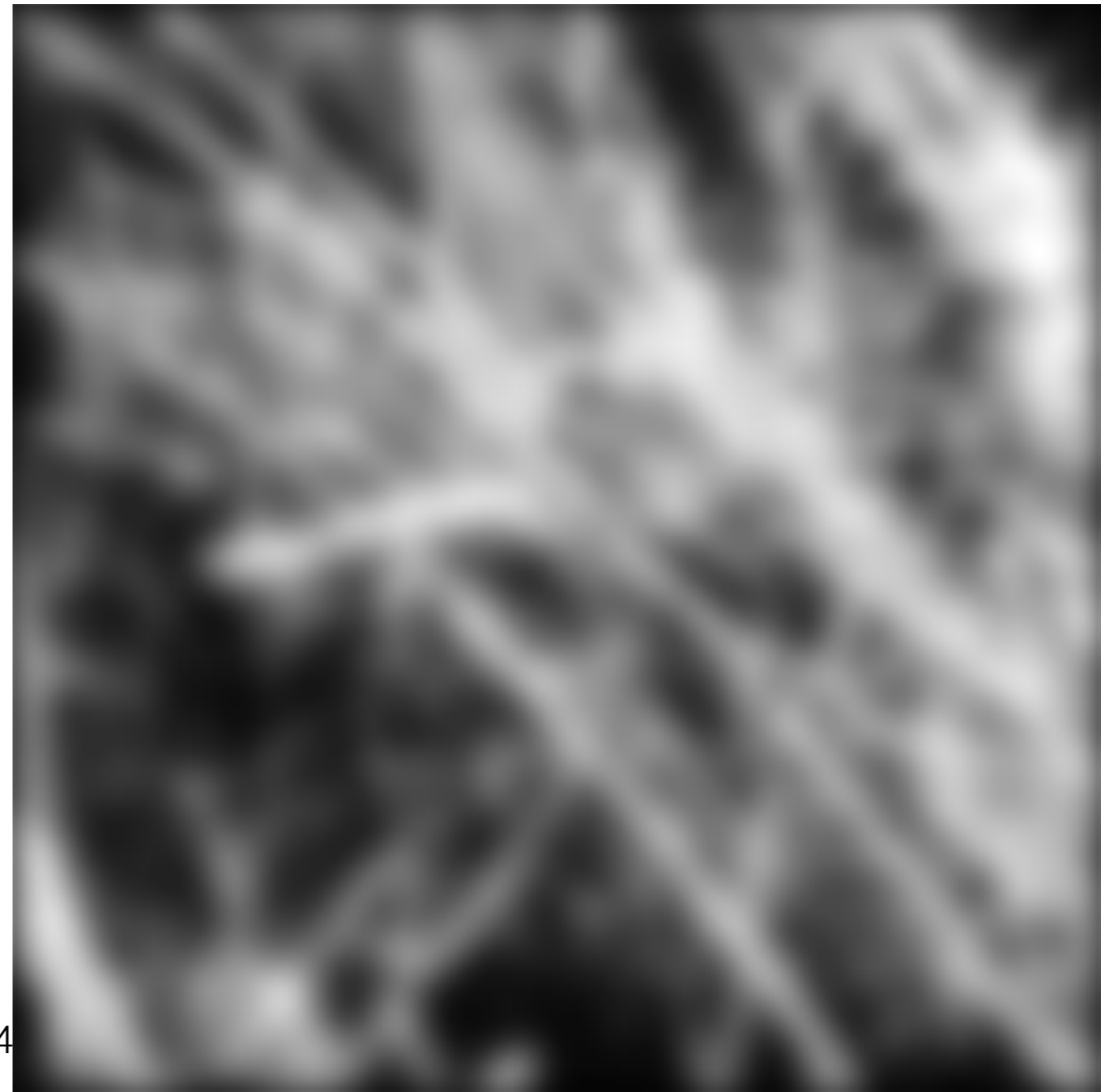
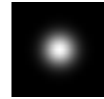


0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

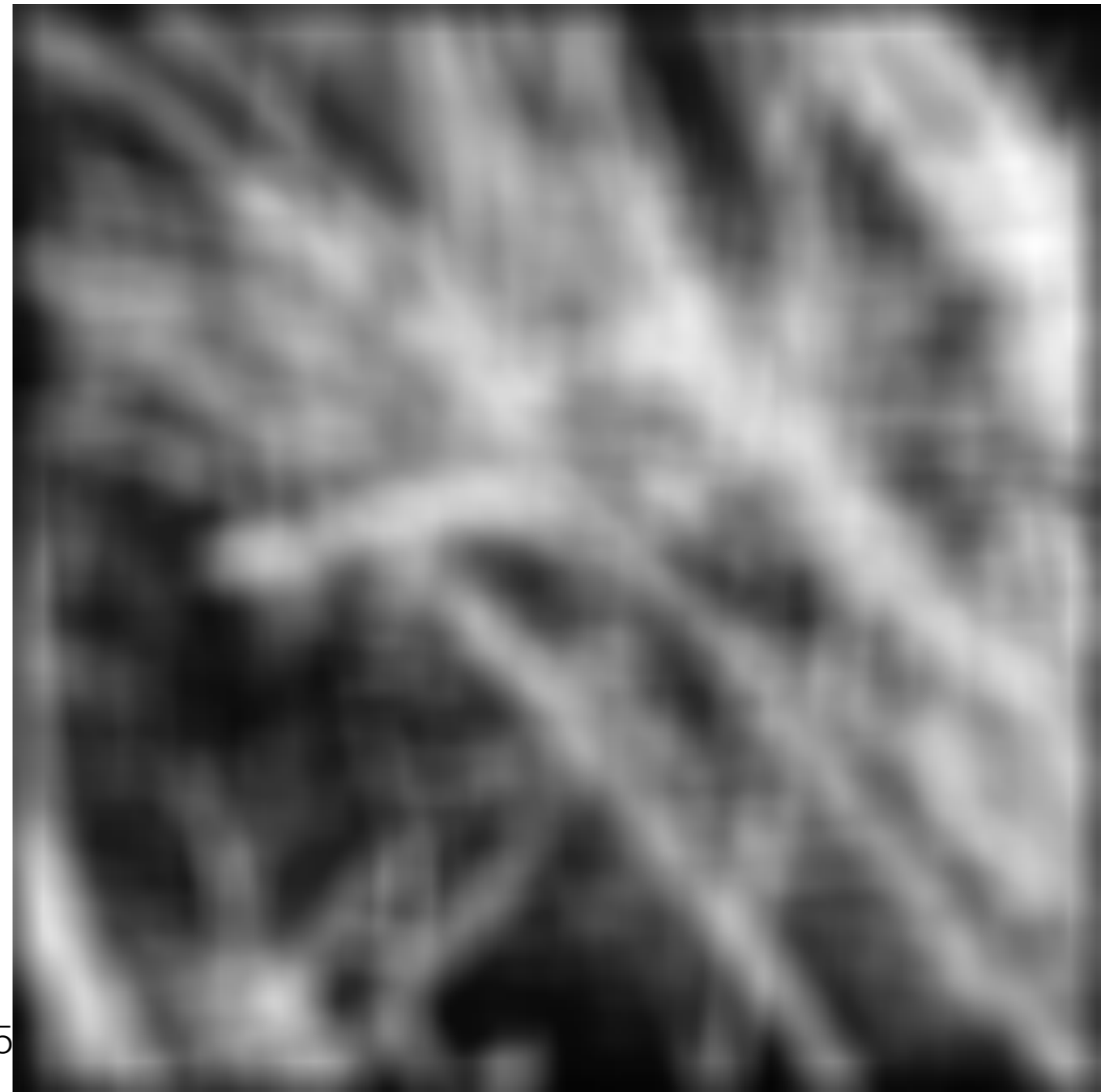
5 x 5, $\sigma = 1$

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

Atténuer avec le filtre gaussien



Atténuer avec le filtre “en boîte”



Filtres gaussiens

- Retire les hautes fréquences de l'image (filtre passe-bas)
 - Les images deviennent plus lisses
- Filter un filtre gaussien avec un autre filtre gaussien?
 - Le résultat est aussi gaussien
 - Si les deux filtres ont un écart-type de σ , c'est équivalent à un filtre d'écart-type $\sigma \cdot \sqrt{2}$
- Séparable
 - filtre gaussien 2D = produit de deux filtres gaussiens 1D

Séparabilité du filtre gaussien

$$\begin{aligned} G_{\sigma}(x, y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right)\right) \end{aligned}$$

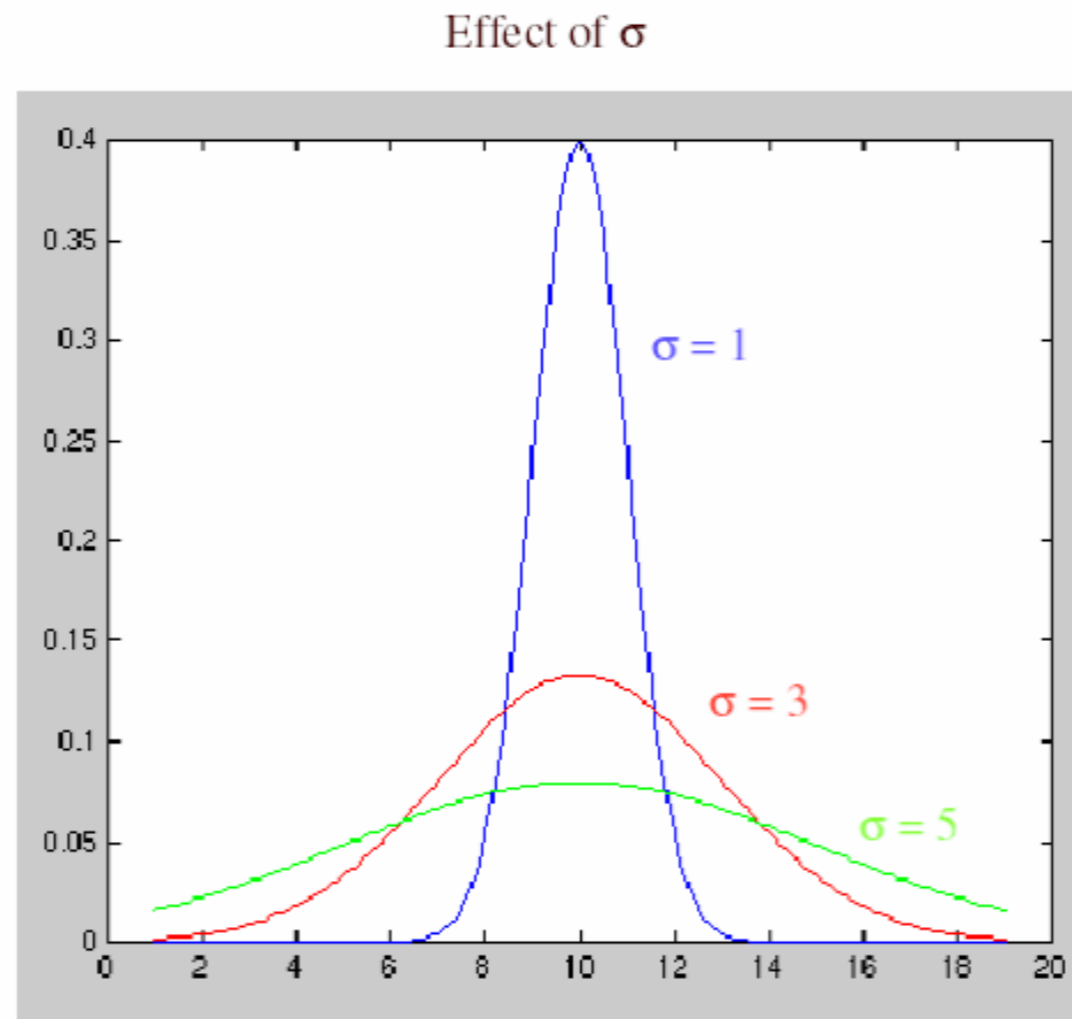
Pourquoi c'est important?

Démo

Filtre gaussien séparable

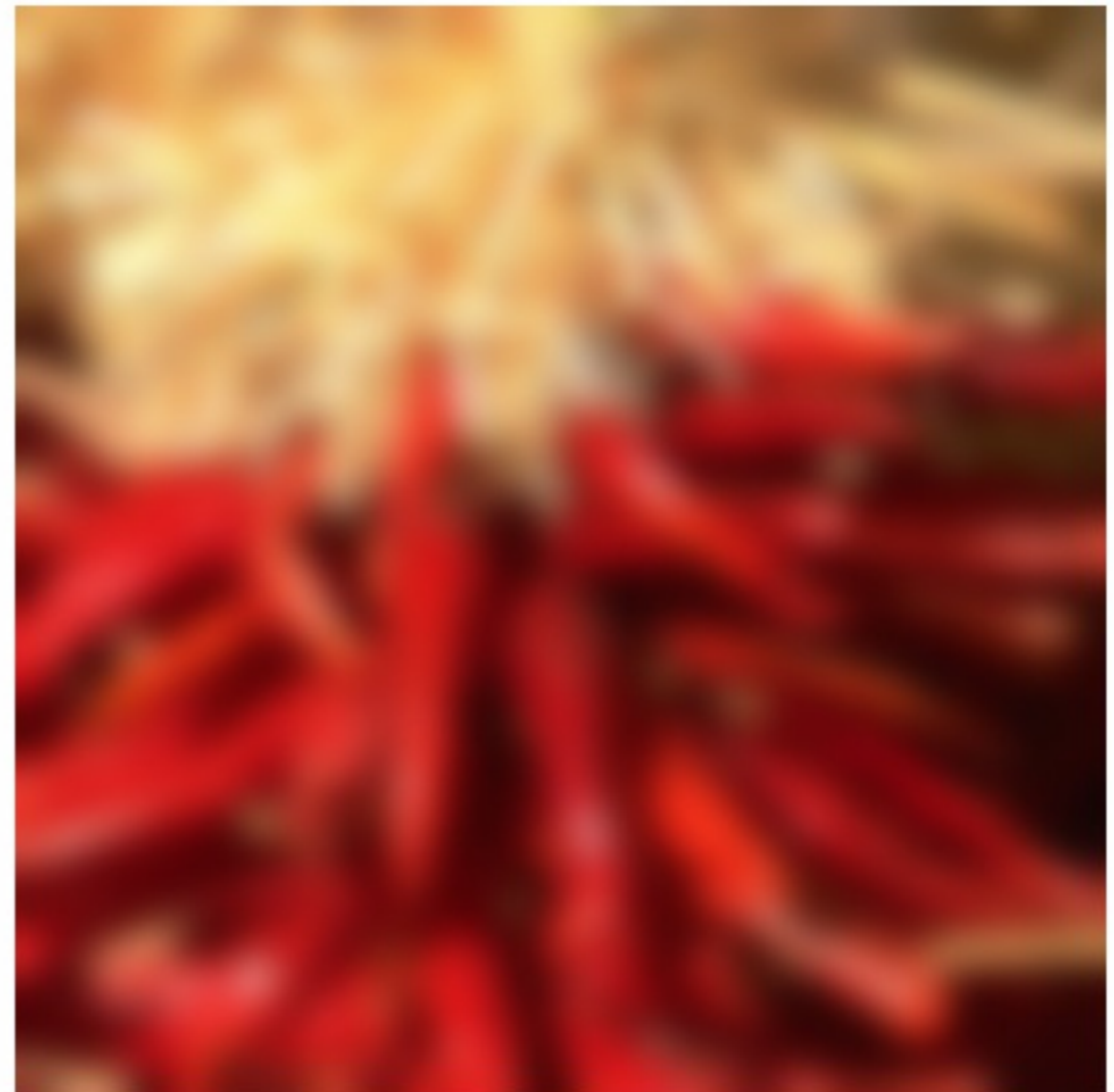
Considérations pratiques

- Quelle devrait être la taille du filtre?
- Les valeurs aux extrémités devraient être proche de 0
- Règle empirique: la demie-taille devrait être 3σ



Considérations pratiques

- Bordure de l'image?
 - le filtre dépasse l'image!
 - extrapolation
 - 0
 - “enrouler”
 - répéter
 - réflexion

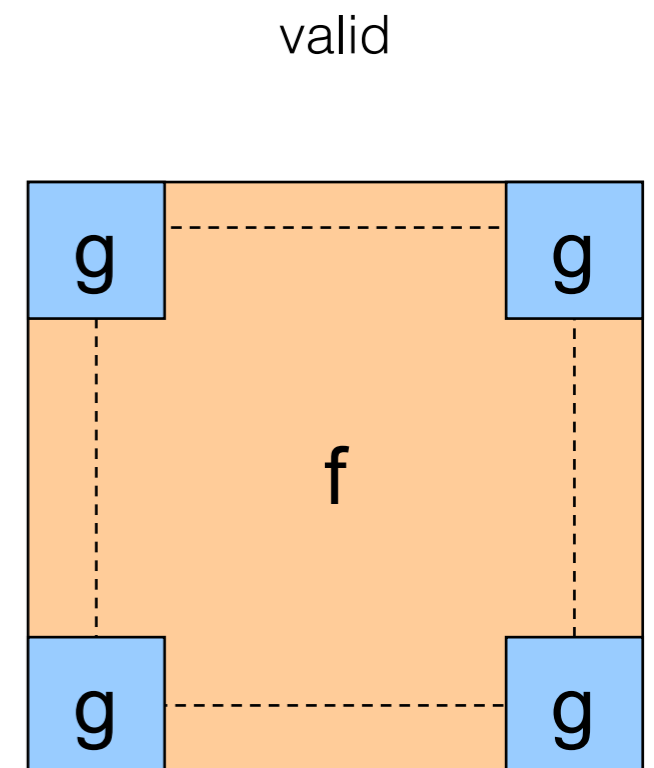
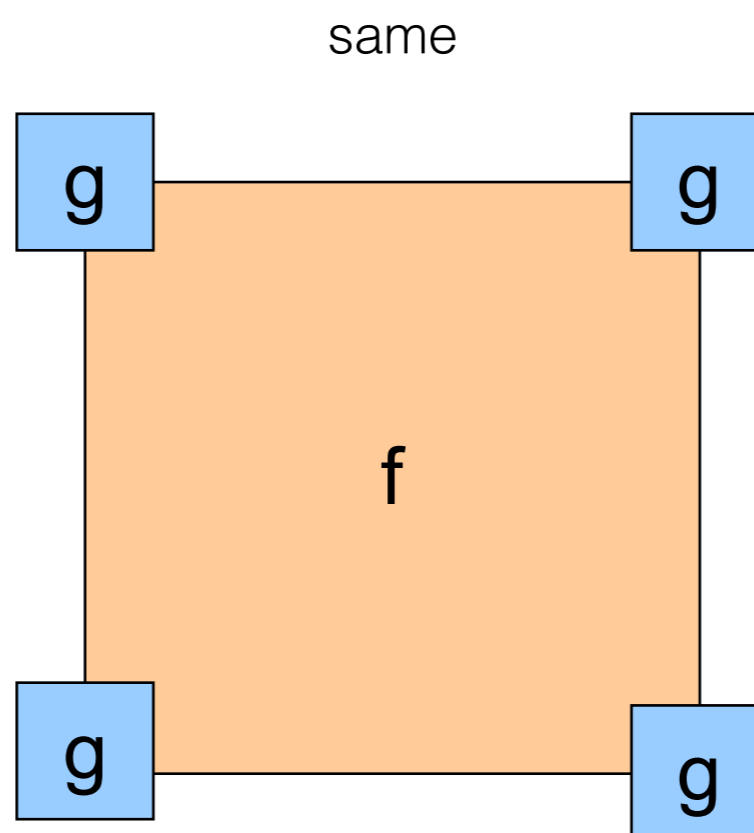
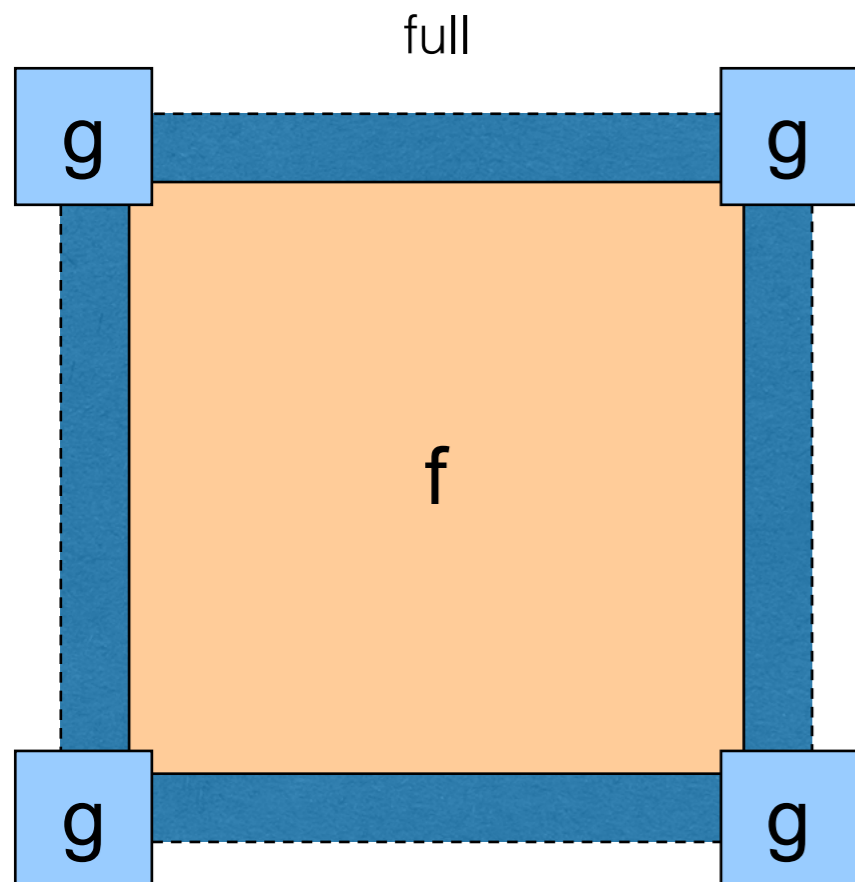


Considérations pratiques

- (MATLAB):
 - 0: `imfilter(f, g, 0)`
 - “enrouler”: `imfilter(f, g, 'circular')`
 - répéter: `imfilter(f, g, 'replicate')`
 - réflexion: `imfilter(f, g, 'symmetric')`

Considérations pratiques

- Quelle est la taille de l'image résultante?
- MATLAB: `filter2(g, f, shape)`
 - `shape = 'full'`: $\text{size}(f) + \text{size}(g)$
 - `shape = 'same'`: $\text{size}(f)$
 - `shape = 'valid'`: $\text{size}(f) - \text{size}(g)$



Questions de révision

- Un filtre 3x3 qui retourne un nombre positif si la valeur moyenne des 4-voisins est plus petite que celle du pixel central, et négatif sinon.
- Un filtre qui calcule le gradient dans la direction horizontale:
 - $\text{gradx}(y,x) = \text{im}(y,x+1) - \text{im}(y,x)$ pour tout x, y

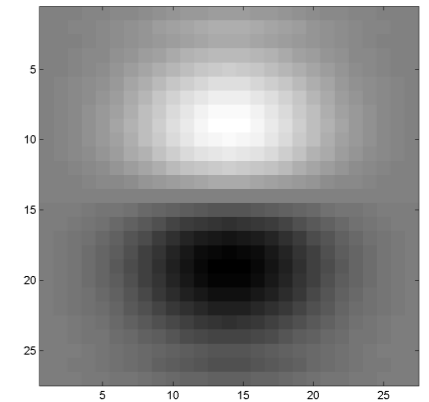
Questions de révision

Remplir les trous:

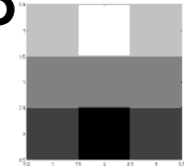
a) $_ = D * B$
b) $A = _ * _$
c) $F = \bar{D} * \bar{_}$
d) $_ = D * \bar{D}$

← Filtrage

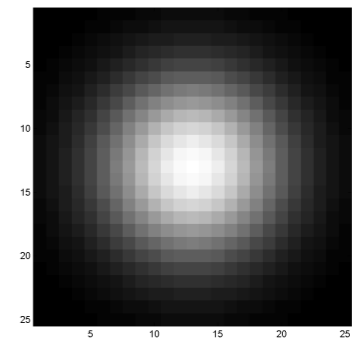
A



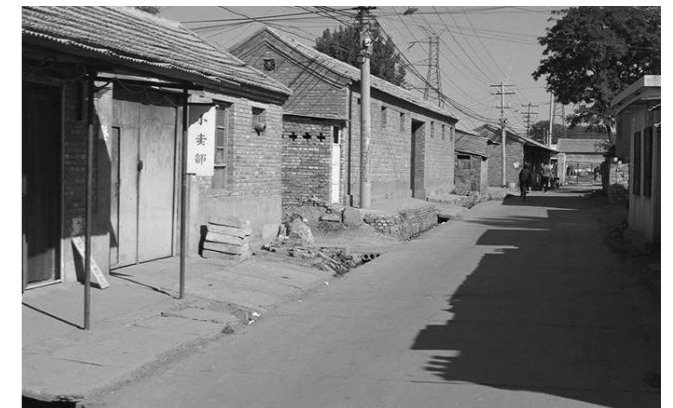
B



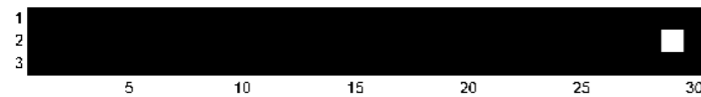
C



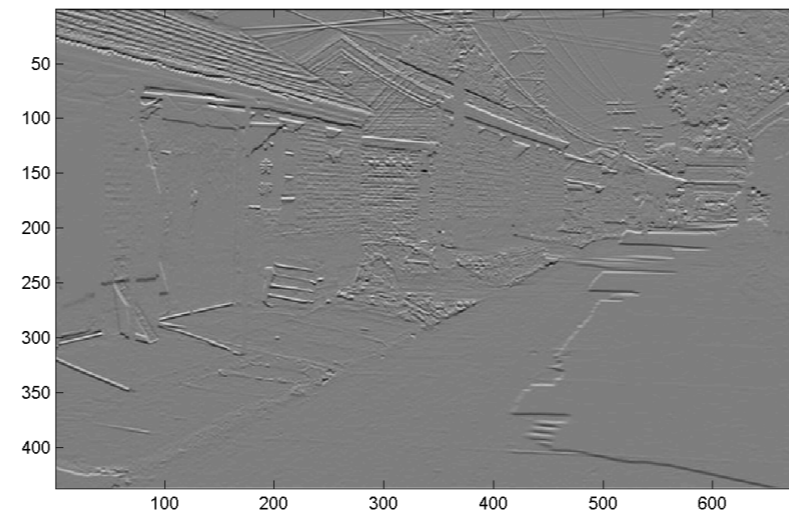
D



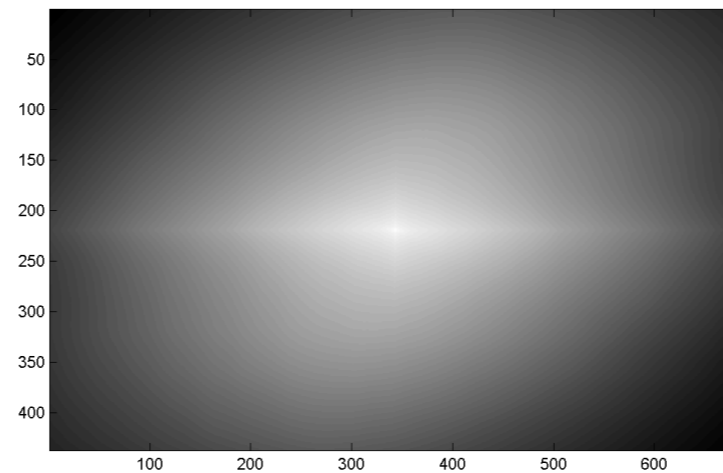
E



G



I



F



H

