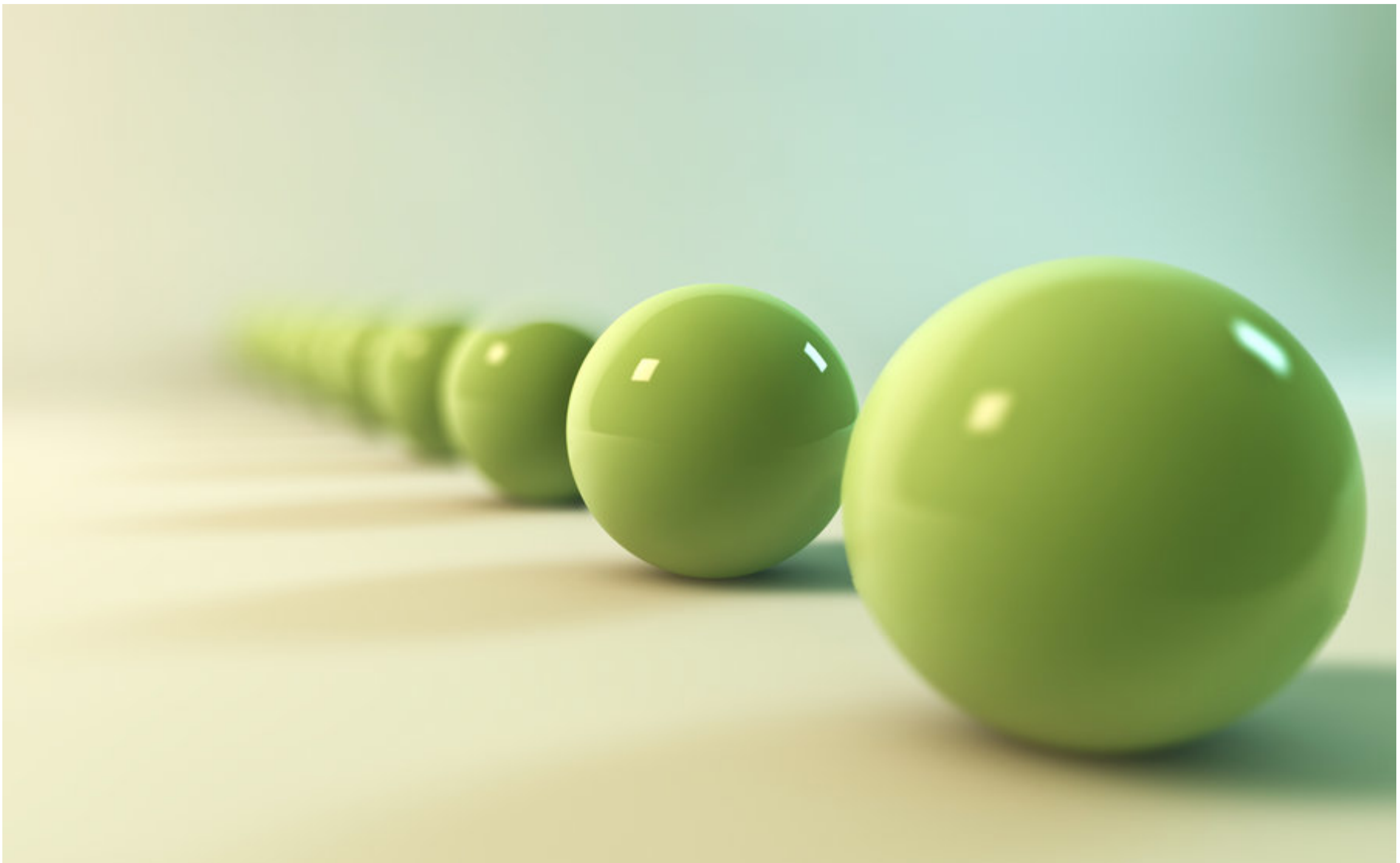
Filtrage dans le domaine spatial



Photographie Algorithmique, H2019

Jean-François Lalonde

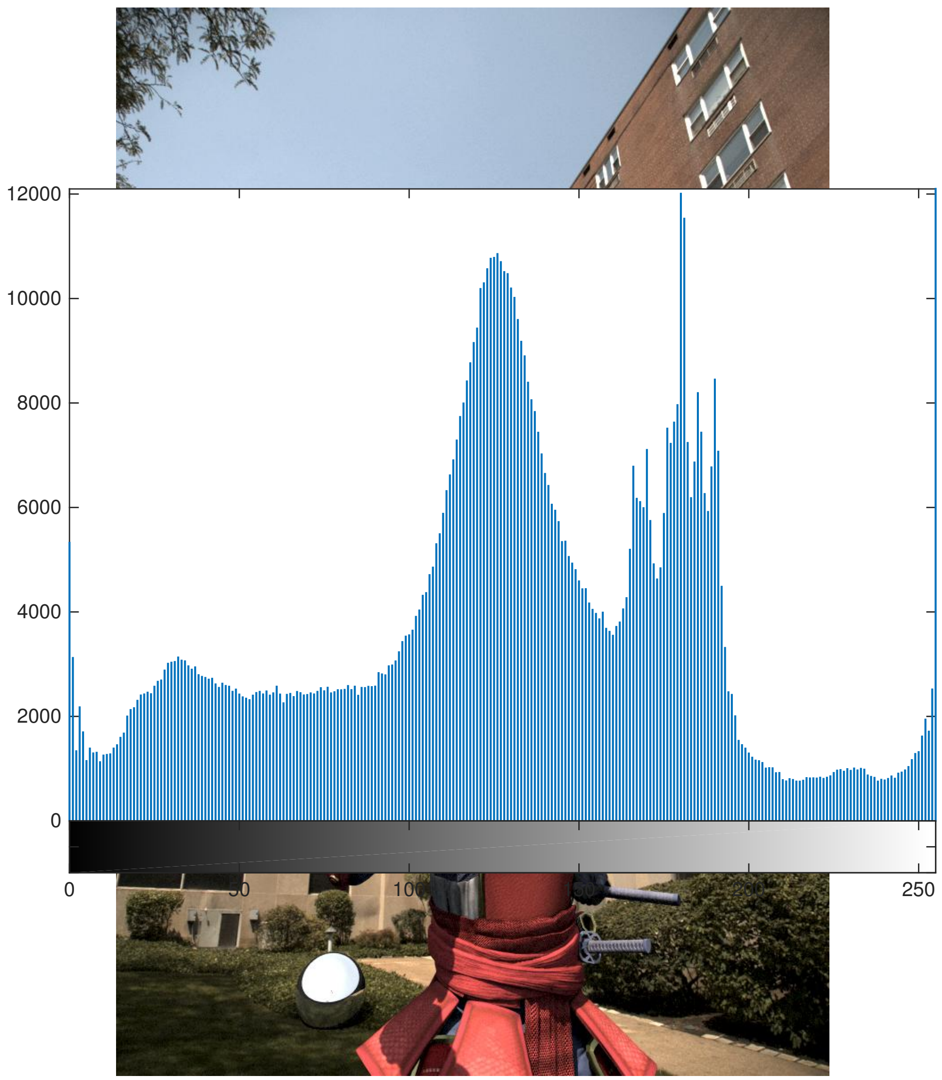
Image: “henriksn”, deviantart

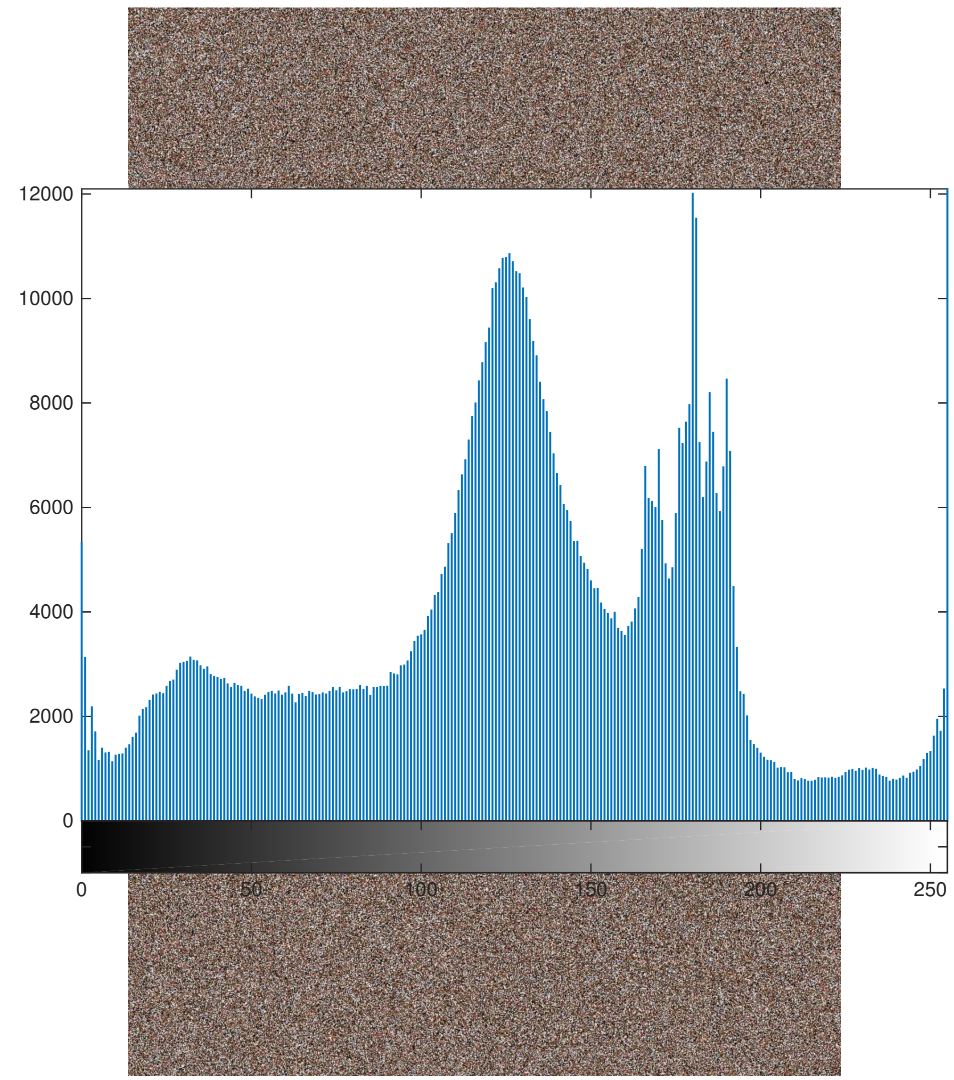
Limites des fonctions sur les pixels

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

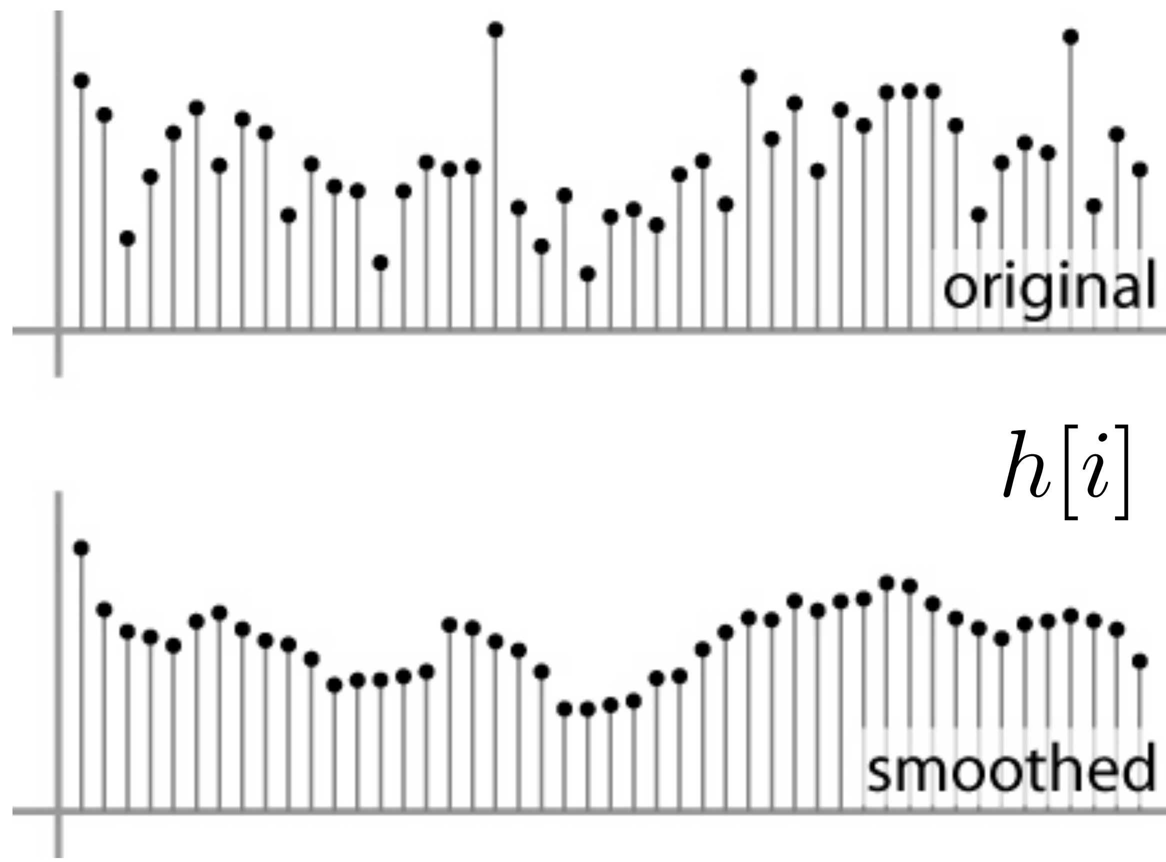
- Filtrage: fonction d’un pixel et de ses voisins
- Très important!
  - Modifier l’image
    - Réduire le bruit, re-dimensionner, contraste, etc.
  - Extraire de l’information
    - Texture, arêtes, points d’intérêt, etc.
  - Detecter des formes
    - «template matching»

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

$f[\cdot]$

$h[\cdot]$

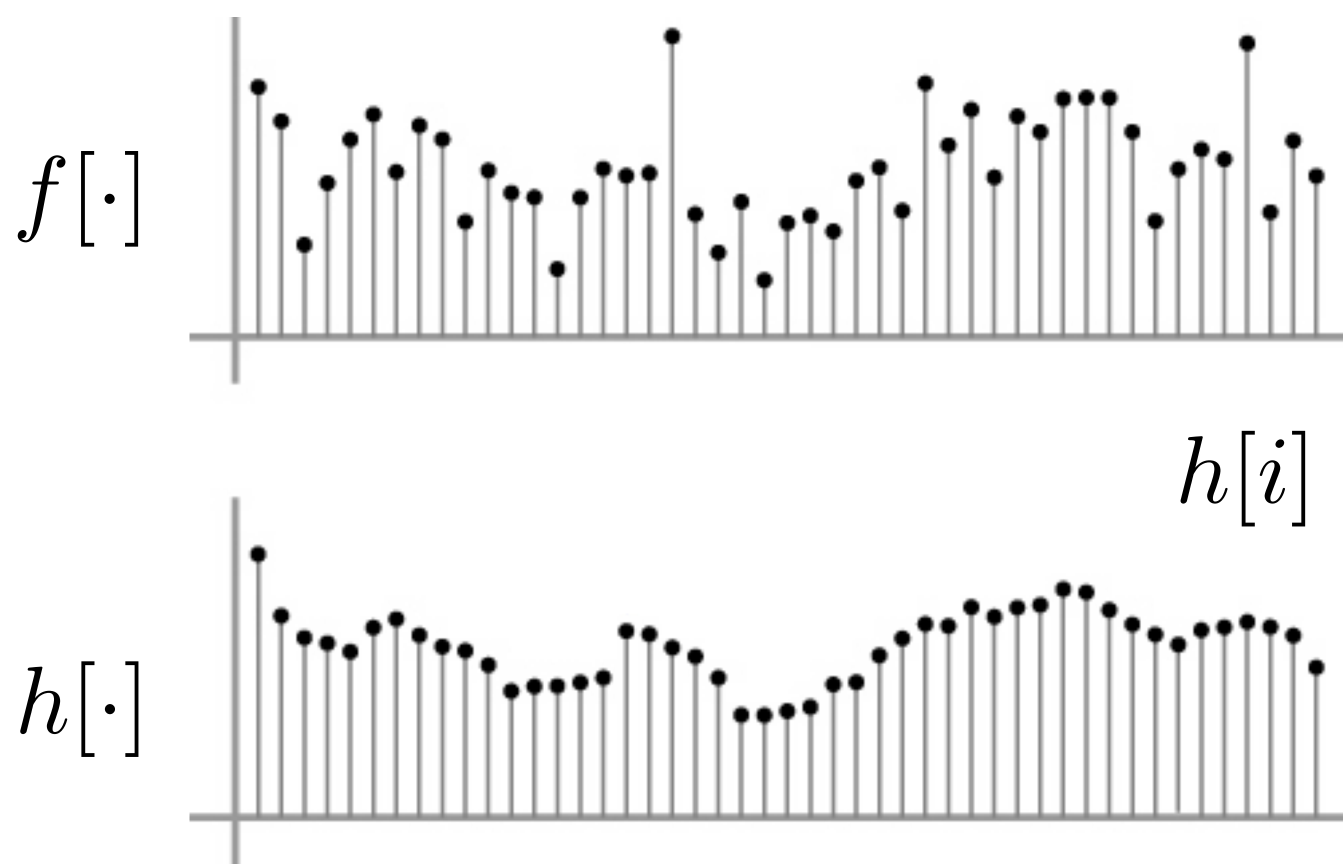


$$h[i] = \sum_{u=-k}^{k} g[u] f[i+u]$$

Moyenne temporelle (1D)

- Quel «filtre» *g* doit-on utiliser pour calculer la moyenne sur 5 points?

- *g* = [..., 0, 1, 1, 1, 1, 1, 0, ...] / 5

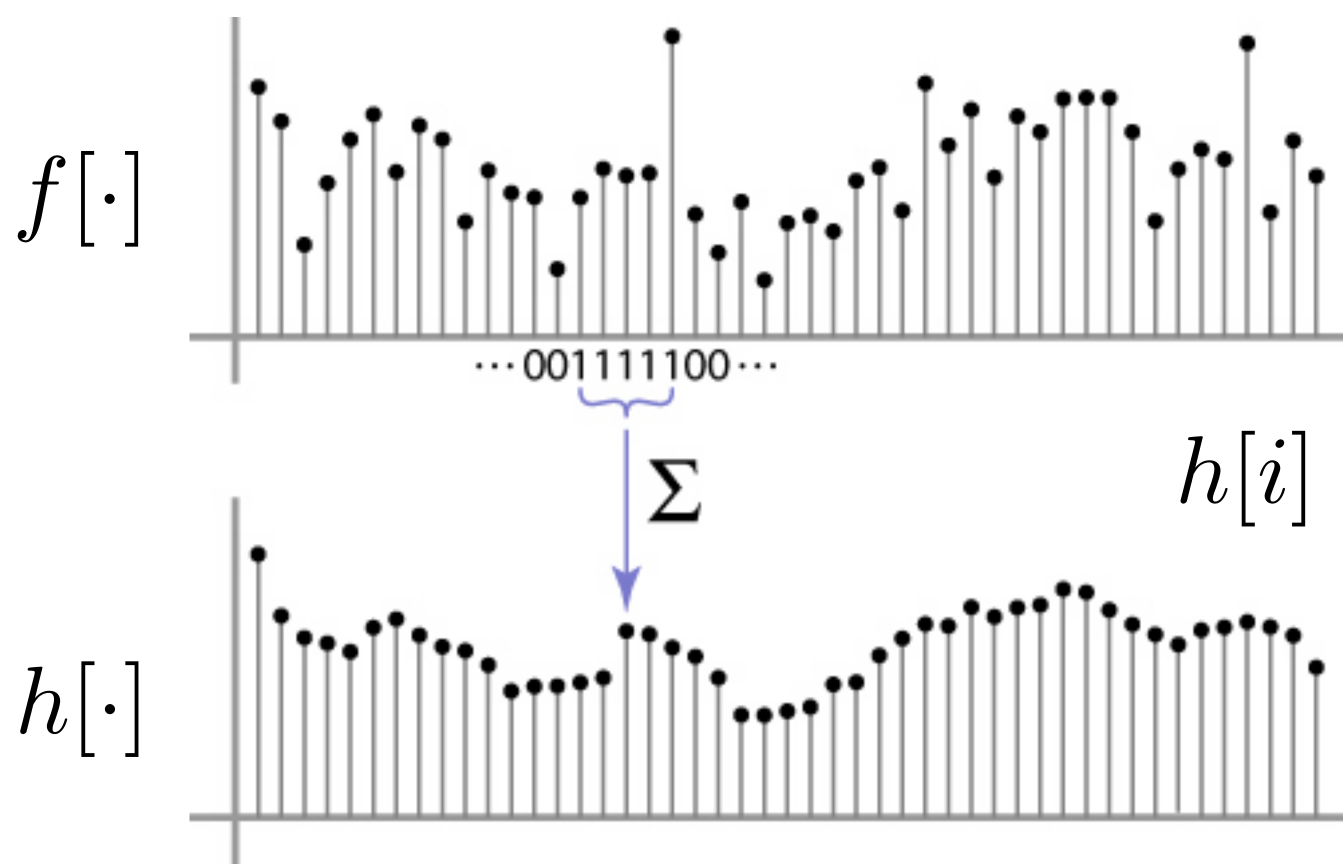


$$h[i] = \sum_{u=-k}^{k} g[u] f[i+u]$$

Moyenne temporelle (1D)

- Quel «filtre» *g* doit-on utiliser pour calculer la moyenne sur 5 points?

- *g* = [..., 0, 1, 1, 1, 1, 1, 0, ...] / 5



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

Corrélation croisée (en 2D)

- Nous avons:
  - une image *f*
  - un filtre *g*, de taille (2*k*+1) x (2*k*+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

$$\mathbf{g}[\cdot\,,\cdot\,]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Source: D. Lowe

Filtrage

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

$f[\cdot,\cdot]$

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

$h[\cdot,\cdot]$

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

Source: S. Seitz

Filtrage

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

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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							

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

Source: S. Seitz

Filtrage

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

$f[\cdot,\cdot]$

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						

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

Source: S. Seitz

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					

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

Source: S. Seitz

Filtrage

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

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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

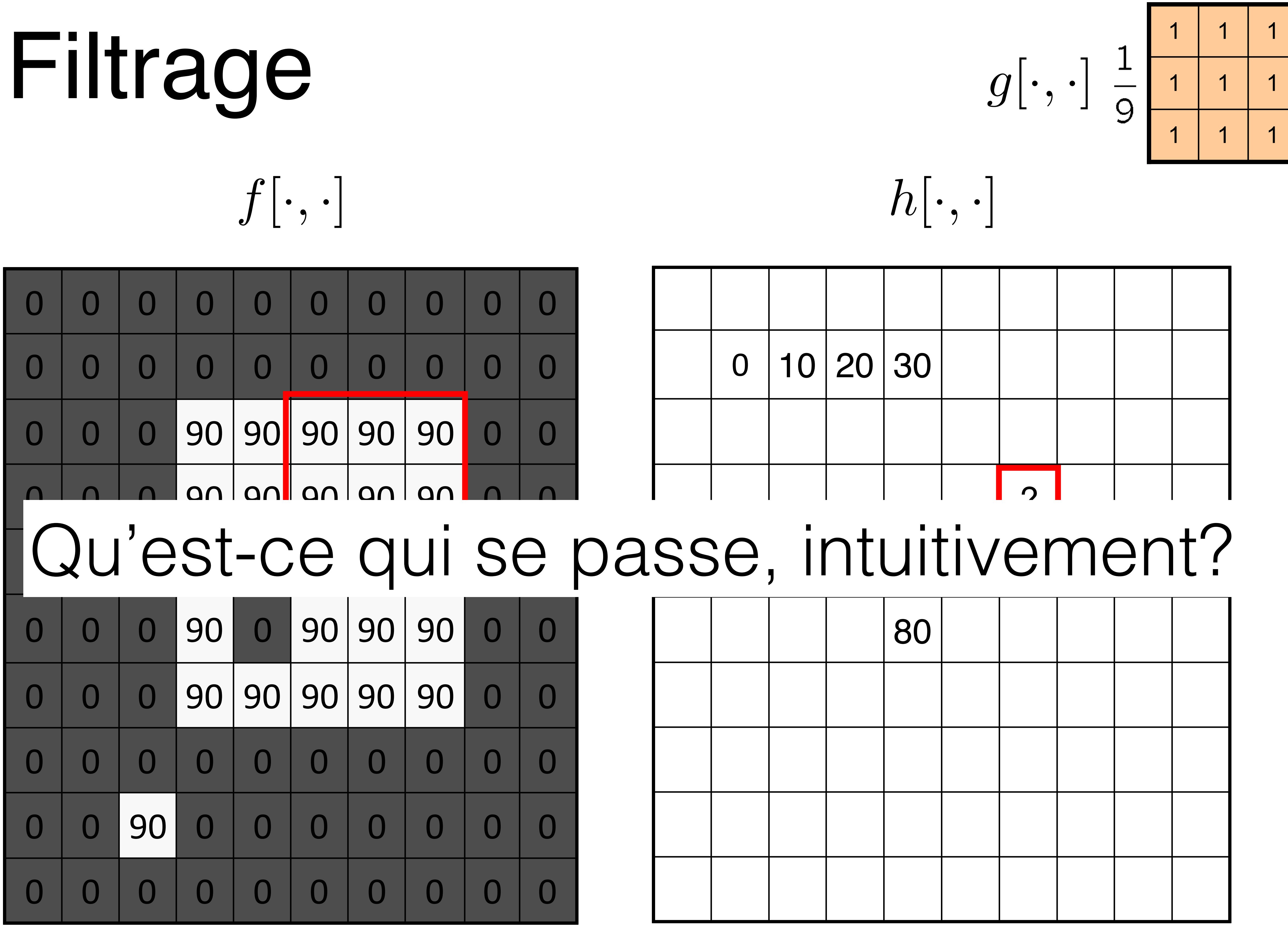
$h[\cdot,\cdot]$

	0	10	20	30					
				?					

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

Source: S. Seitz

Filtrage



Qu’est-ce qui se passe, intuitivement?

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

Source: S. Seitz

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	

Source: S. Seitz

Filtrage “en boîte” (box filter)

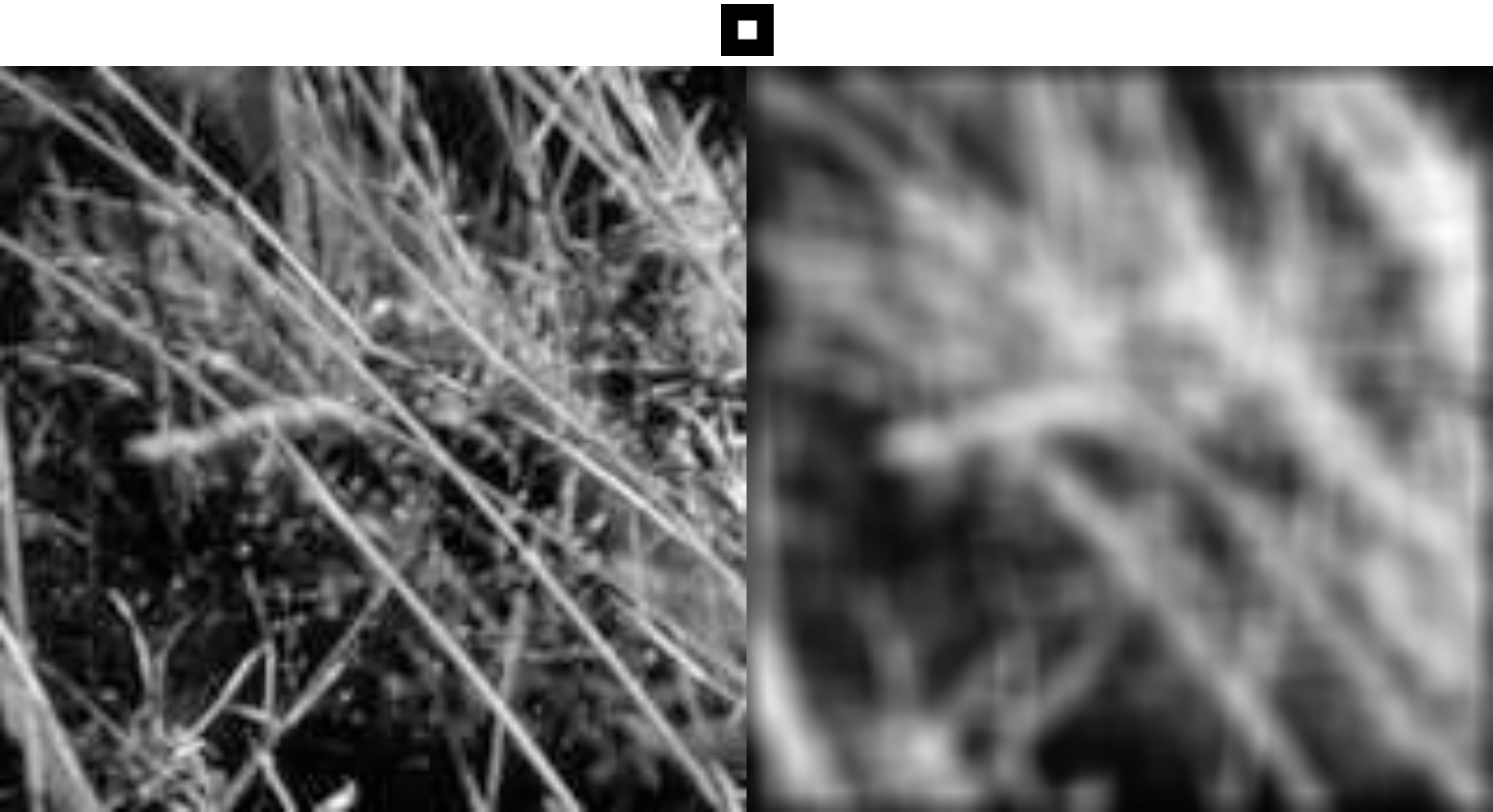
- Remplace chaque pixel par la moyenne de son voisinage
- On adoucit l’image (enlève les hautes fréquences)

$$g[\cdot,\cdot]$$

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

Source: D. Lowe

“Atténuer” l’image avec le filtre boîte



Petite pratique avec les filtres linéaires

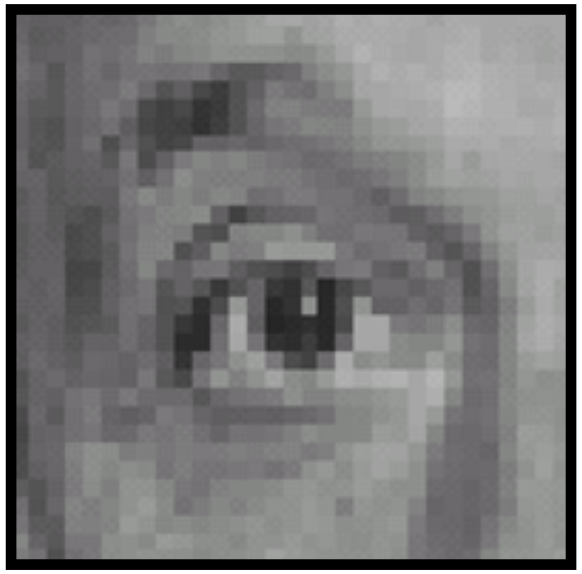


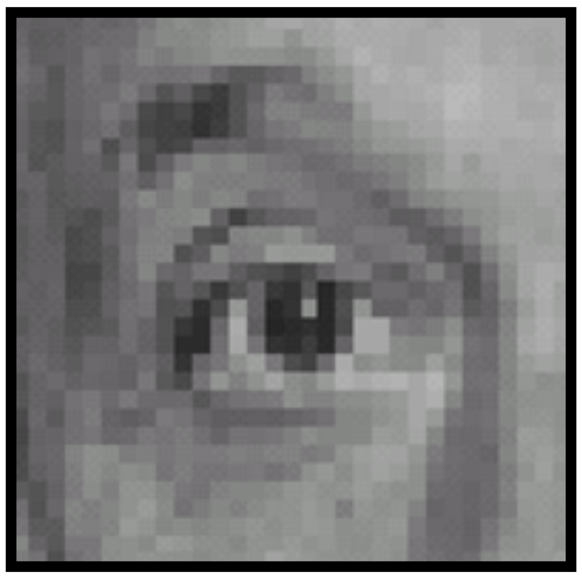
Image originale

0	0	0
0	1	0
0	0	0

?

Source: D. Lowe

Petite pratique avec les filtres linéaires



0	0	0
0	1	0
0	0	0

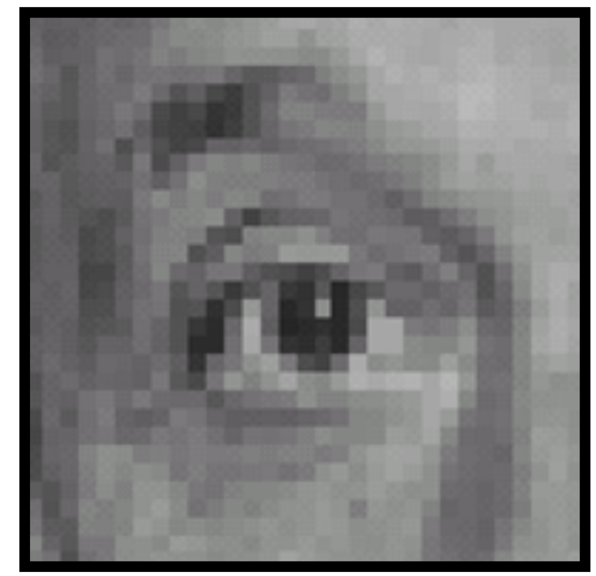


Image originale

Résultat
(identique!)

Source: D. Lowe

Petite pratique avec les filtres linéaires

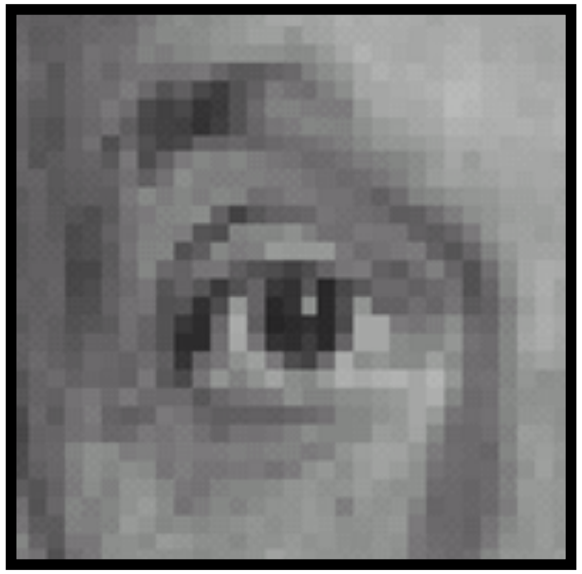


Image originale

0	0	0
0	0	1
0	0	0

?

Source: D. Lowe

Petite pratique avec les filtres linéaires

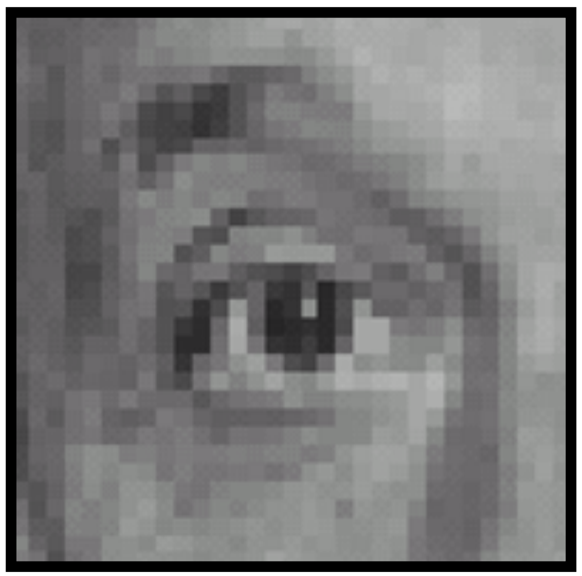
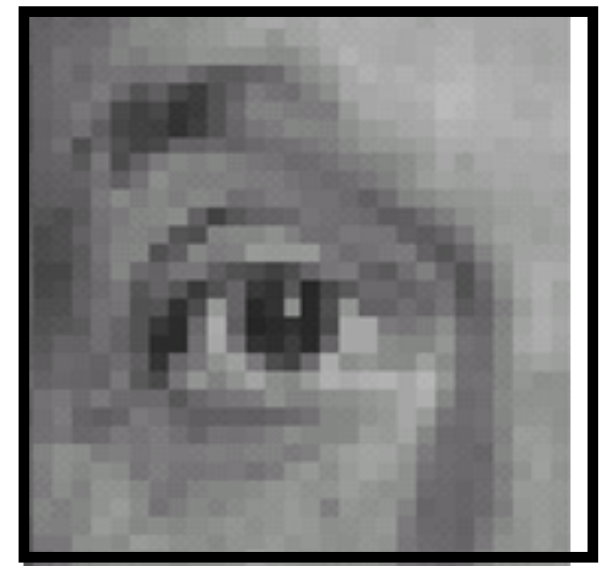


Image originale

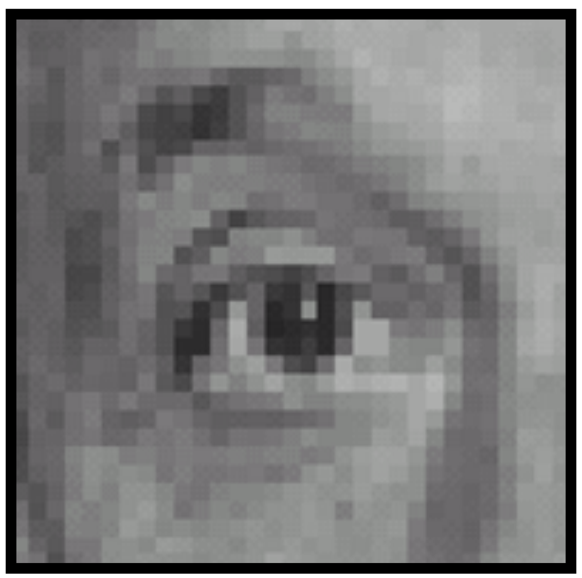
0	0	0
0	0	1
0	0	0



À gauche de 1 pixel

Source: D. Lowe

Petite pratique avec les filtres linéaires



$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \; - \; \frac{1}{9} \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad ?$$

Image originale

Source: D. Lowe

Petite pratique avec les filtres linéaires

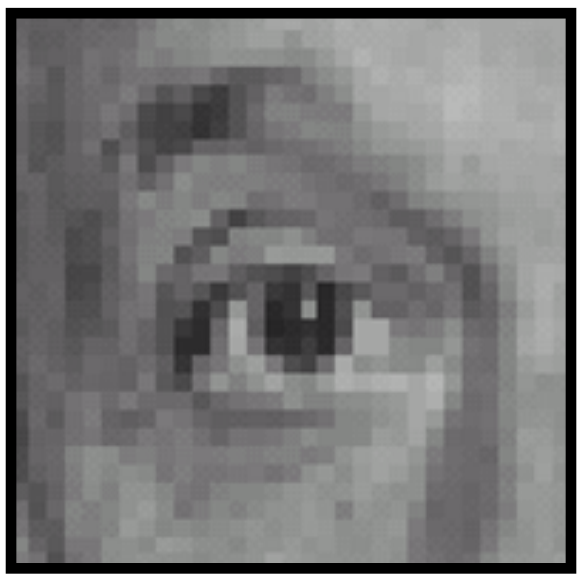


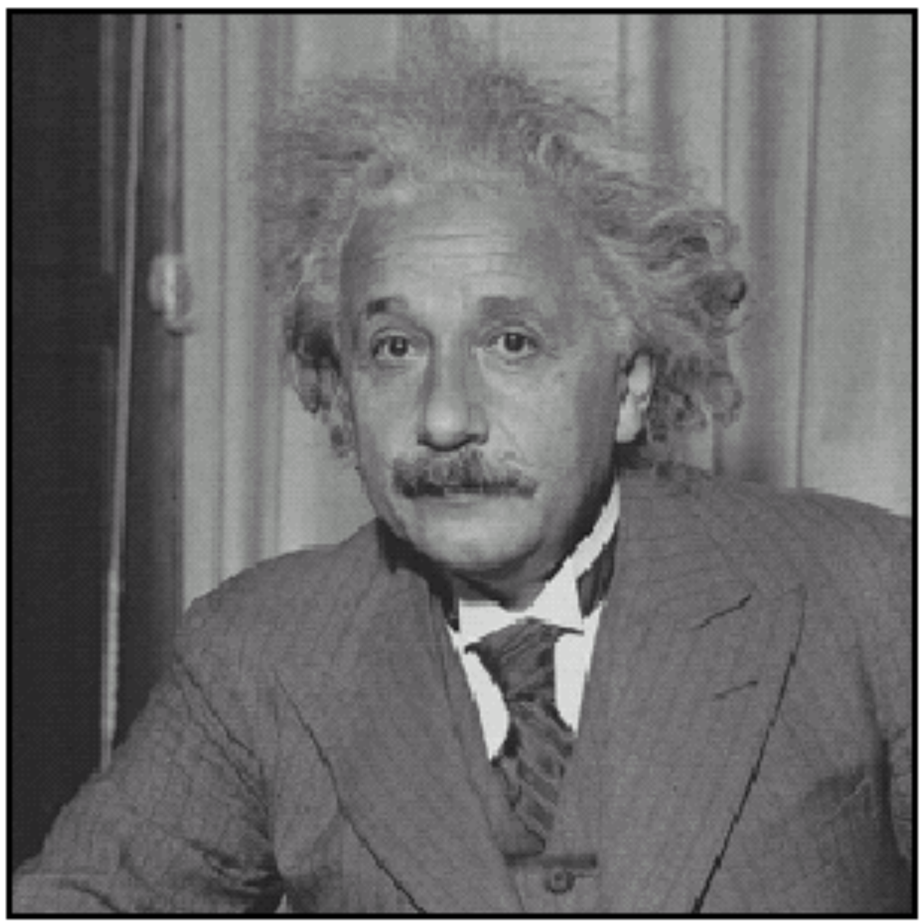
Image originale

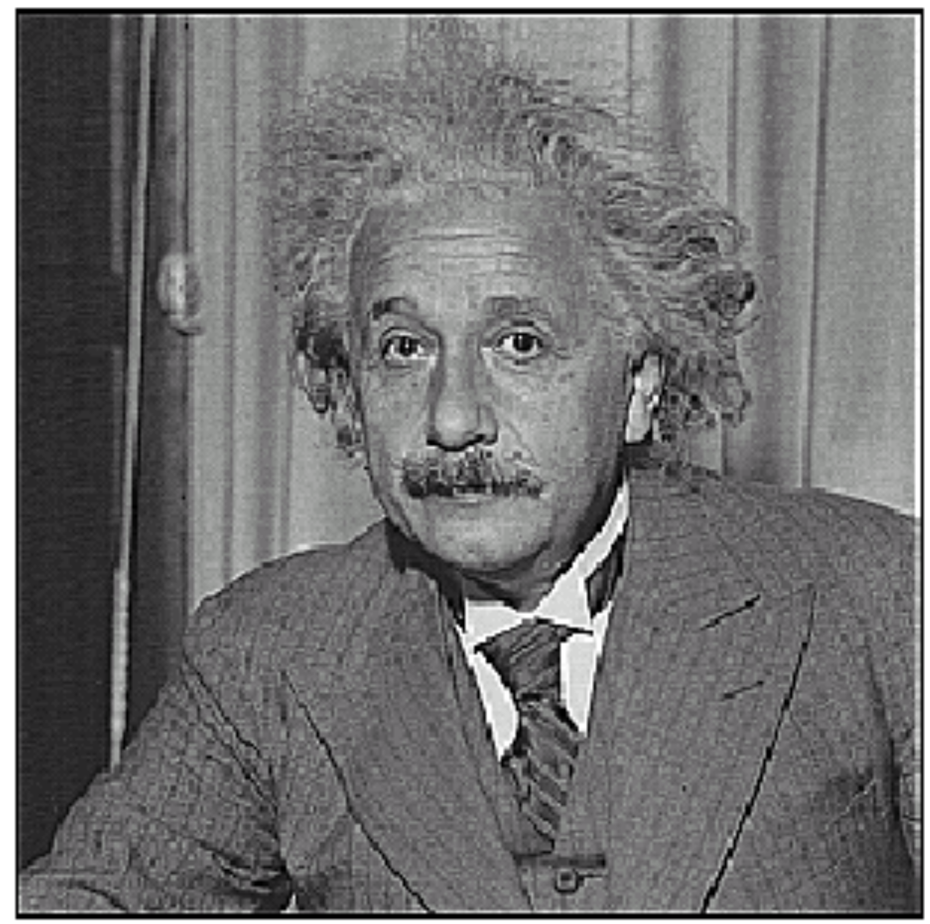
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Accentue les différences par rapport à la moyenne

Source: D. Lowe

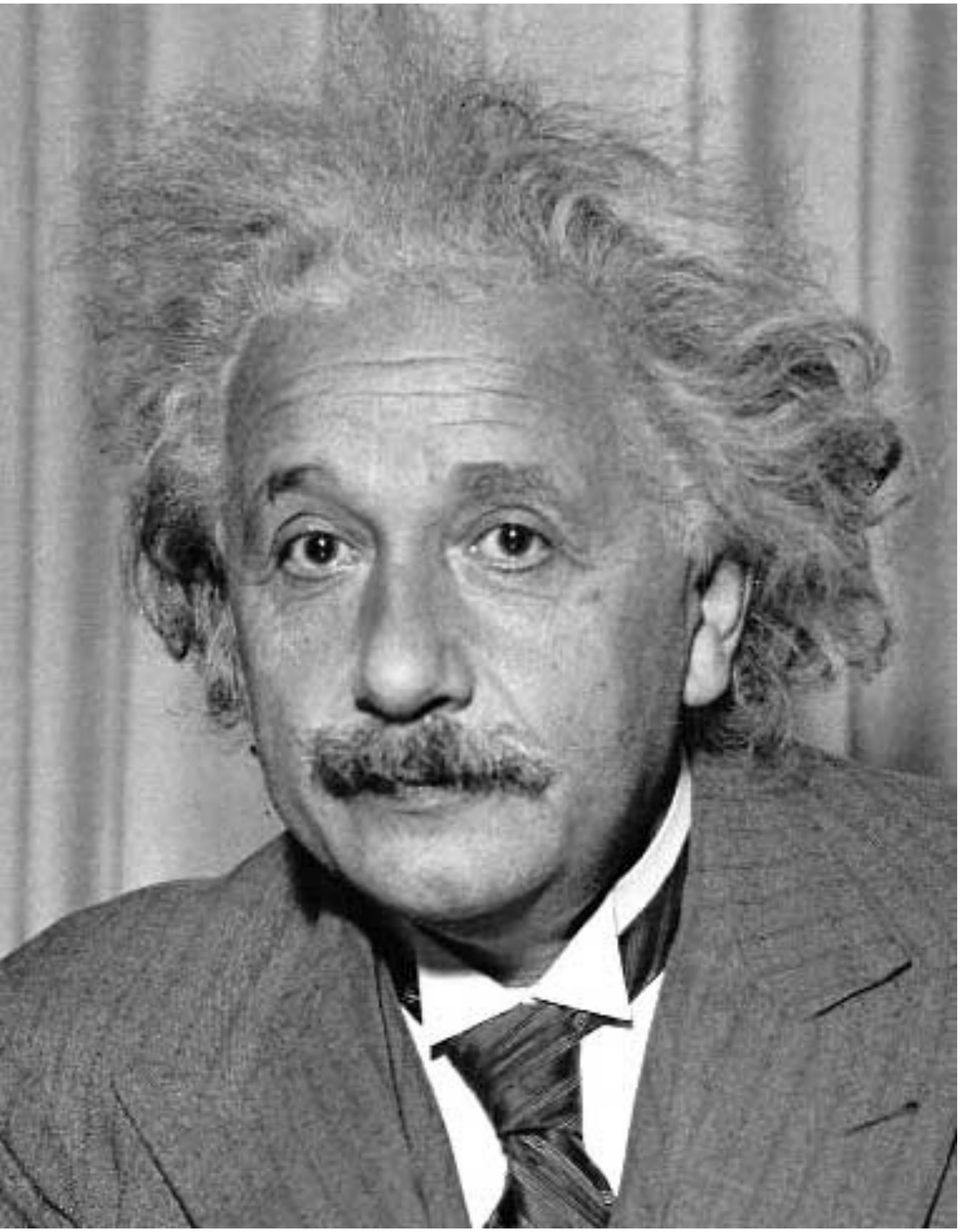
Accentuation “sharpening”





Source: D. Lowe

Autres filtres

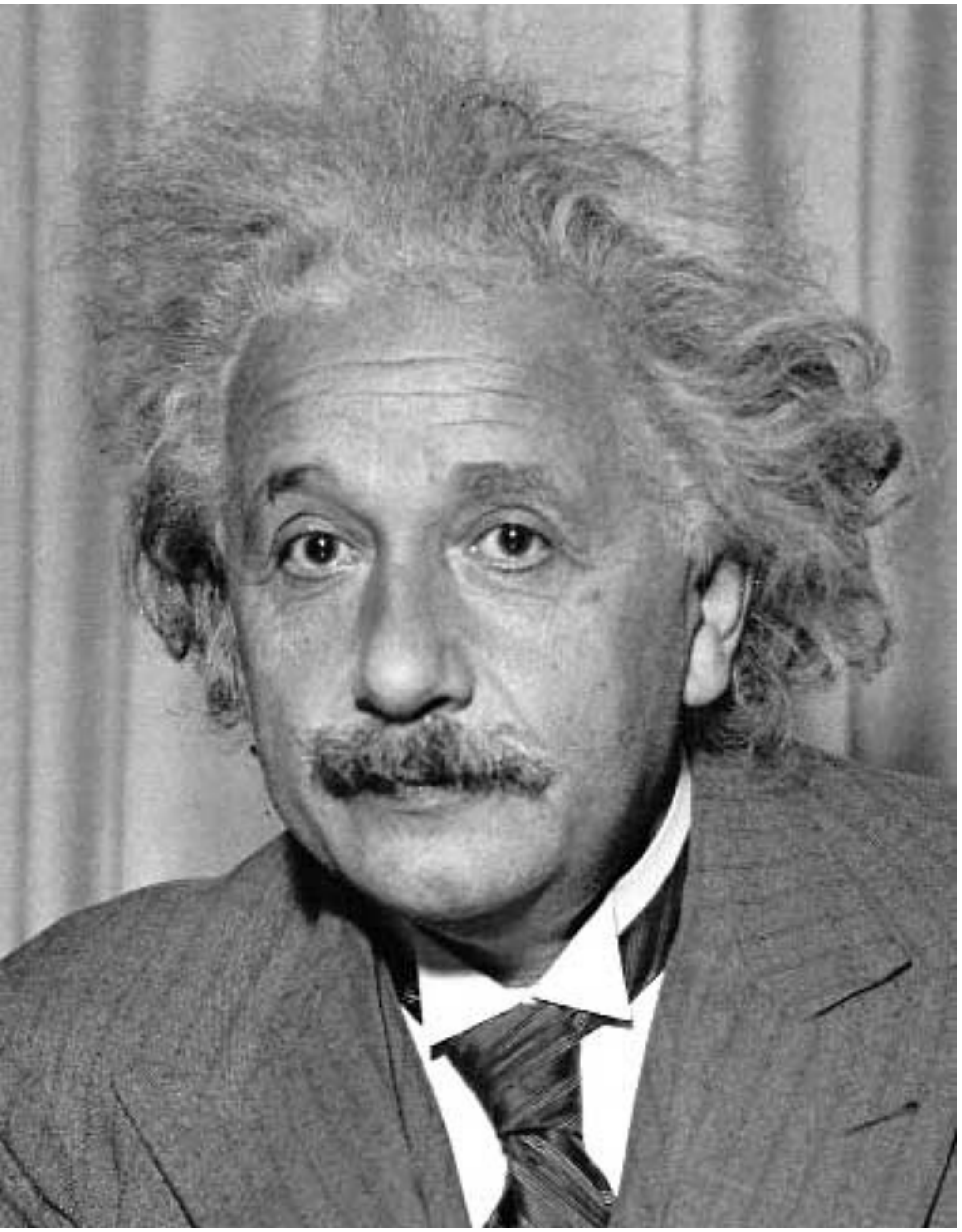


1	0	-1
2	0	-2
1	0	-1

Sobel

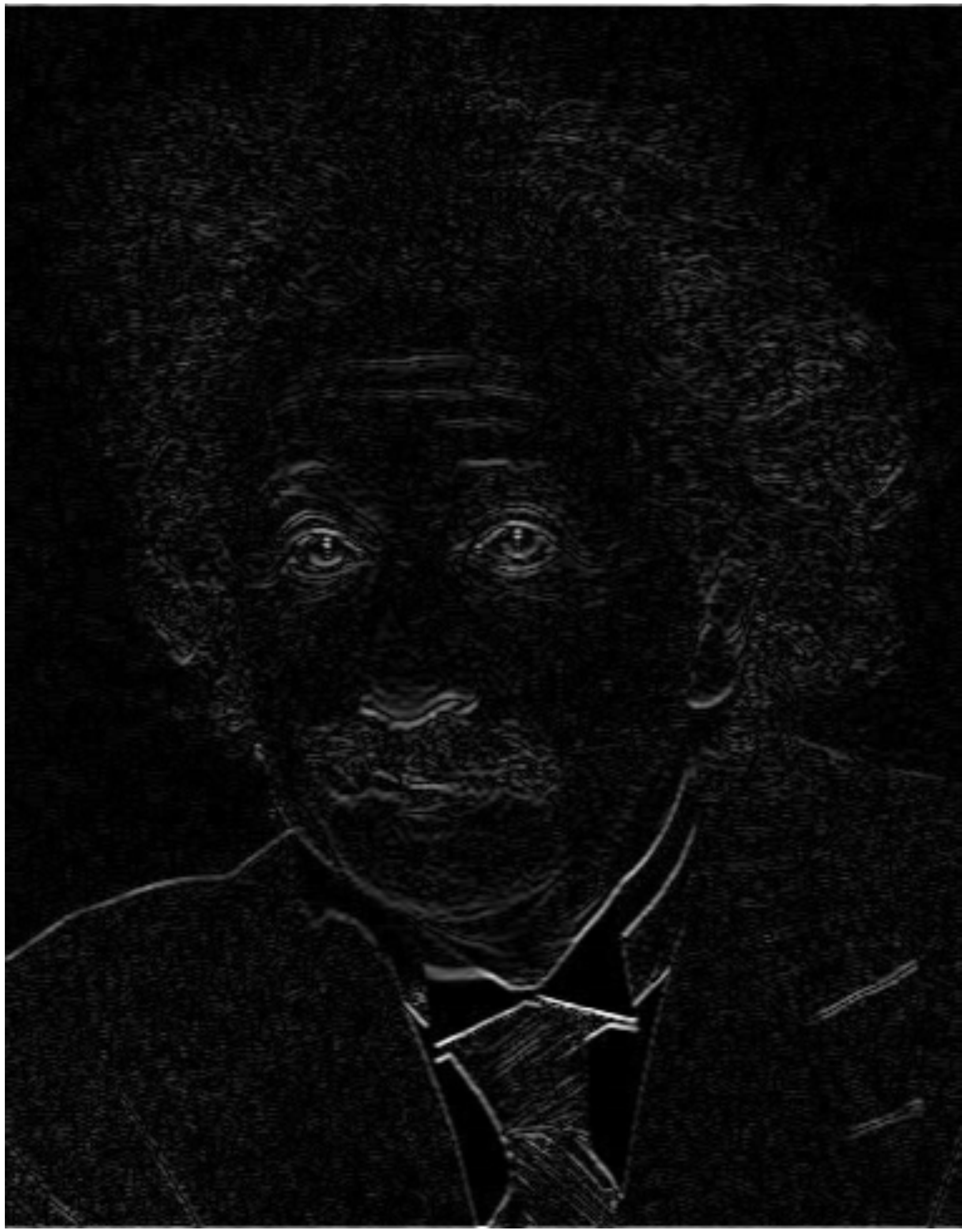
Arêtes verticales
(valeur absolue)

Autres filtres



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

Sobel



Arêtes horizontales
(valeur absolue)

Démo

Filtrage

$$G = H \otimes F$$

Corrélation vs. convolution

- La **corrélation 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:

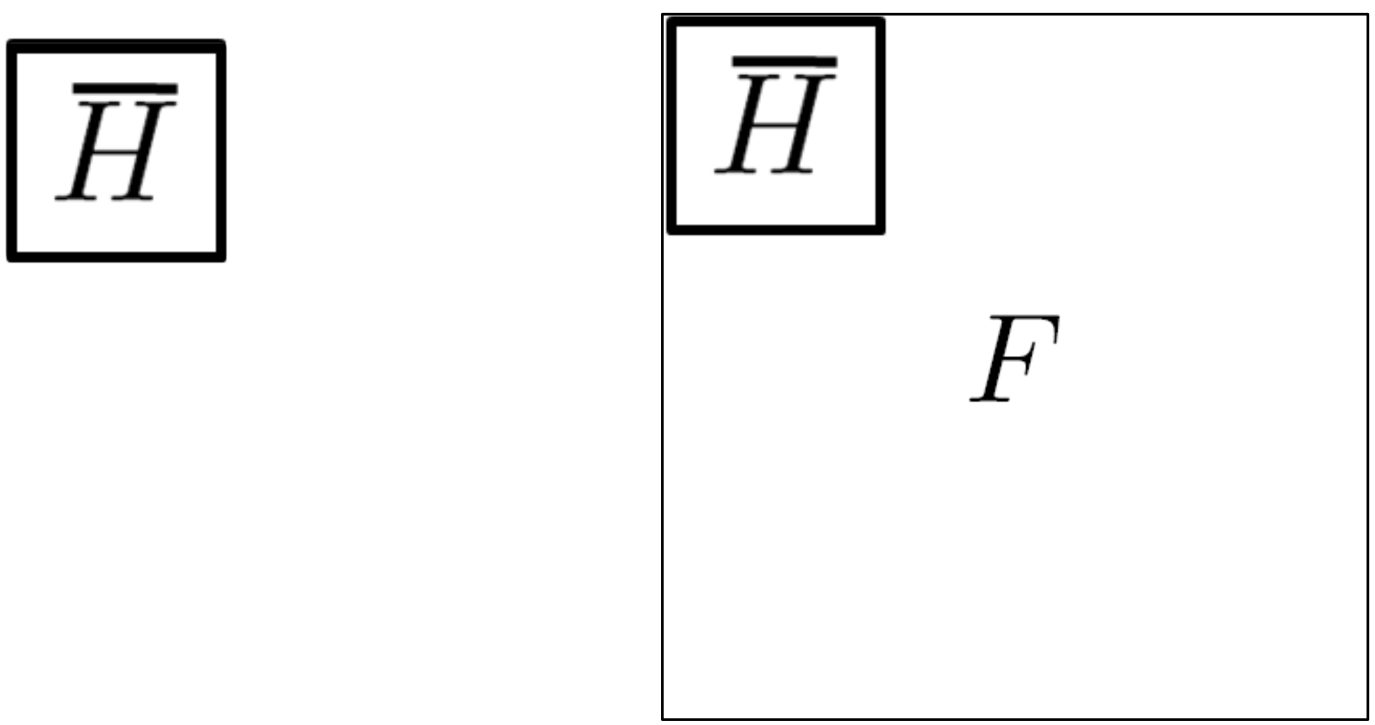
$$h[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} g[u,v] f[i-u, j-v]$$

- On l'écrit:

$$h = g * f$$

Steve Seitz

Convolution



de F. Durand

Propriété des filtres linéaires

- Linéarité (!):
  - filtre(f1 + f2) = filtre(f1) + filtre(f2)
- Invariance aux déplacement: ne dépend pas de l'emplacement du pixel
  - filtre(shift(f)) = shift(filtre(f))

Source: S. Lazebnik

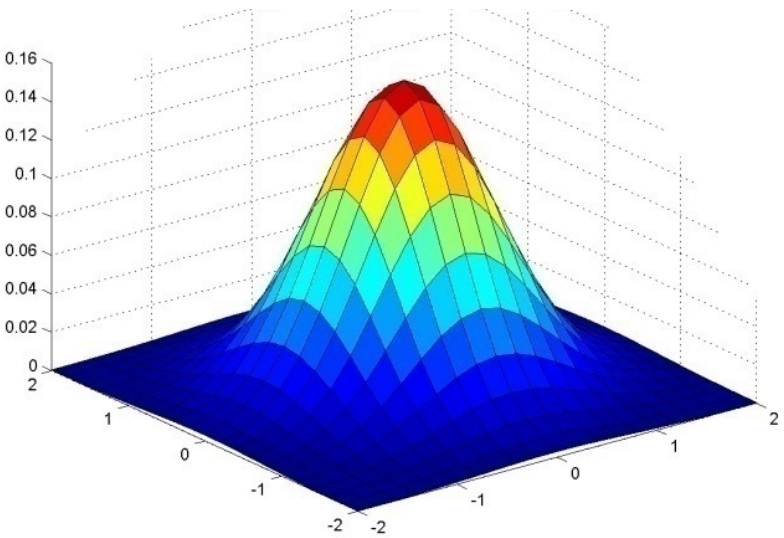
Propriétés des filtres linéaires

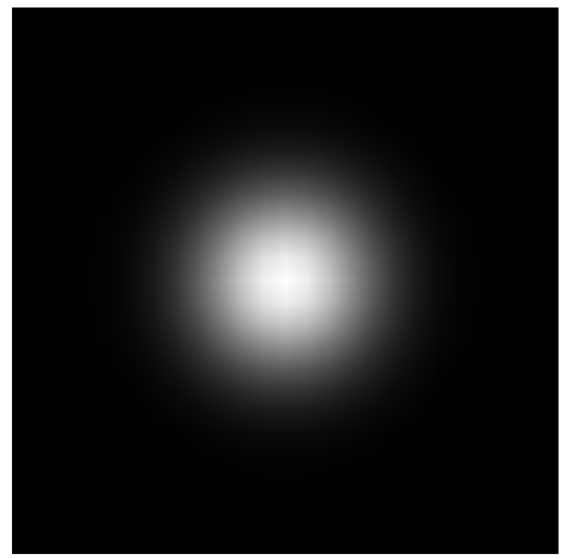
- Commutatif: a * b = b * a
  - On peut donc traiter l'image comme un filtre
- Associatif: a * (b * c) = (a * b) * c
  - Souvent, on applique plusieurs filtres: (((a * b1) * b2) * b3)
  - Équivalent à filtrer l'image avec le “filtre du filtre”: a * (b1 * b2 * b3)
- Distribution: a * (b + c) = (a * b) + (a * c)
- Factoriser les scalaires: ka * b = a * kb = k (a * b)
- Identité e = [0, 0, 1, 0, 0], a * e = a

Source: S. Lazebnik

Filtre important: gaussien

- Pondère les contributions des voisins en fonction de leur distance





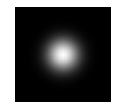
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

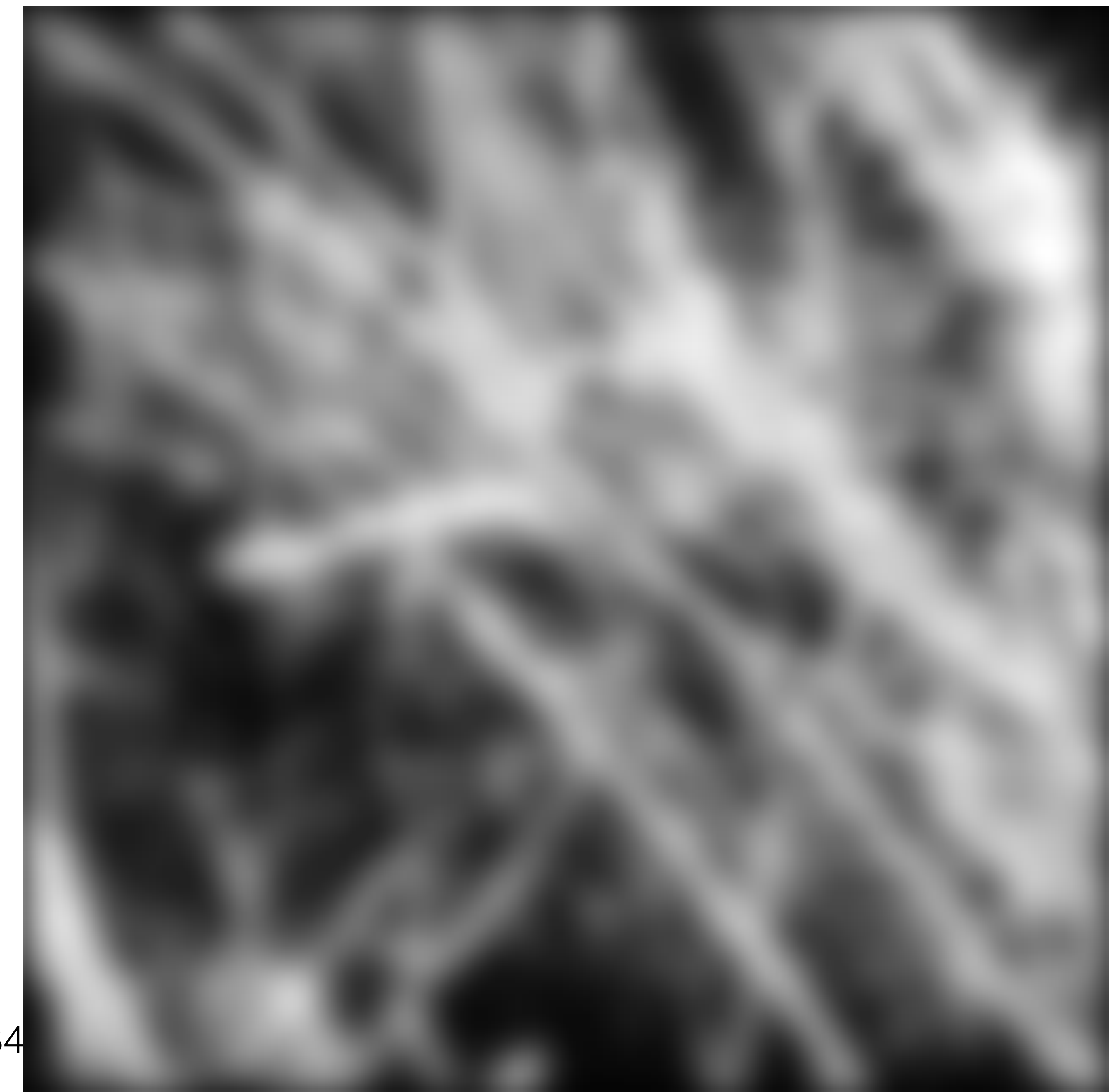
5 x 5, σ = 1

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

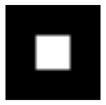
Source: C. Rasmussen

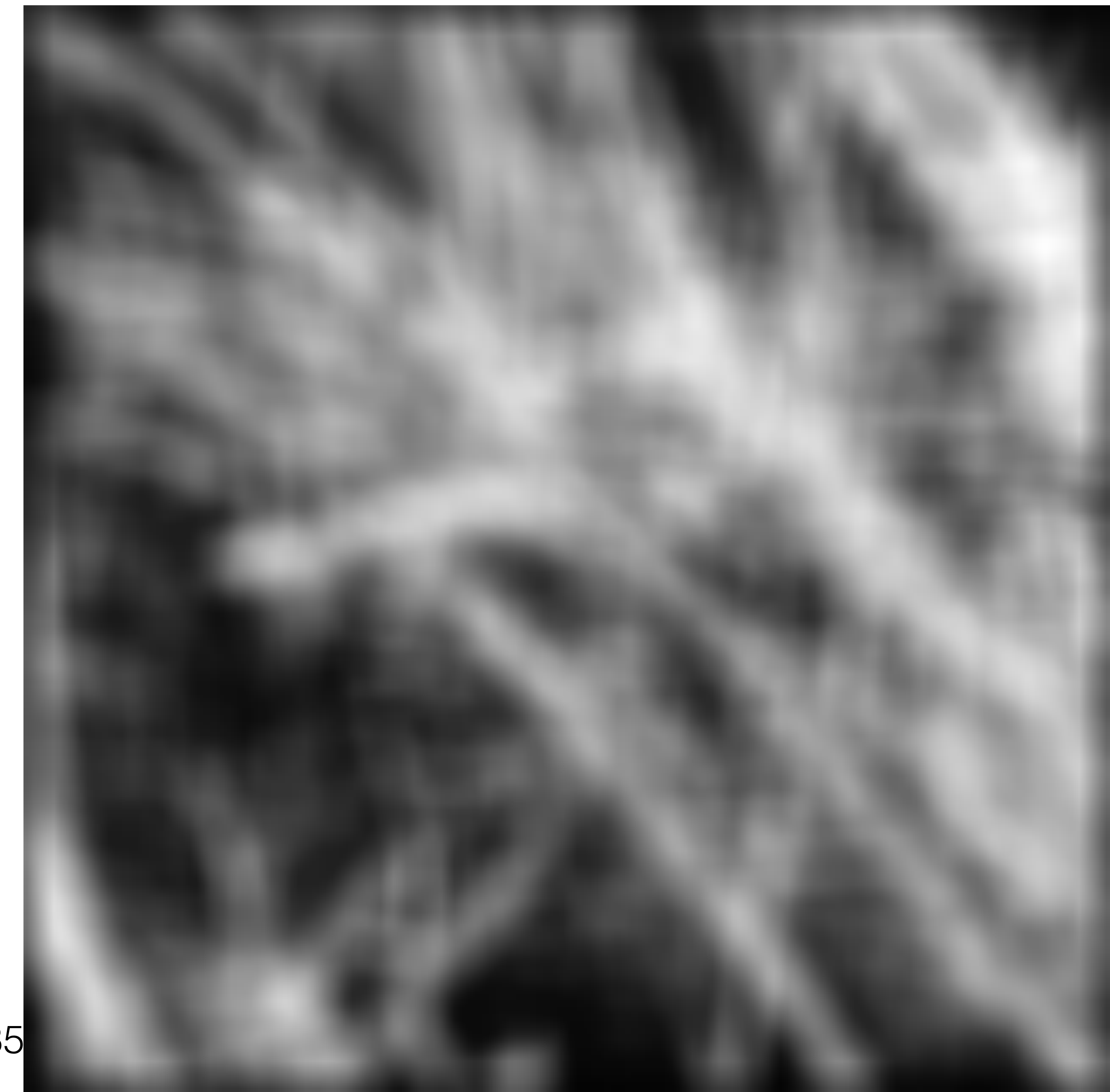
Atténuer avec le filtre gaussien





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 sigma, c’est équivalent à un filtre d’écart-type sigma*sqrt(2)
- Séparable
  - filtre gaussien 2D = produit de deux filtres gaussiens 1D

Séparabilité du filtre gaussien

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

Pourquoi c'est important?

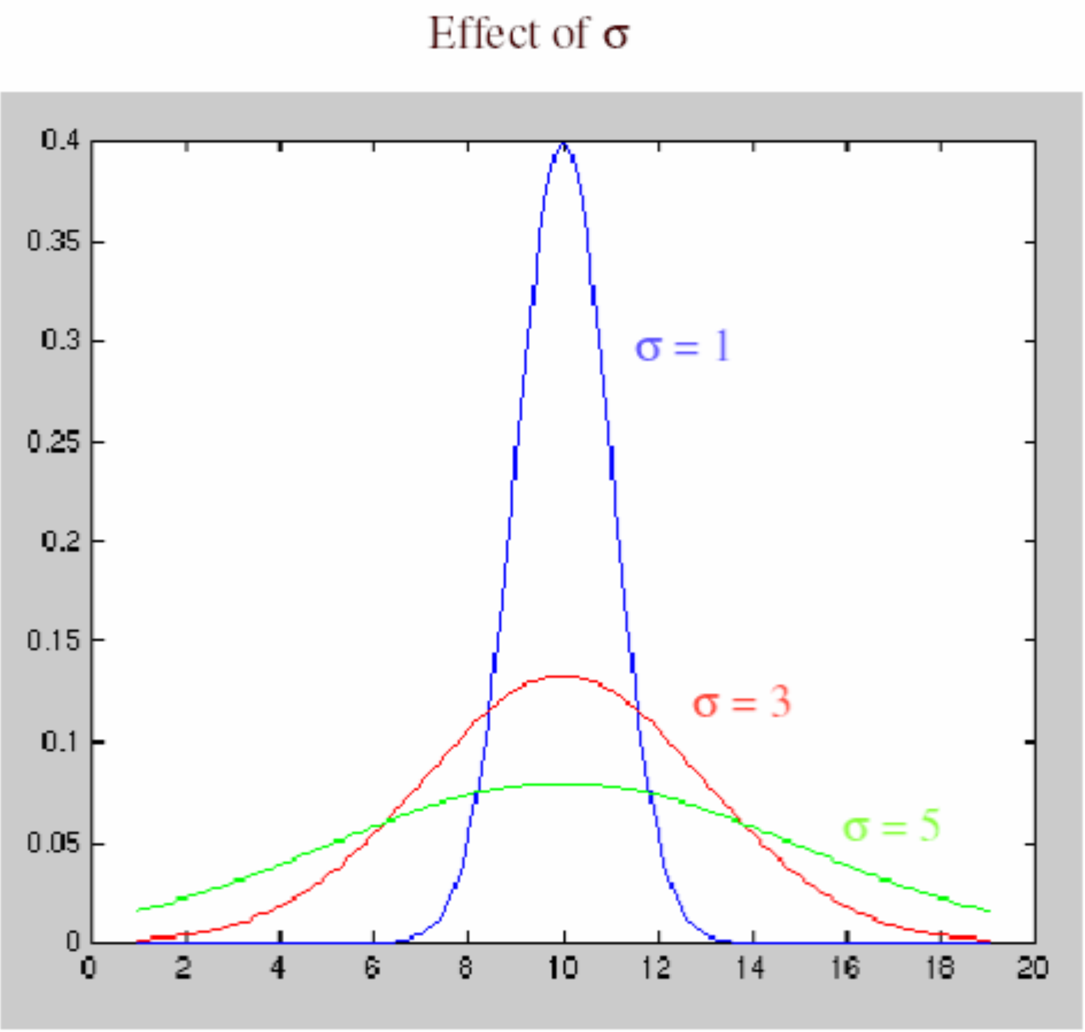
Source: D. Lowe

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

Source: S. Marschner

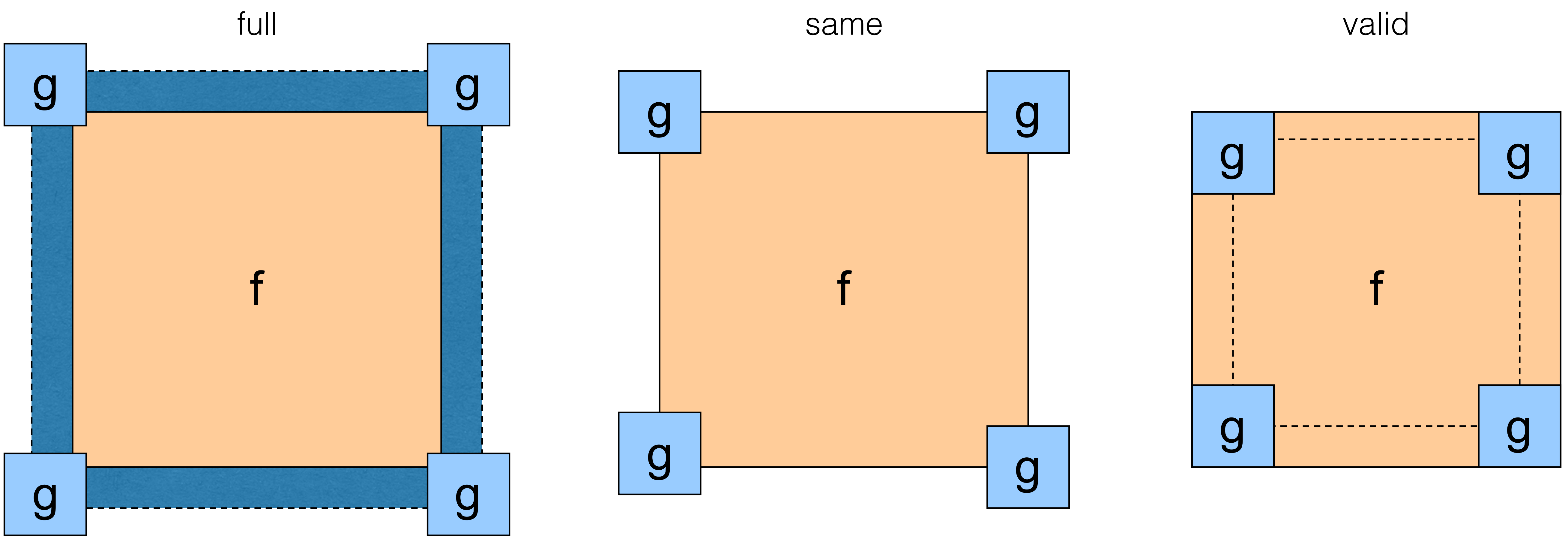
Considérations pratiques

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

Source: S. Marschner

Considérations pratiques

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



Source: S. Lazebnik

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:
  - gradx(y,x) = im(y,x+1) - im(y,x) pour tout x, y

Questions de révision

Remplir les trous:

Filtrage

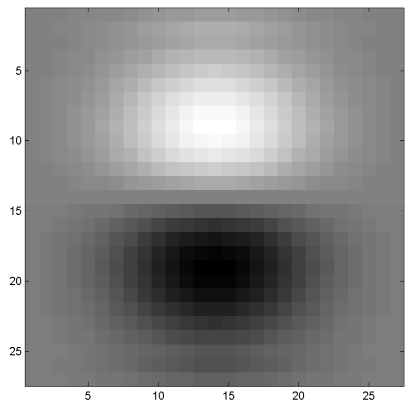
a) _ = D * B

b) A = _ * _

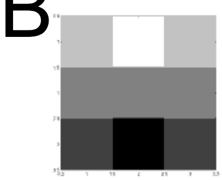
c) F = D * _

d) _ = D * D

A

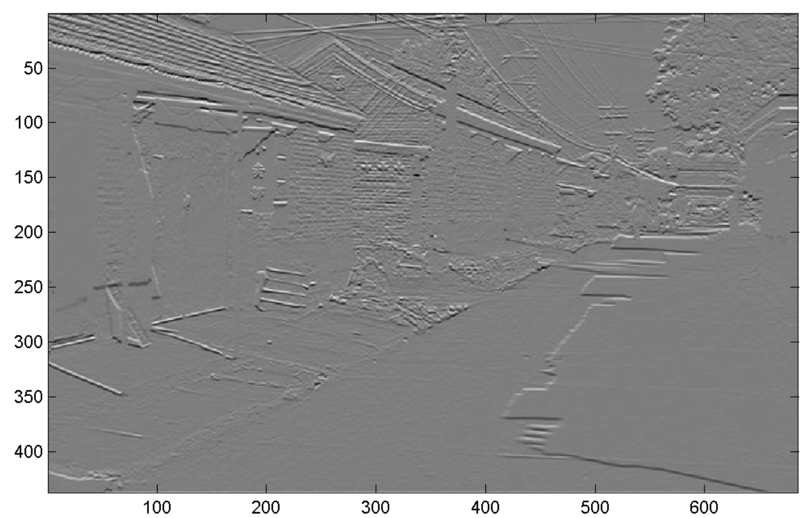


B

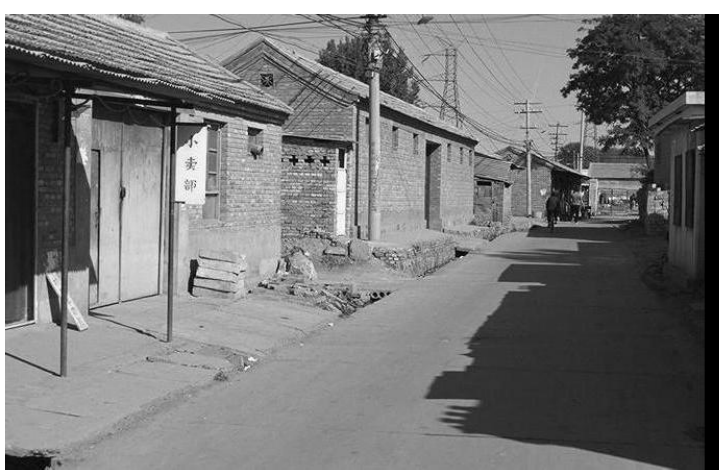


E

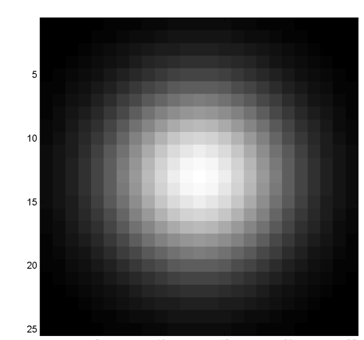
G



F

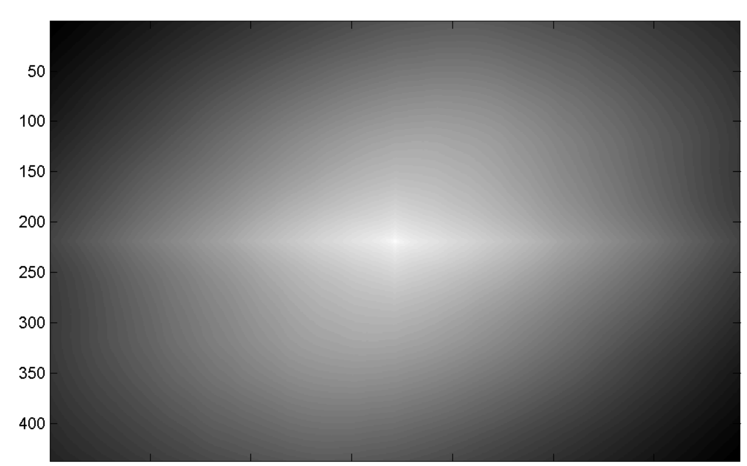


C

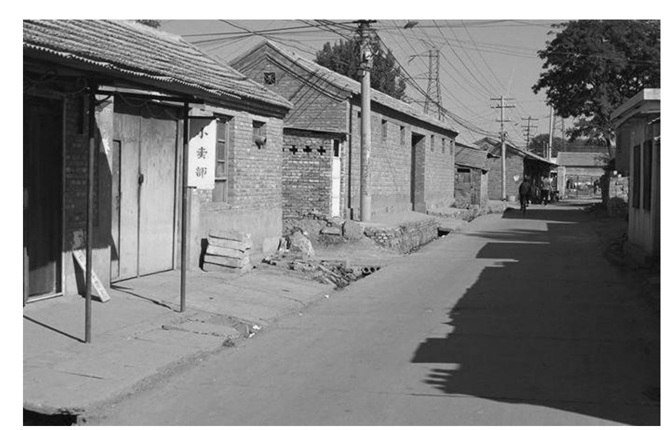


H

I



D



Source: D. Hoiem