



Imagerie et Vision Industrielle

4: Filtrage Linéaire d'Image



Menu du Jour

Filtrage d'Images

- **Image** comme **fonction**
- **Filtres Linéaires**
- **Exemples de filtres:** Box, Gaussien

Filtrage d'Images

Filtrage d'Images

```
graph TD; A([Filtrage d'Images]) --> B[Filtrage Spatial];
```

Filtrage Spatial

```
graph TD; A([Filtrage d'Images]) -->|Red Arrow| B[Filtrage Spatial]; A -->|Blue Arrow| C[Filtrage Frequentiel];
```

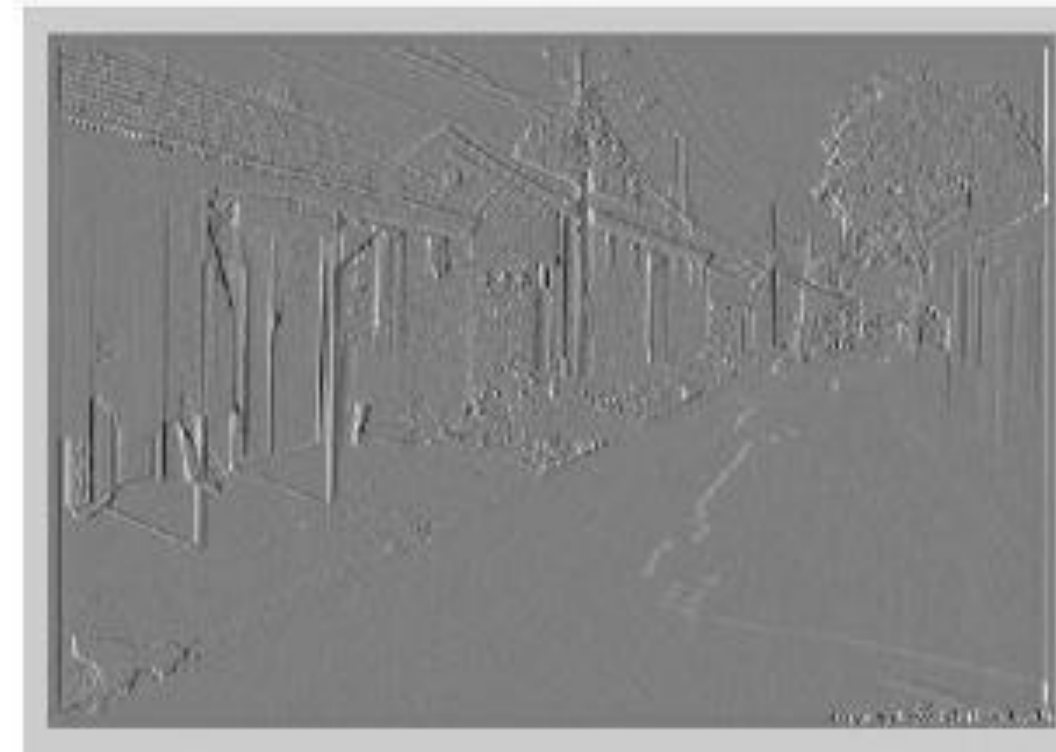
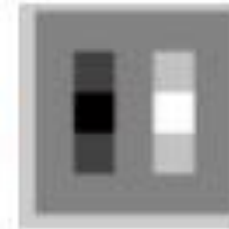
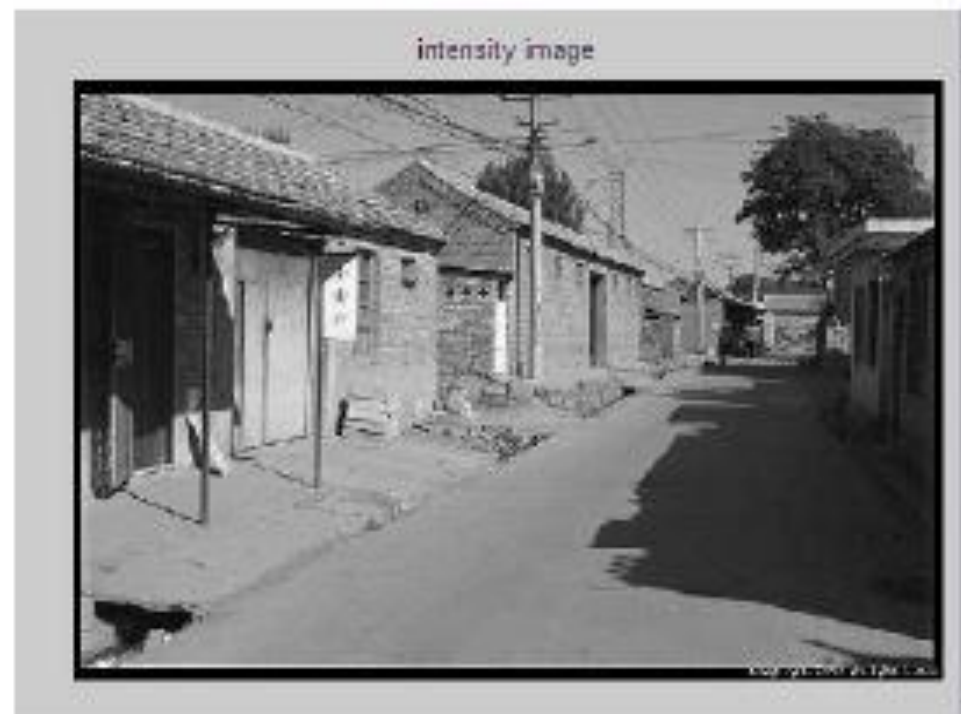
Filtrage d'Images

Filtrage Spatial

Filtrage Frequentiel

Filtrage dans le domaine spatial

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



Filtrage dans le domaine Fréquentiel

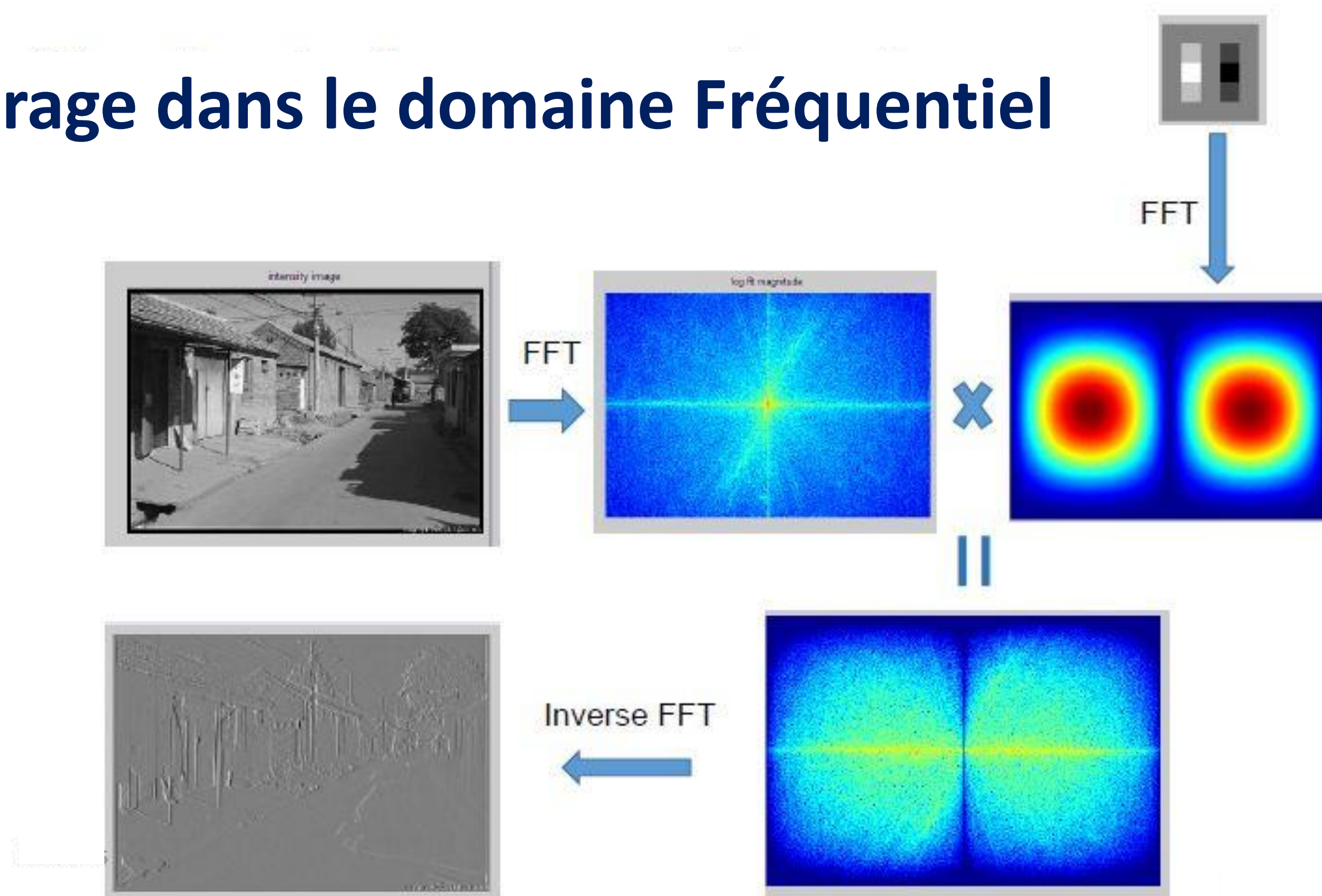


Image comme Fonction 2D

Une image à niveaux de gris (grayscale) est une fonction 2D



Image à niveaux de gris

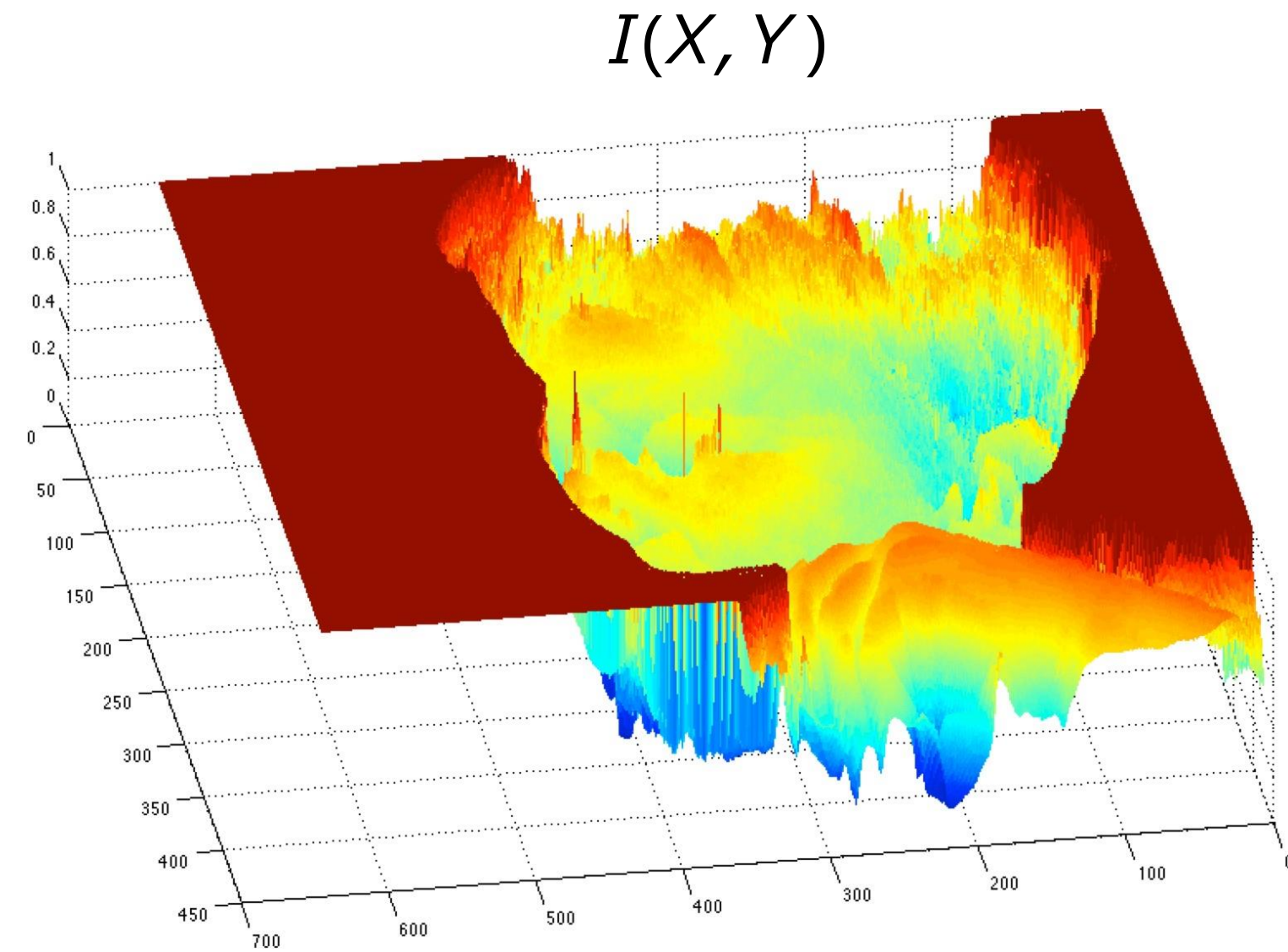


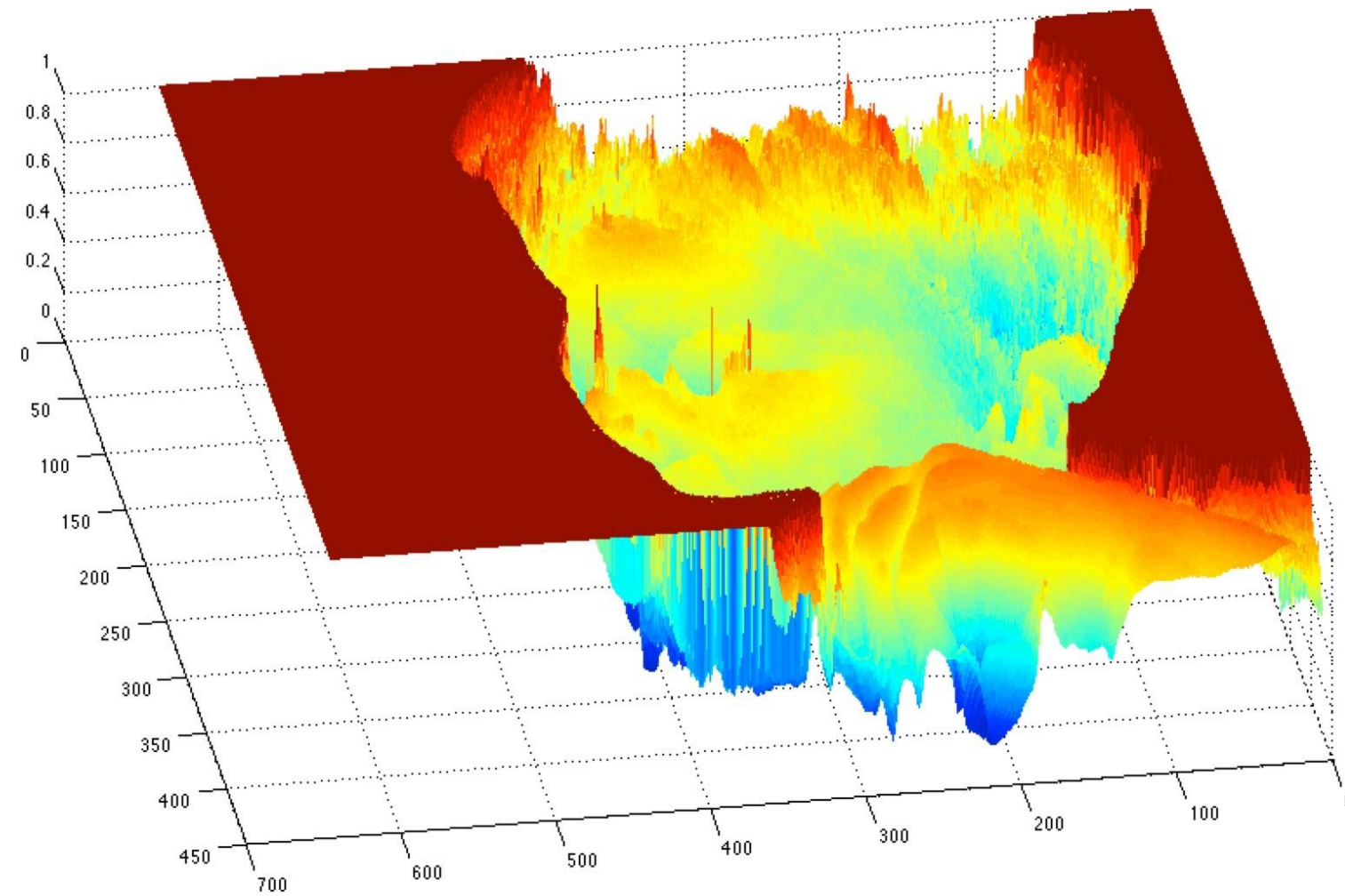
Image comme Fonction 2D

Une image à niveaux de gris (grayscale) est une fonction 2D



Image à niveaux de gris

$$I(X, Y)$$



domaine: $(X, Y) \equiv ([1, \text{largeur}], [1, \text{hauteur}])$

Image comme fonction 2D

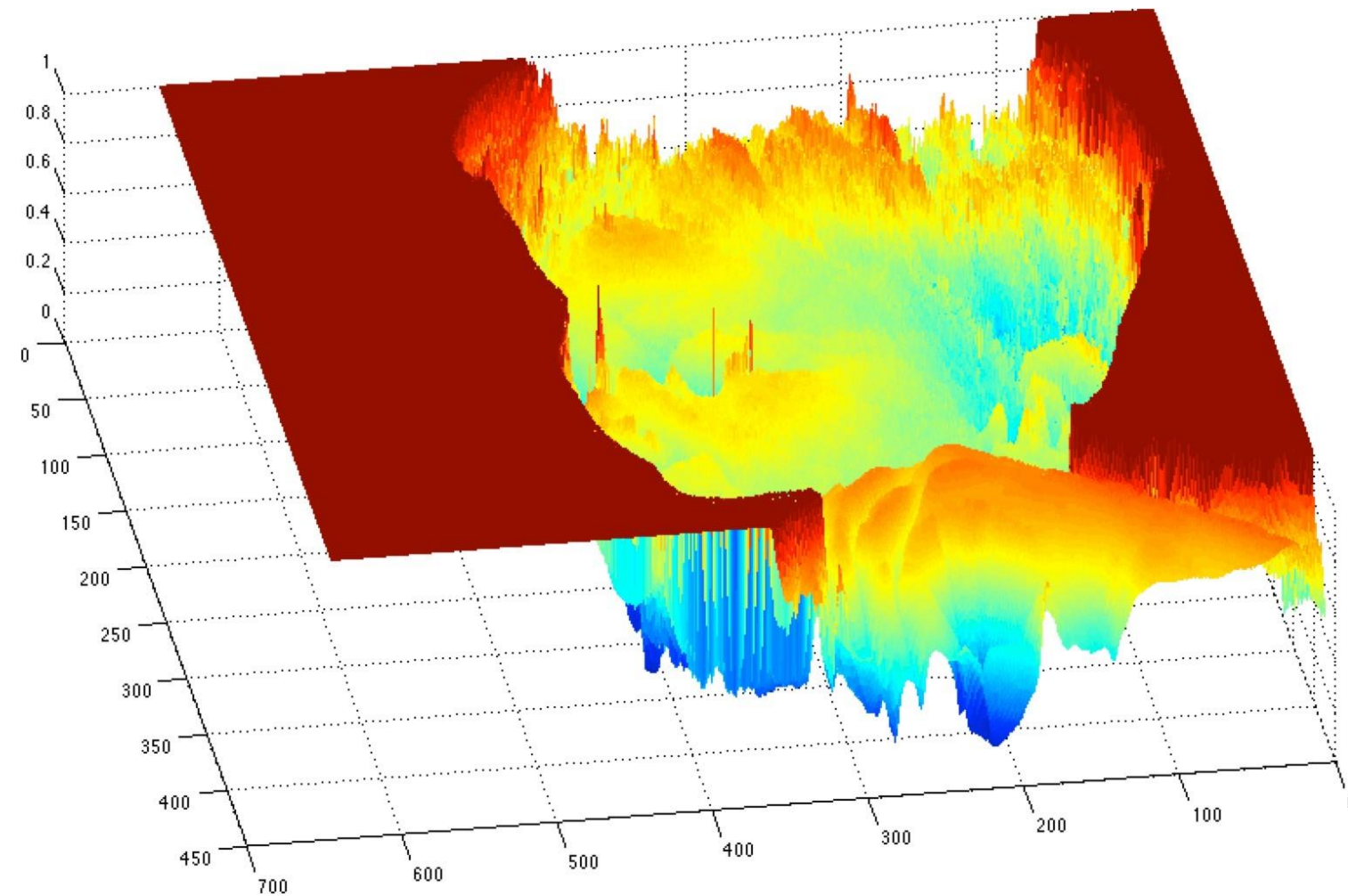
Une image à niveaux de gris (grayscale) est une fonction 2D



Image à niveaux de gris

Quelle est la dynamique
de cette fonction
image?

$$I(X, Y)$$



domaine: $(X, Y) \equiv ([1, \text{largeur}], [1, \text{hauteur}])$

Image comme fonction 2D

image (niveaux de gris) (grayscale)

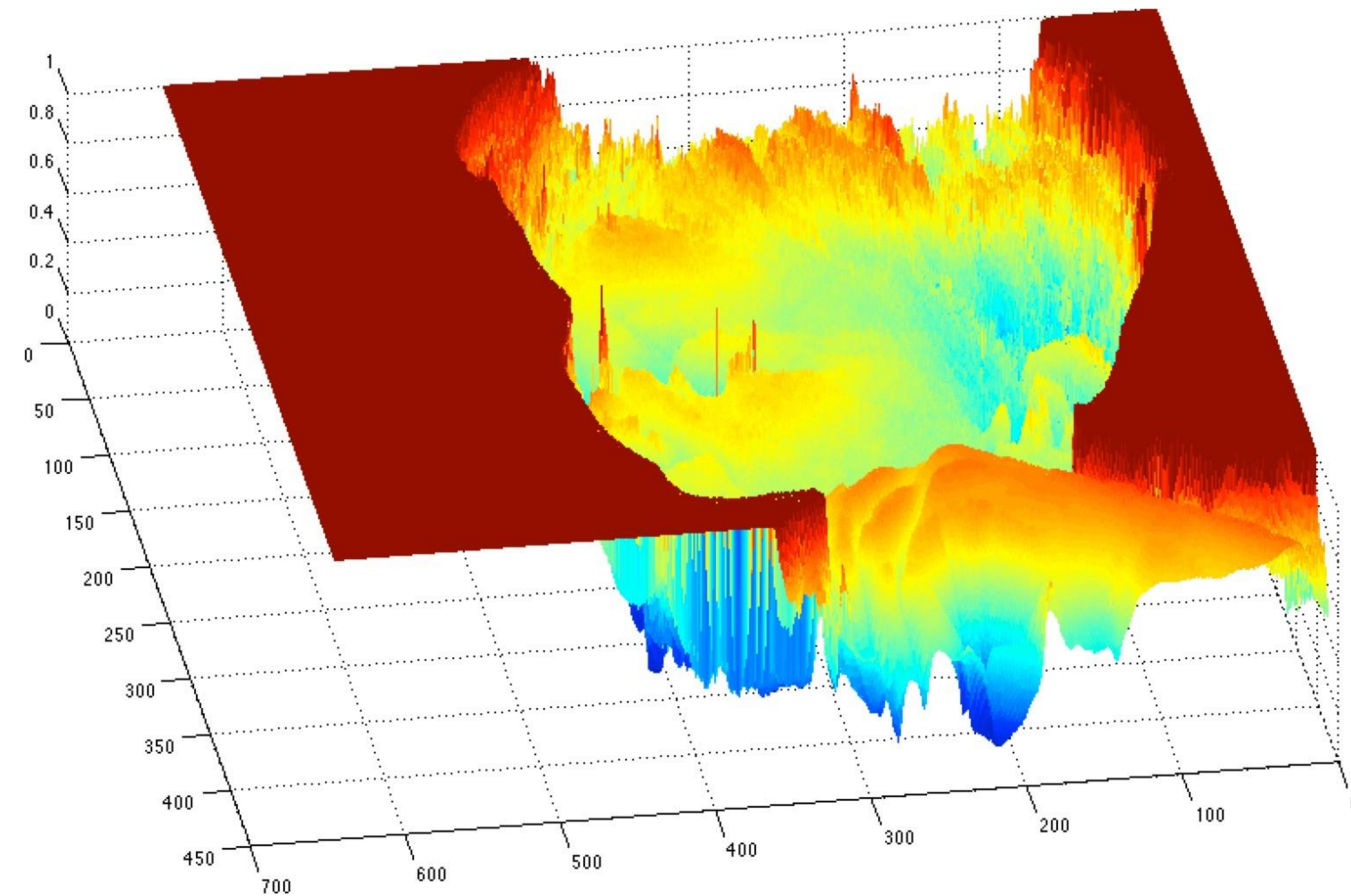


Image à niveaux de gris

Quelle est la dynamique
de cette fonction image?

$$I(X, Y) \in [0, 255]$$

$$I(X, Y)$$



domaine: $(X, Y) \equiv ([1, \text{largeur}], [1, \text{hauteur}])$

Addition de deux Images

Puisque les images sont des fonctions, on peut donc leur appliquer des opérations, e.g., **moyenne** (average)



$I(X, Y)$



$G(X, Y)$



$$\frac{I(X, Y)}{2} + \frac{G(X, Y)}{2}$$

Addition de deux Images



$$a = \frac{I(X, Y)}{2} + \frac{G(X, Y)}{2}$$



$$b = \frac{I(X, Y) + G(X, Y)}{2}$$

Addition de deux Images



$$a = \frac{I(X, Y)}{2} + \frac{G(X, Y)}{2}$$

Question:

$$a = b$$

$$a > b$$

$$a < b$$



$$b = \frac{I(X, Y) + G(X, Y)}{2}$$

Addition de deux Images



pixel en rouge (image cameraman) = 98
pixel en rouge (image de la lune) = 200

$$\frac{98}{2} + \frac{200}{2} = 49 + 100 = 149$$

Question:

$$a = b$$

$$a > b$$

$$a < b$$



$$\frac{98 + 200}{2} = \frac{[298]}{2} = \frac{255}{2} = 127$$

Types de transformations qu'on peut faire?

$I(X, Y)$



Filtrage



$I^0(X, Y)$



Changer la dynamique de la fonction image

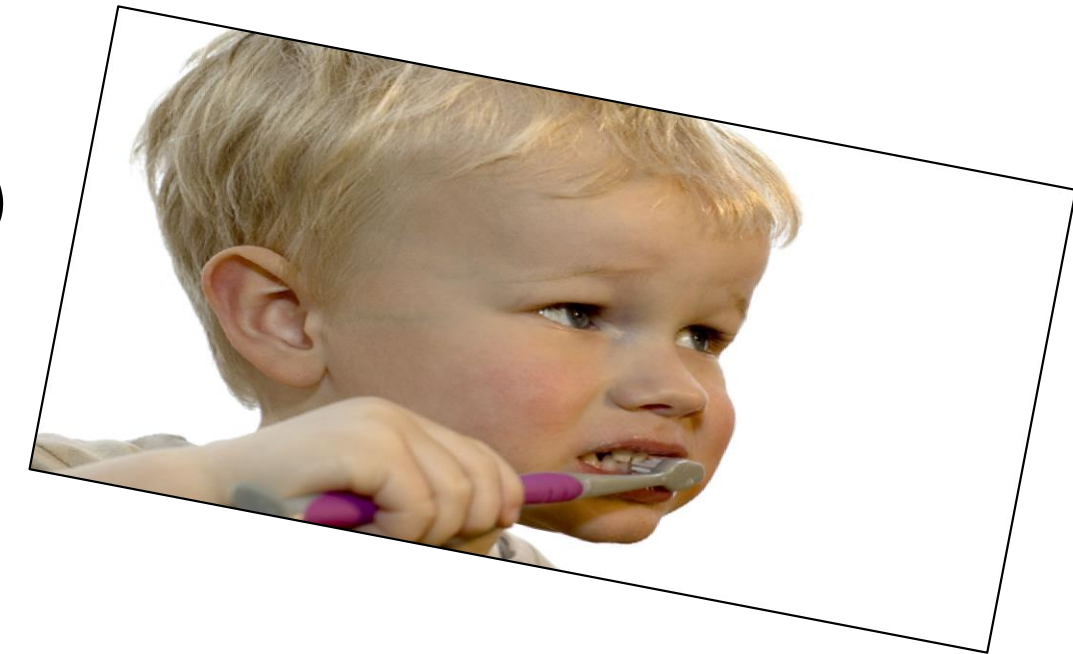
$I(X, Y)$



Déformer



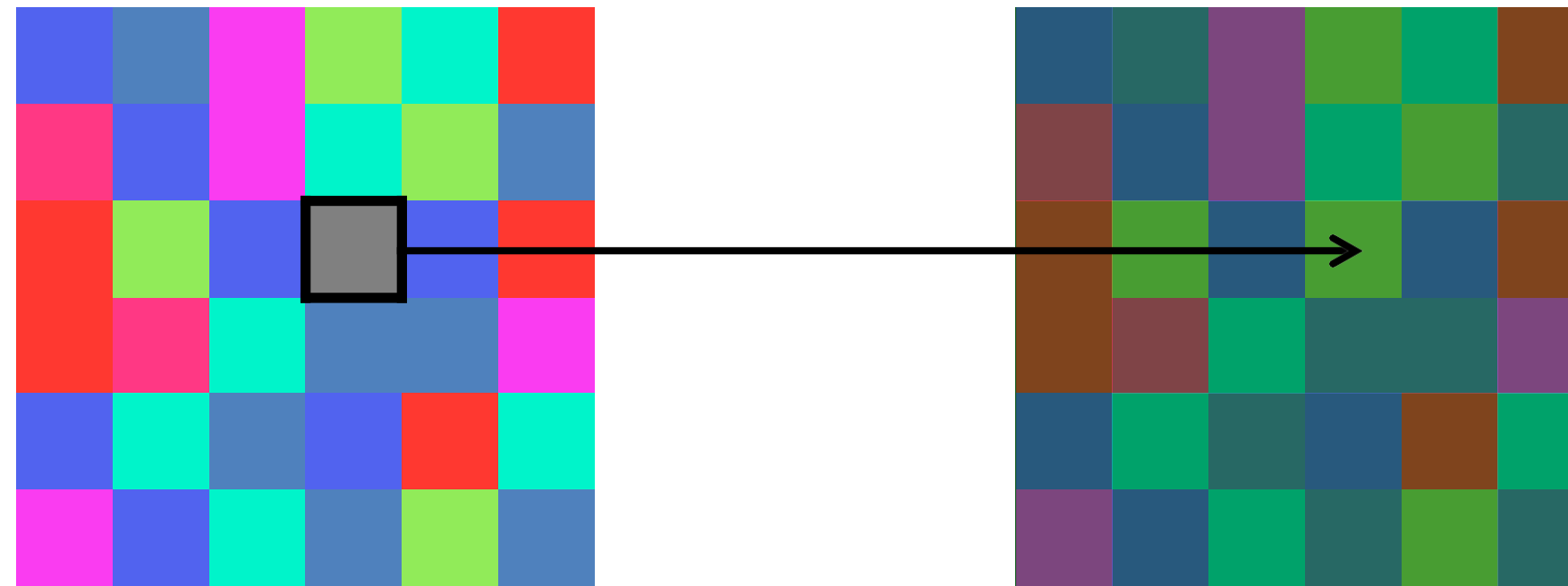
$I^0(X, Y)$



Changer le domaine de la fonction image

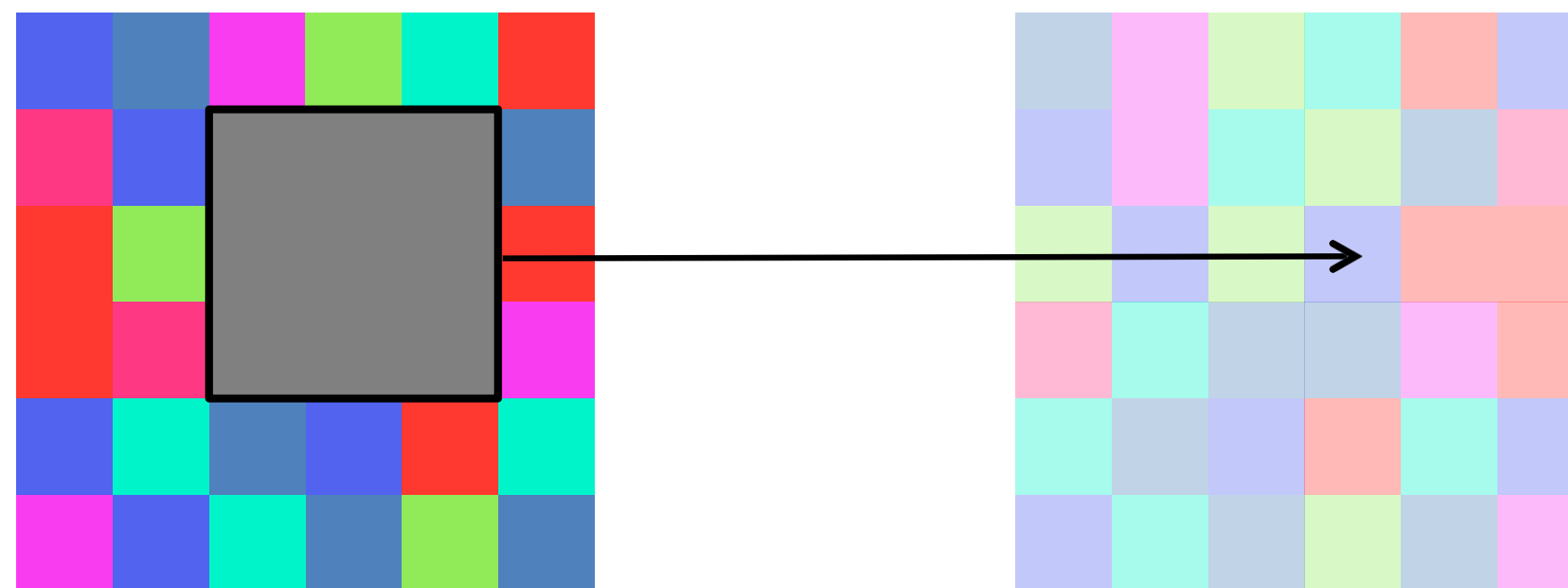
Types de **filtrage**?

Operation **Point**



Traitement ponctuel

Operation sur **Voisinage**



“filtrage”

Exemples de Traitement par Point

originale



noircie



Contraste réduit



$$I(X, Y)$$

inverser



illuminer



Contraste élevé



Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



inverser



illuminer



Contraste élevé



Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



$$\frac{I(X, Y)}{255}$$

inverser



illuminer



Contraste élevé



Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



$$\frac{I(X, Y)}{2}$$

inverser



illuminer



Contraste élevé



Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



$$\frac{I(X, Y)}{2}$$

inverser



$$255 - I(X, Y)$$

illuminer



Contraste élevé



Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



$$\frac{I(X, Y)}{2}$$

inverser



$$255 - I(X, Y)$$

illuminer



$$I(X, Y) + 128$$

Contraste élevé



Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



$$\frac{I(X, Y)}{2}$$

inverser



$$255 - I(X, Y)$$

illuminer



$$I(X, Y) + 128$$

Contraste élevé



$$I(X, Y) * 2$$

Exemples de Traitement par Point

originale



$$I(X, Y)$$

noircie



$$I(X, Y) - 128$$

Contraste réduit



$$\frac{I(X, Y)}{2}$$

inverser



$$255 - I(X, Y)$$

illuminer



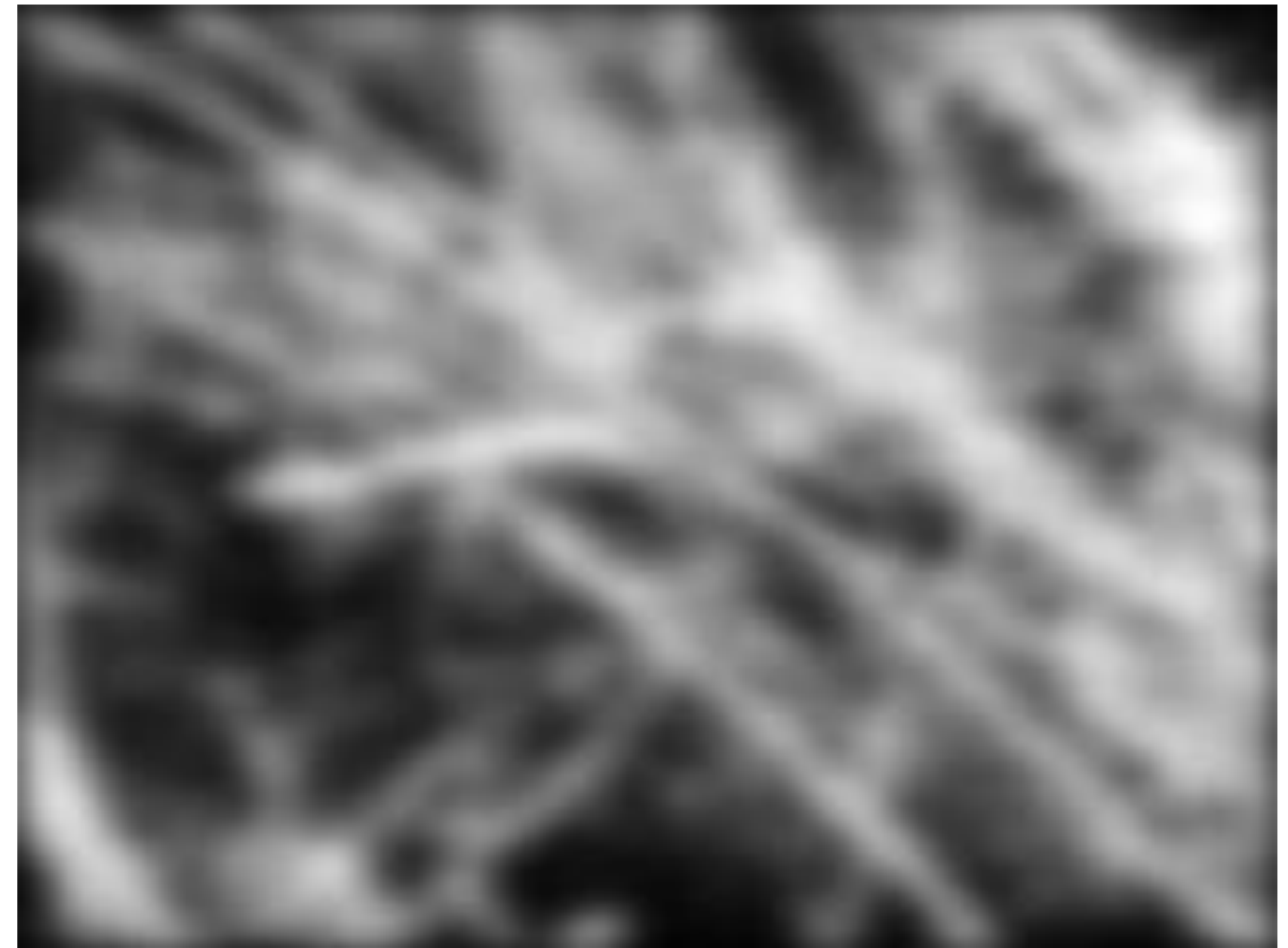
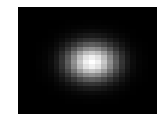
$$I(X, Y) + 128$$

Contraste élevé



$$I(X, Y) * 2$$

Filtrage Linéaire



Exemple 'amusant' du Jour: Clever Hans



Clever Hans
(Orlov Trotter horse)

**Wilhelm
von Osten**

Exemple 'amusant' du Jour: Clever Hans



Hans peut répondre à 89% des questions de math
correctement

Exemple 'amusant' du Jour: Clever Hans



Clever Hans
(Orlov Trotter horse)

Wilhelm
von Osten

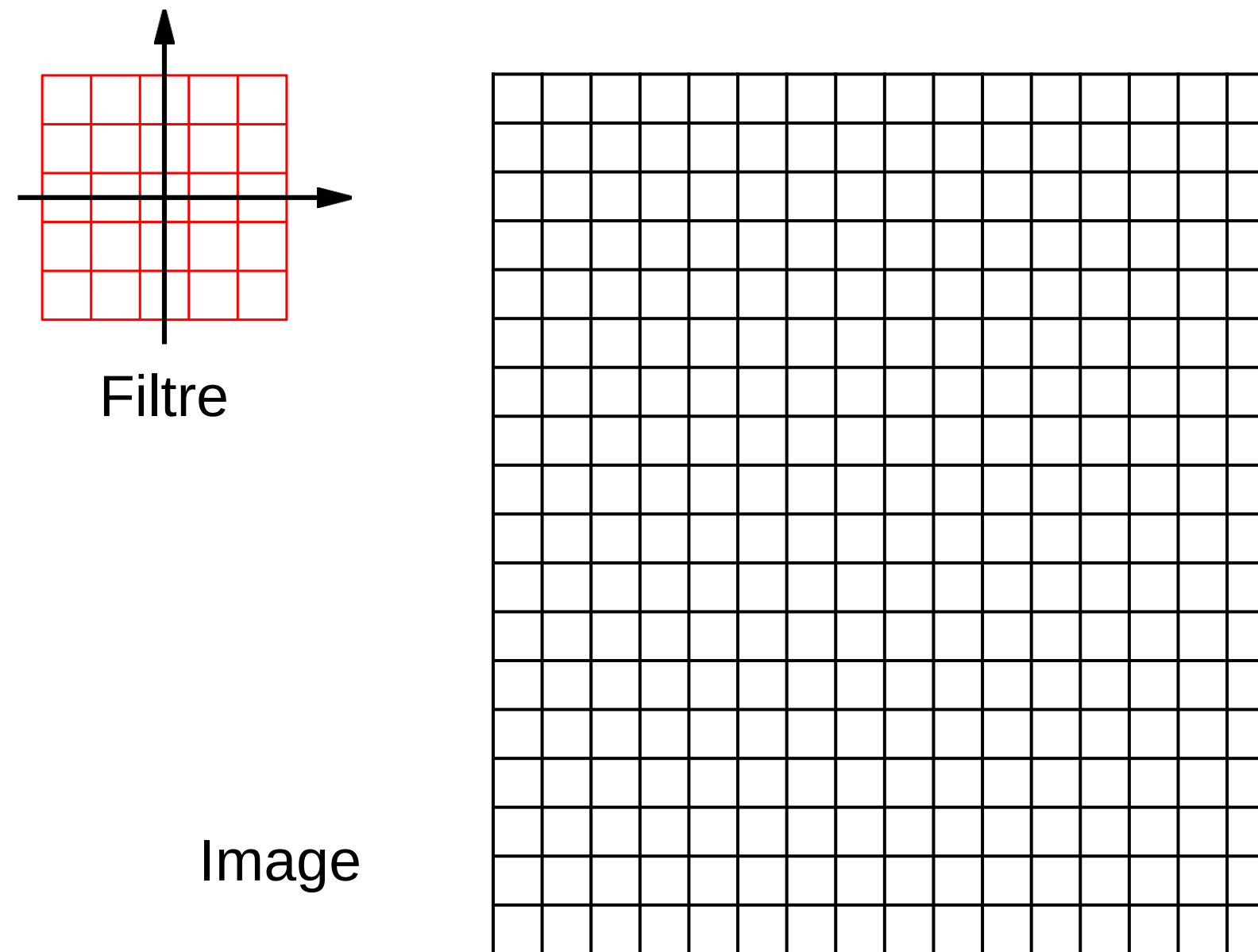
Le cours était intelligent, juste à l'inverse de la façon que pensait van Osten!

Hans peut répondre à 89% des questions de math correctement

Filtres Linéaires

Soit $I(X, Y)$ une image numérique $n \rightarrow n$ (on prend largeur = hauteur)

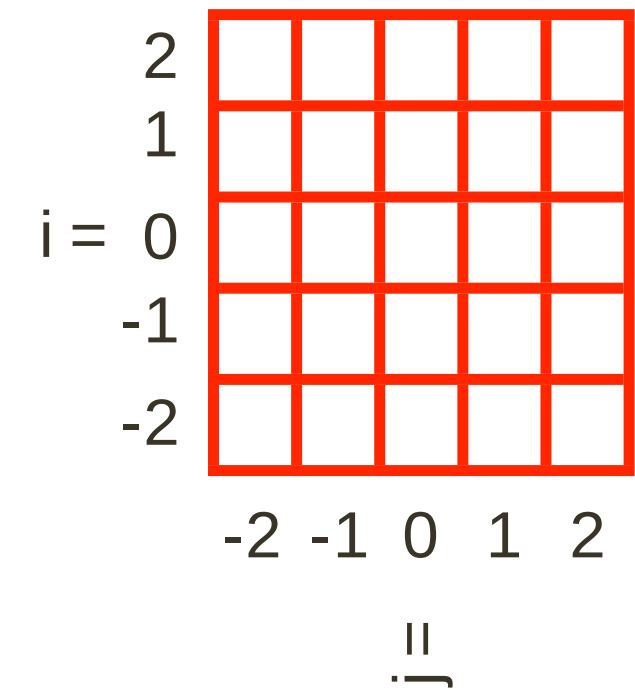
Soit $F(X, Y)$ une autre image $m \rightarrow m$ (notre “**filtre**” ou “**noyau**”)



On suppose m est impaire. (ici, $m = 5$)

Filtres Linéaires

Soit $k = \frac{m}{2}$



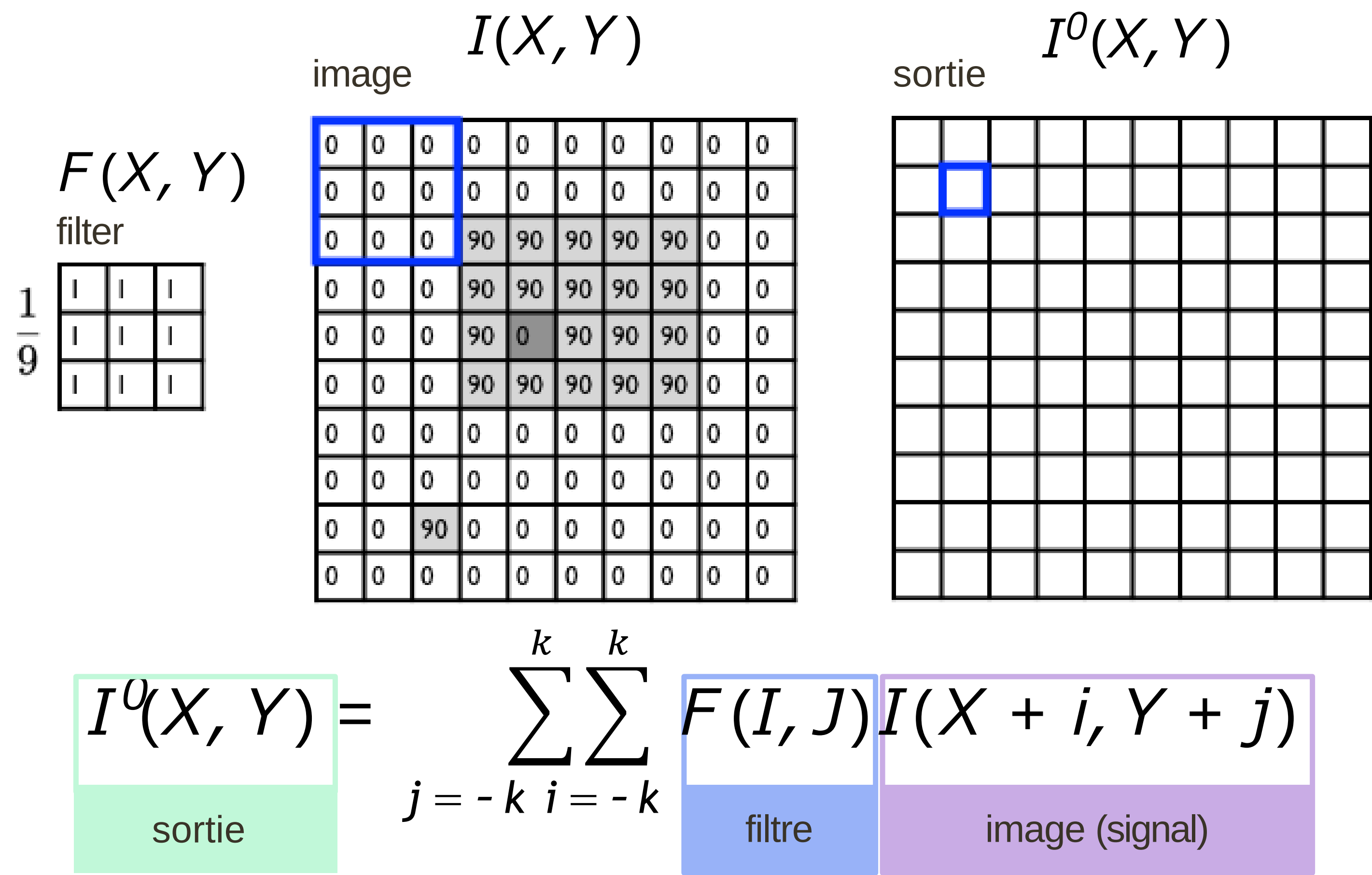
On calcule une nouvelle image, $I^0(X, Y)$, comme suit:

$$\boxed{I^0(X, Y)} = \sum_{j=-k}^k \sum_{i=-k}^k \boxed{F(I, J)} \boxed{I(X + i, Y + j)}$$

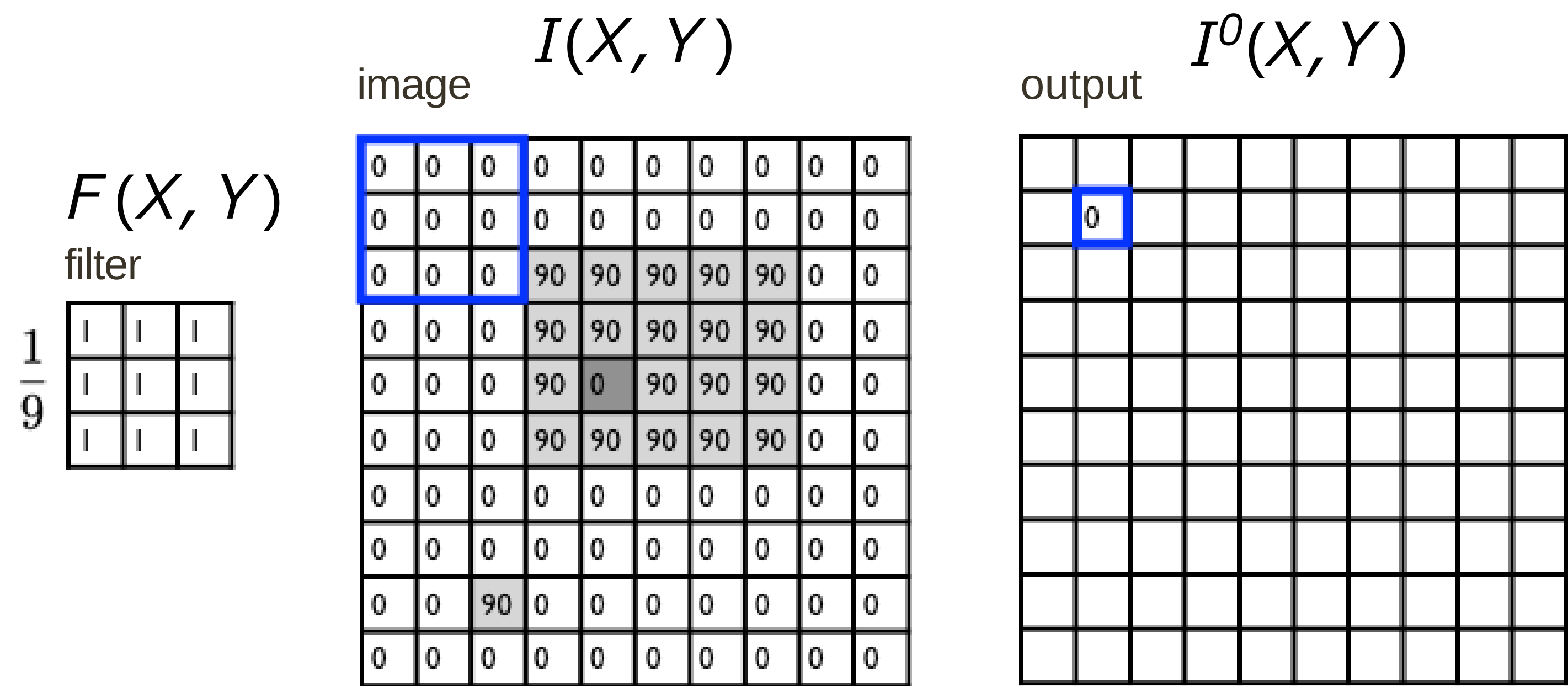
sortie filtre image (signal)

Intuition: chaque pixel dans l'image de sortie est une combinaison linéaire du même pixel et ses pixels voisins dans l'image d'origine

Exemple de Filtre Linéaire



Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

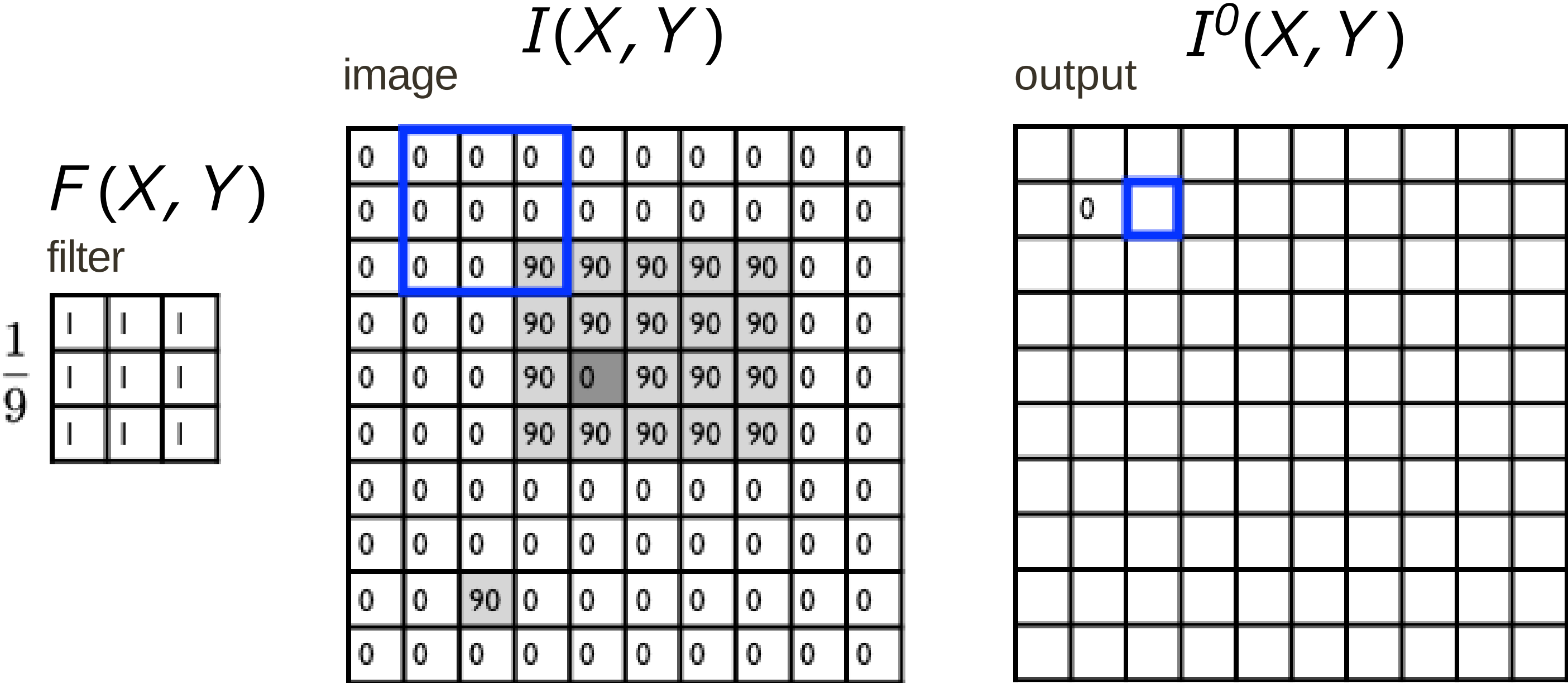
$=$

$\sum_{j = -k}^k \sum_{i = -k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

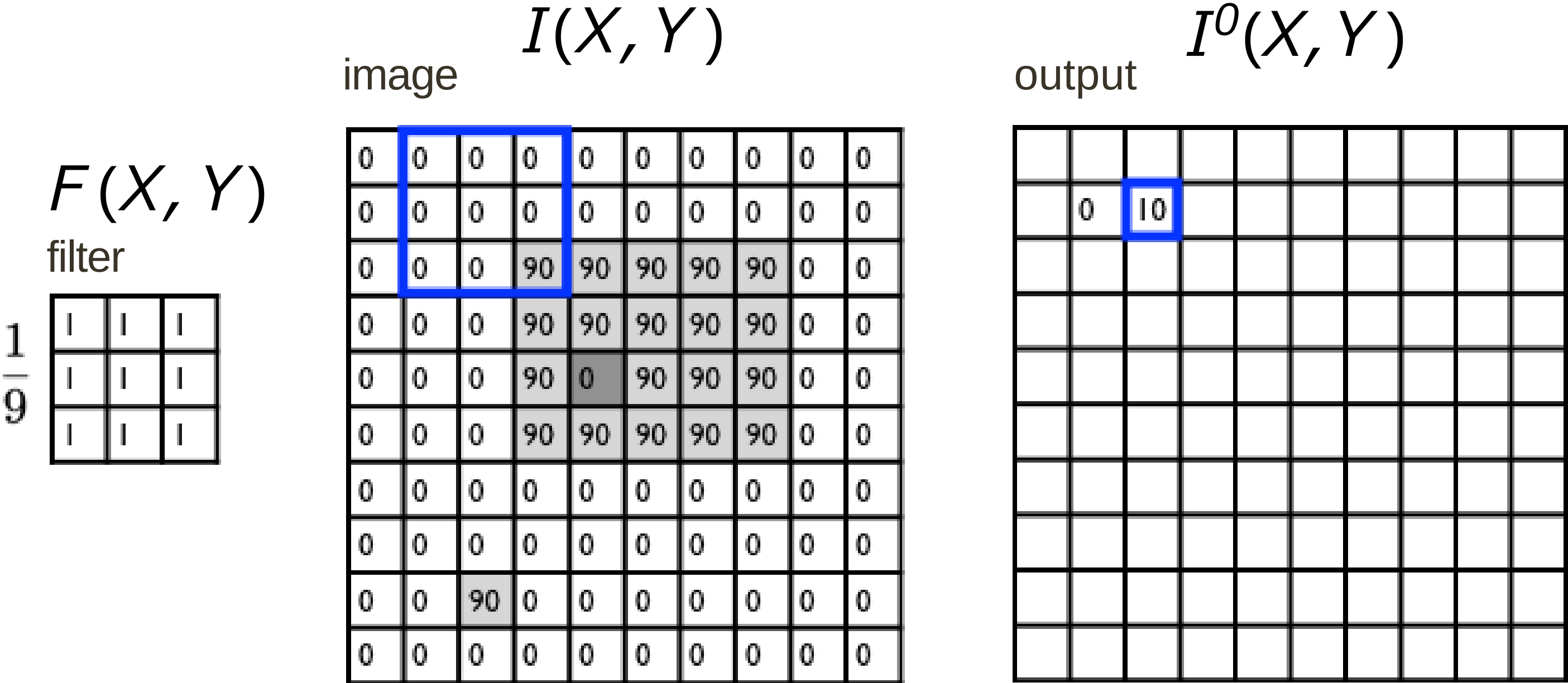
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

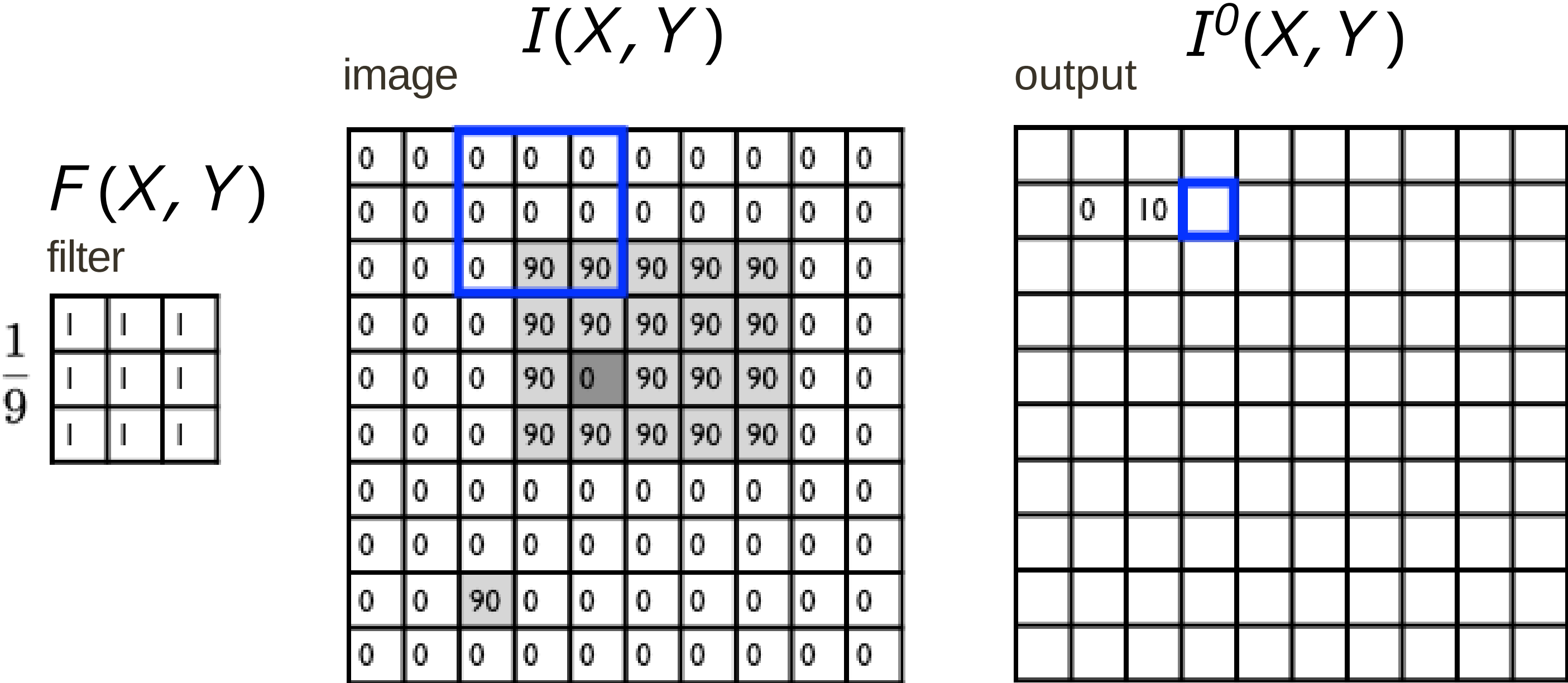
 $=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$

sortie

$=$

$\sum_{j = -k}^k \sum_{i = -k}^k$

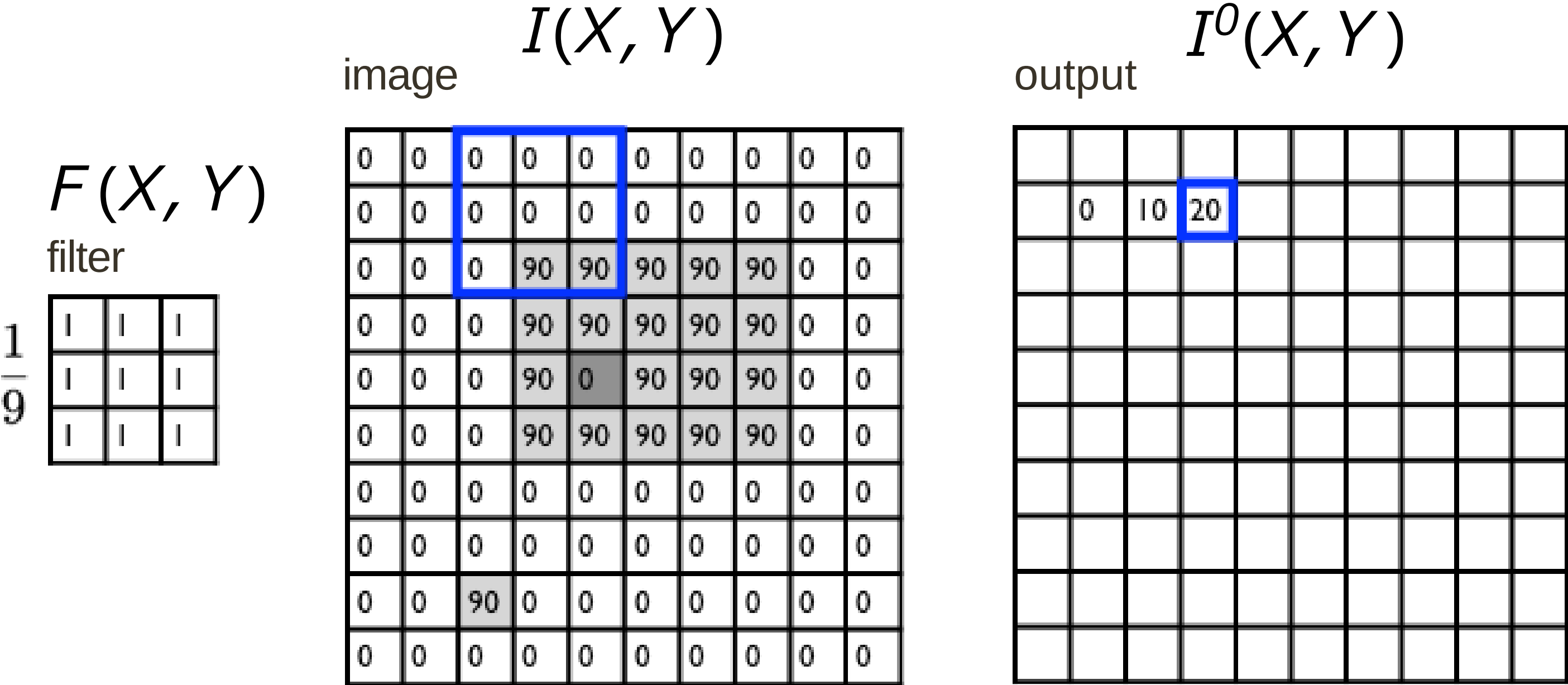
$F(I, J)$

filtre

$I(X + i, Y + j)$

image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

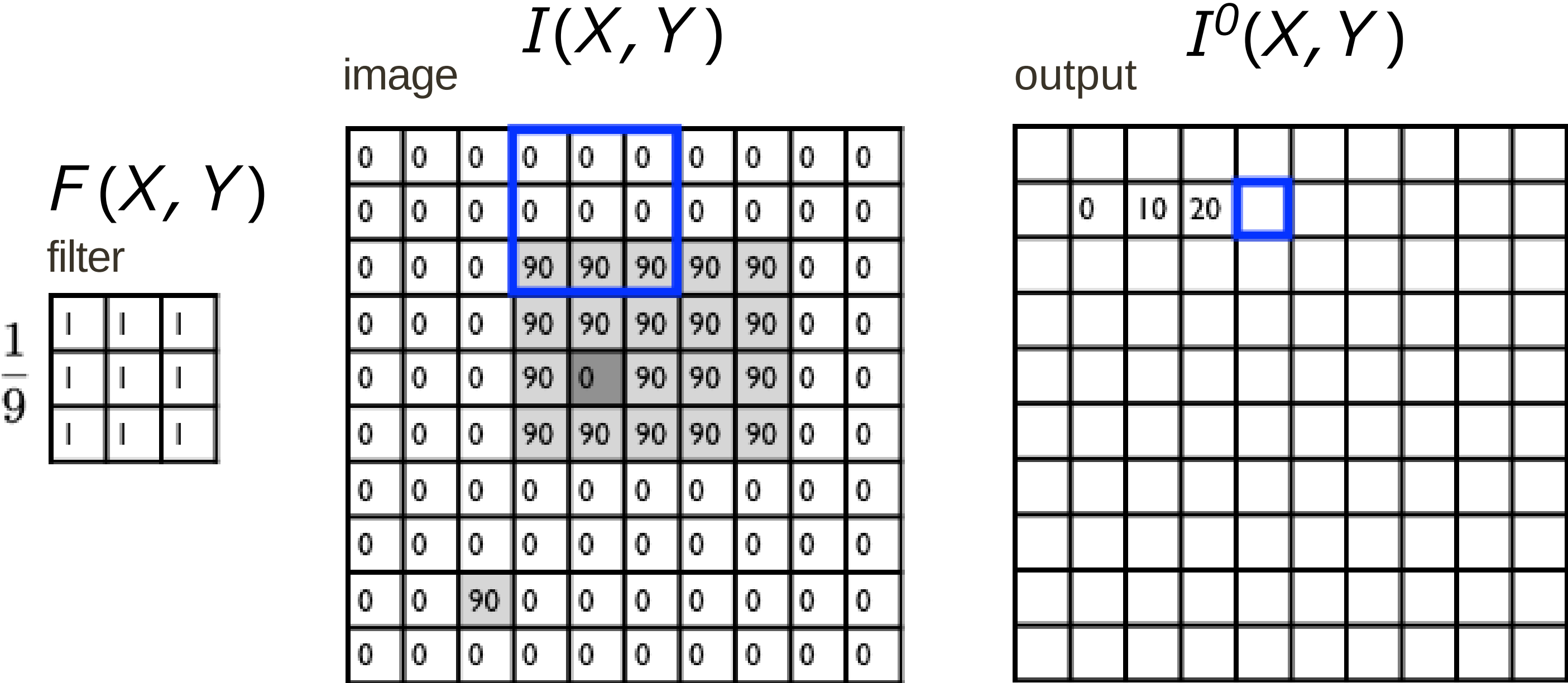
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

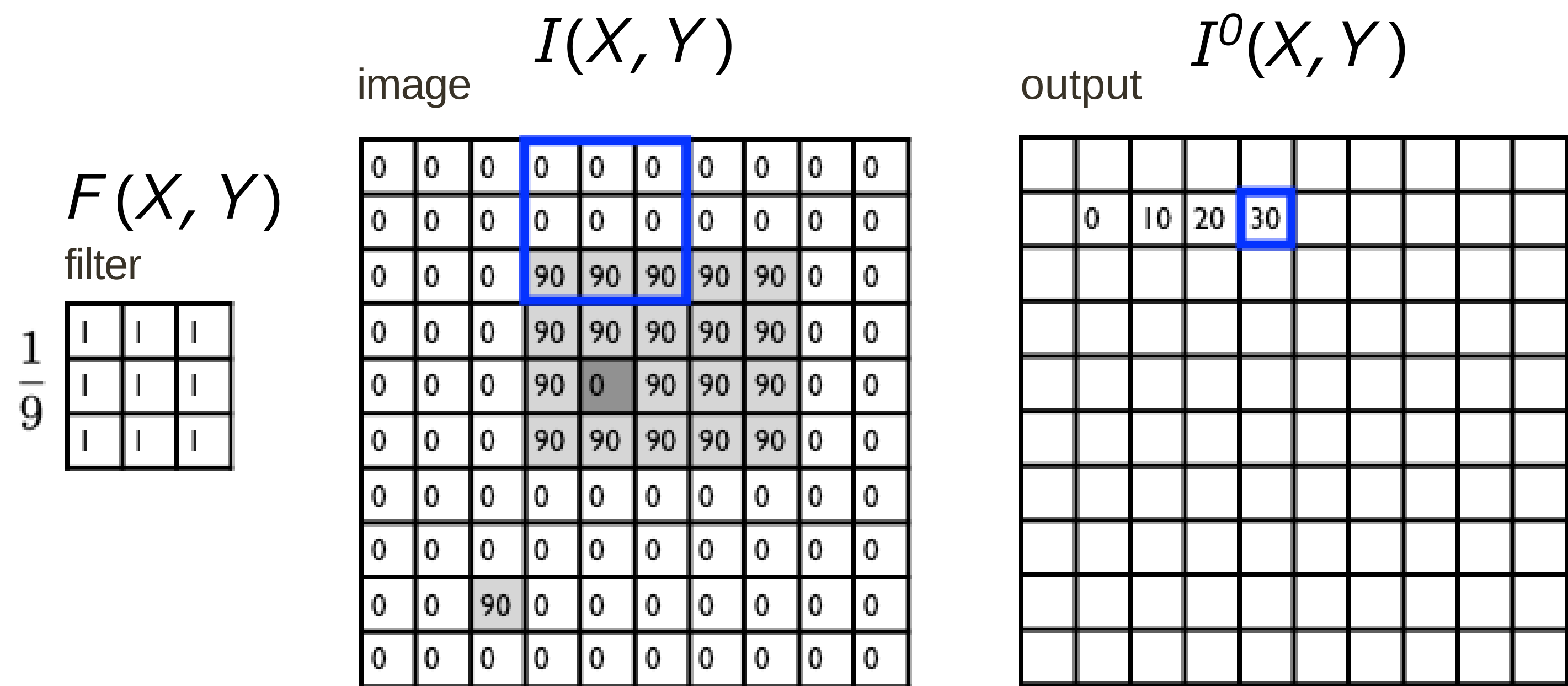
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

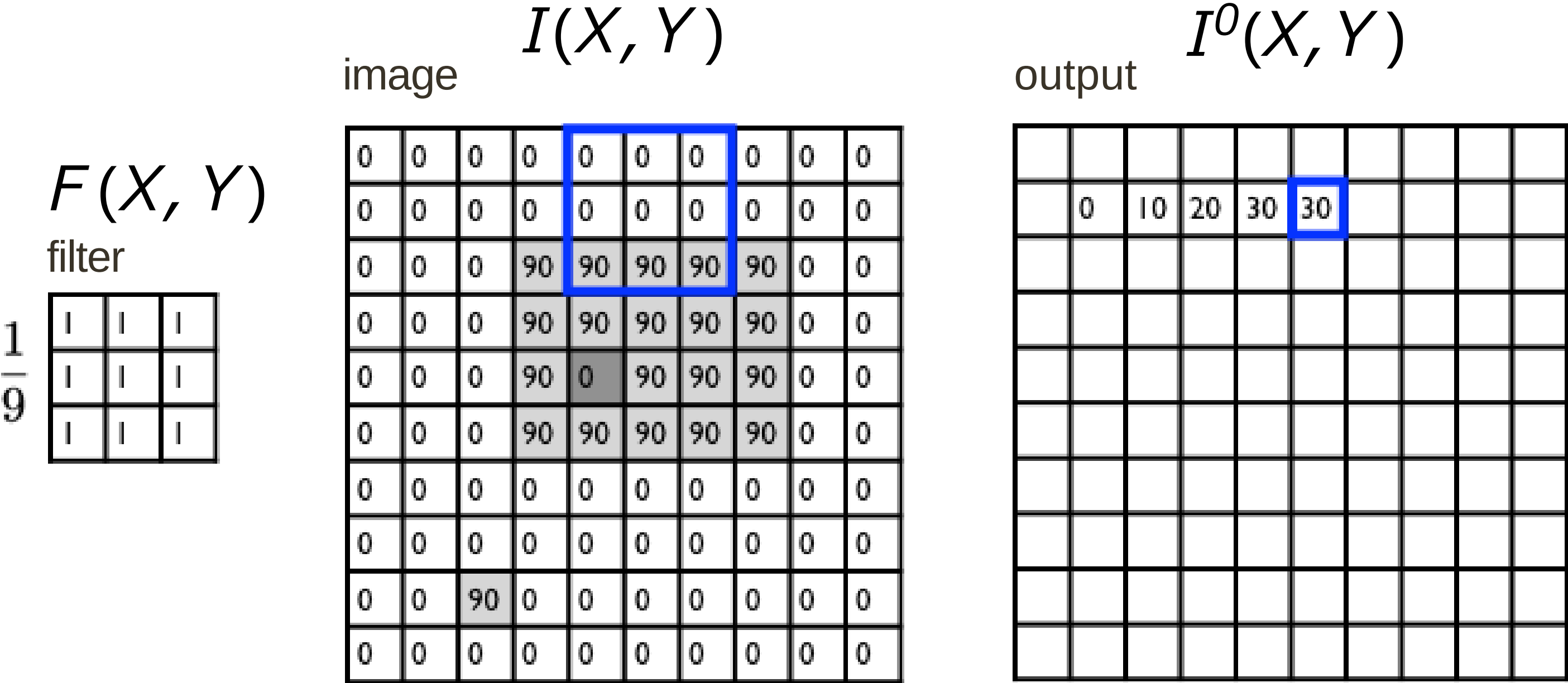
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

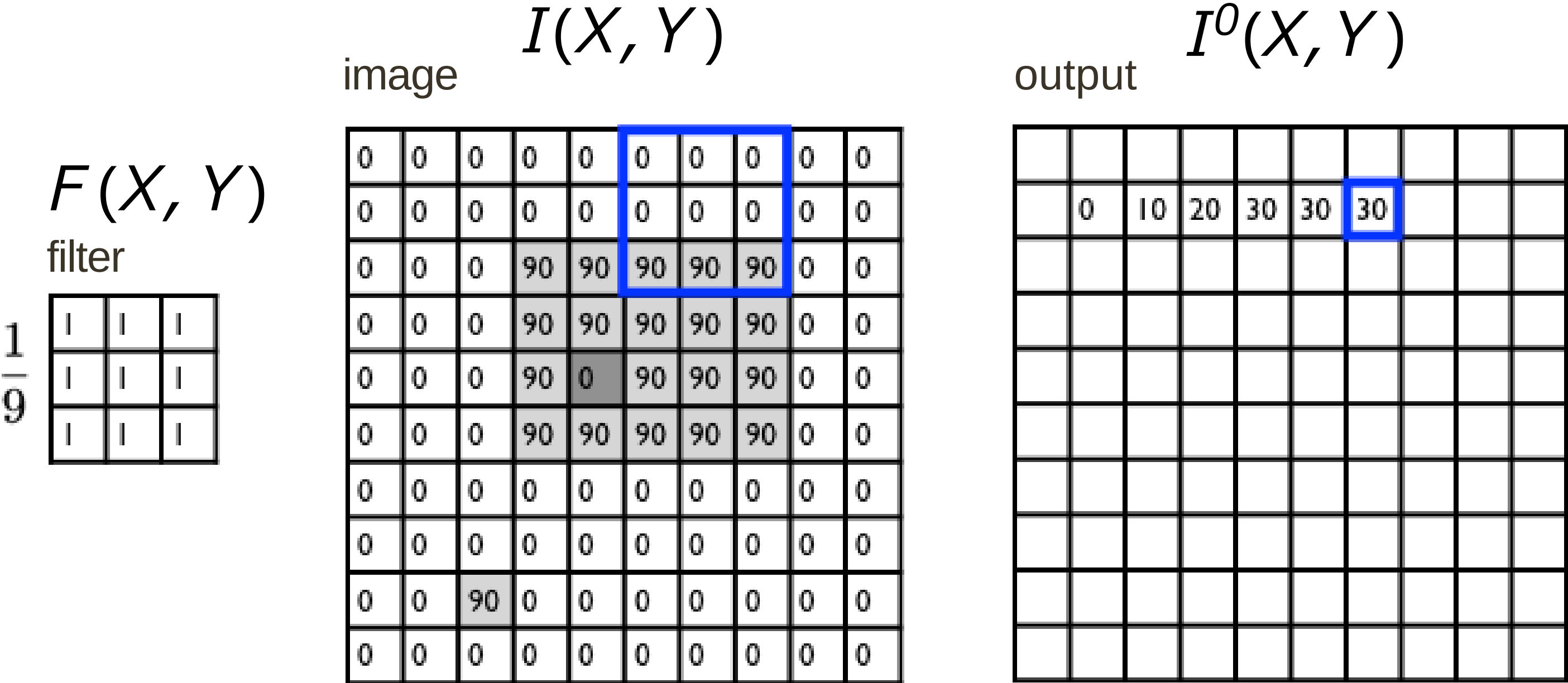
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

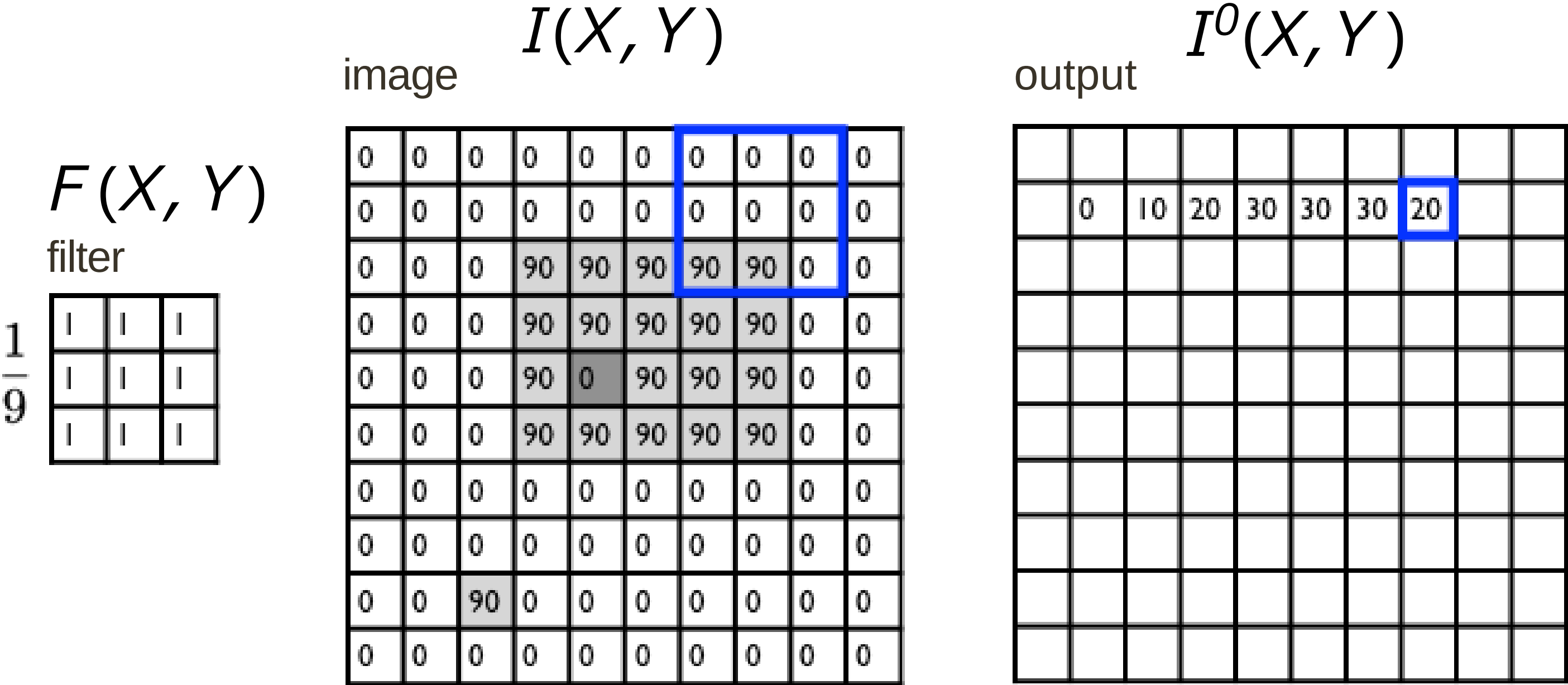
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

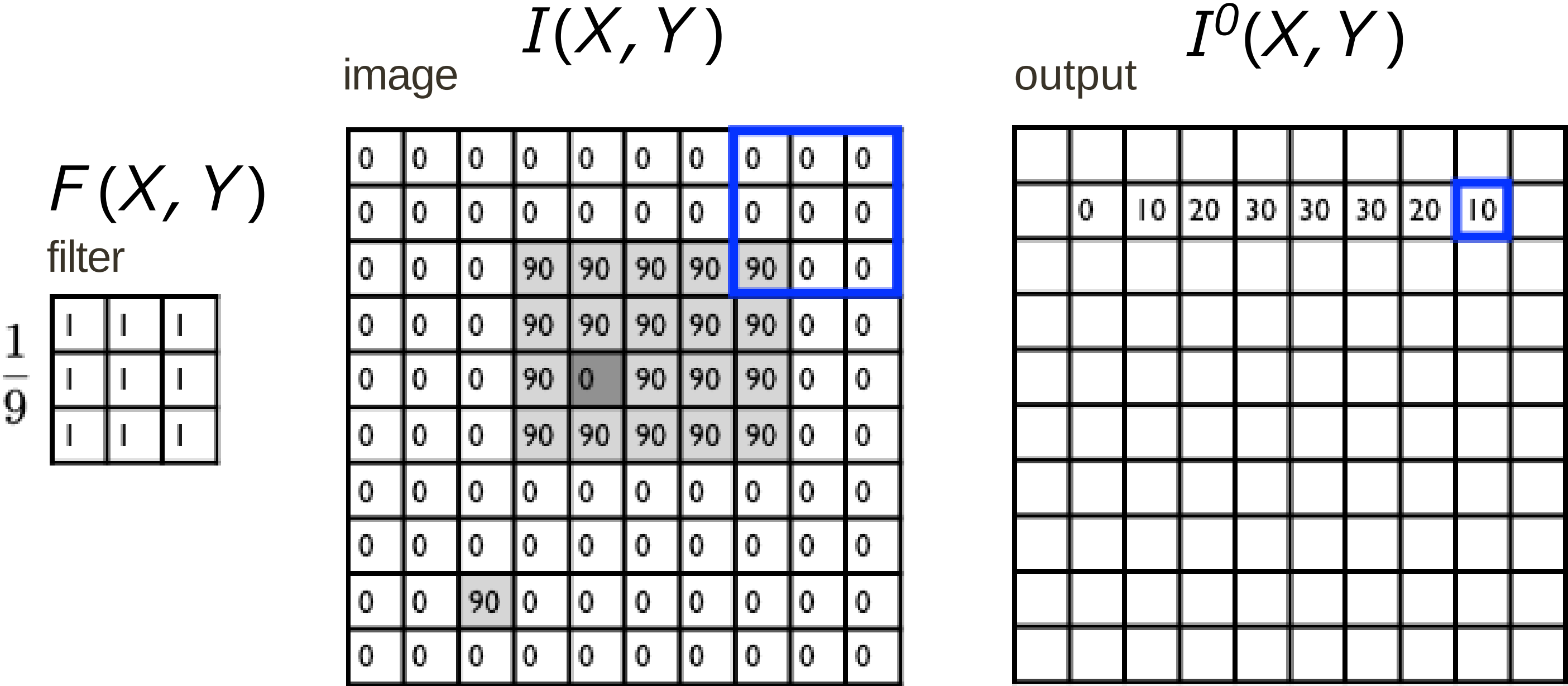
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

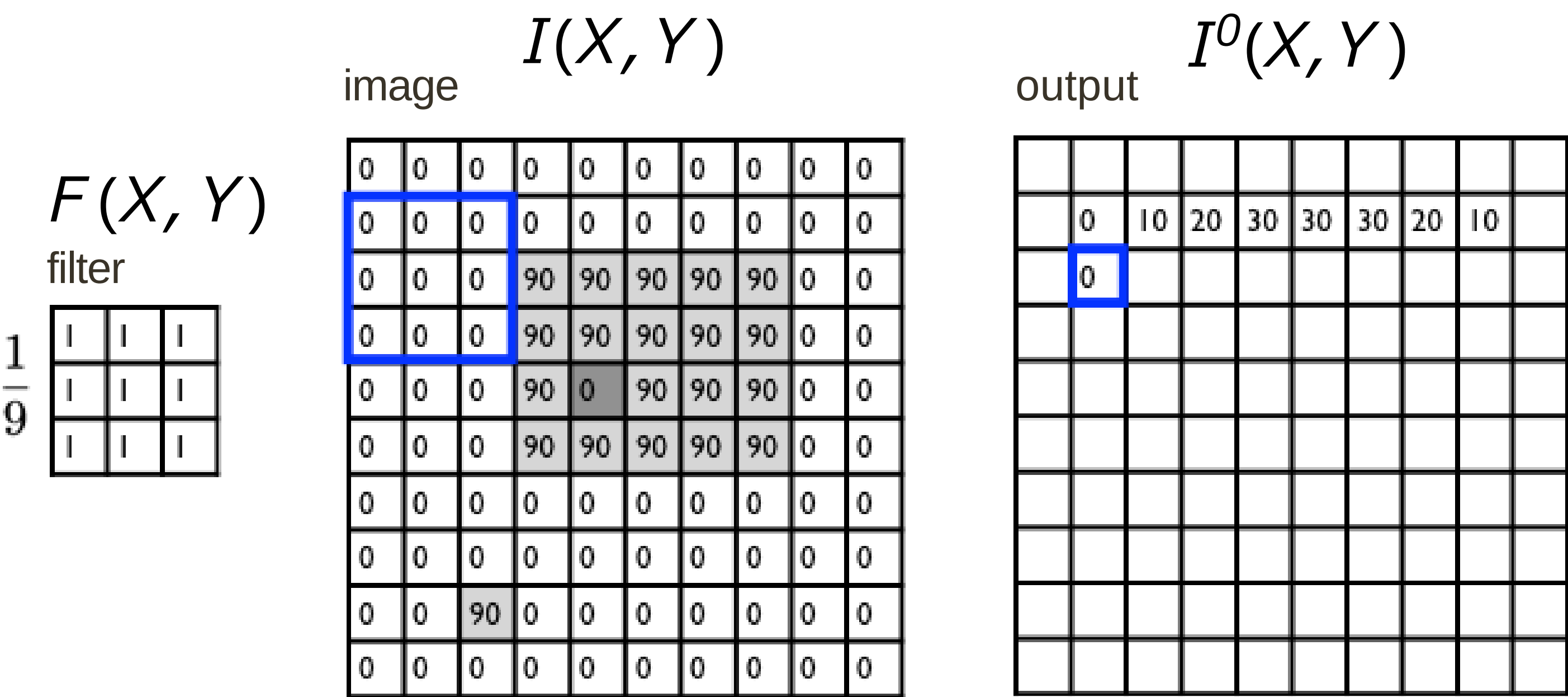
Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

$$= \sum_{j=-k}^k \sum_{i=-k}^k \underbrace{F(I, J)}_{\text{filtre}} \underbrace{I(X + i, Y + j)}_{\text{image (signal)}}$$

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

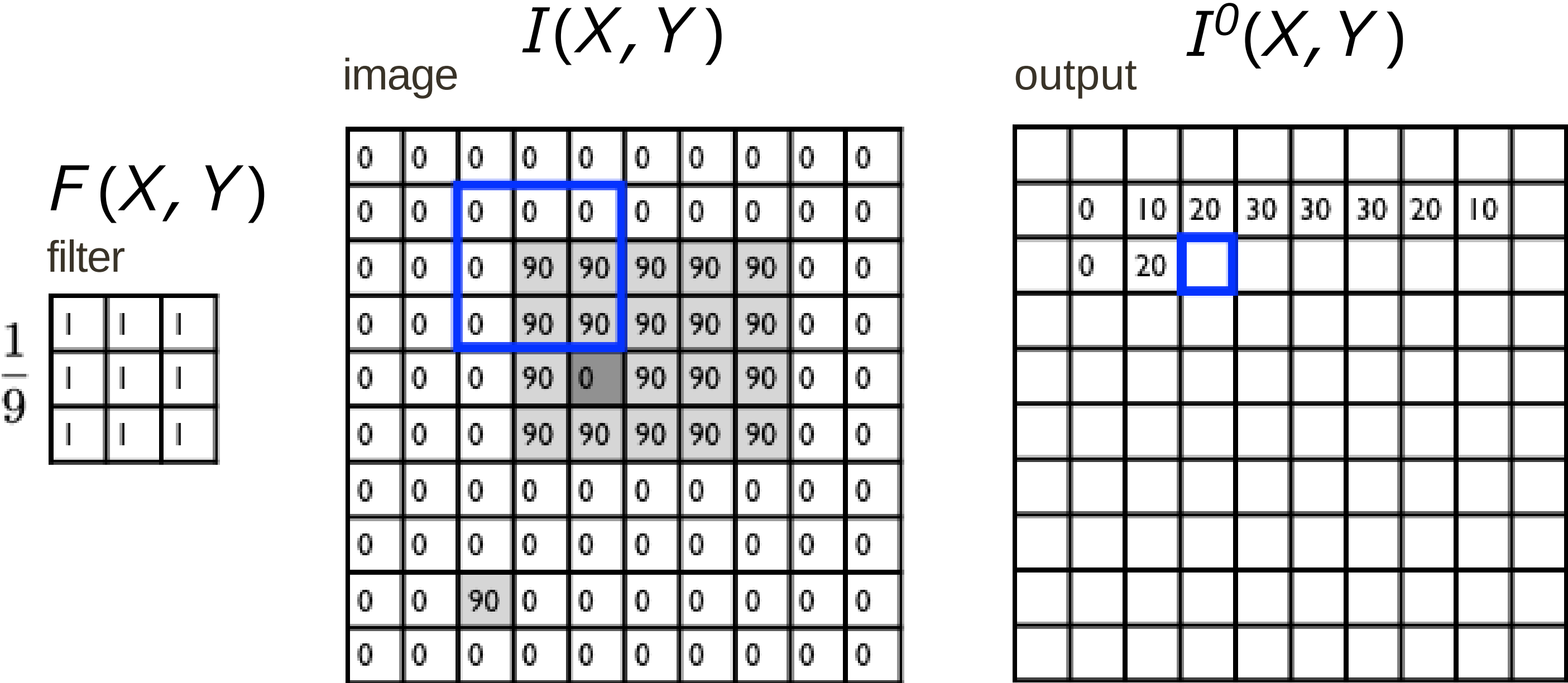
$=$

$\sum_{j = -k}^k \sum_{i = -k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$

sortie

$=$

$\sum_{j = -k}^k \sum_{i = -k}^k$

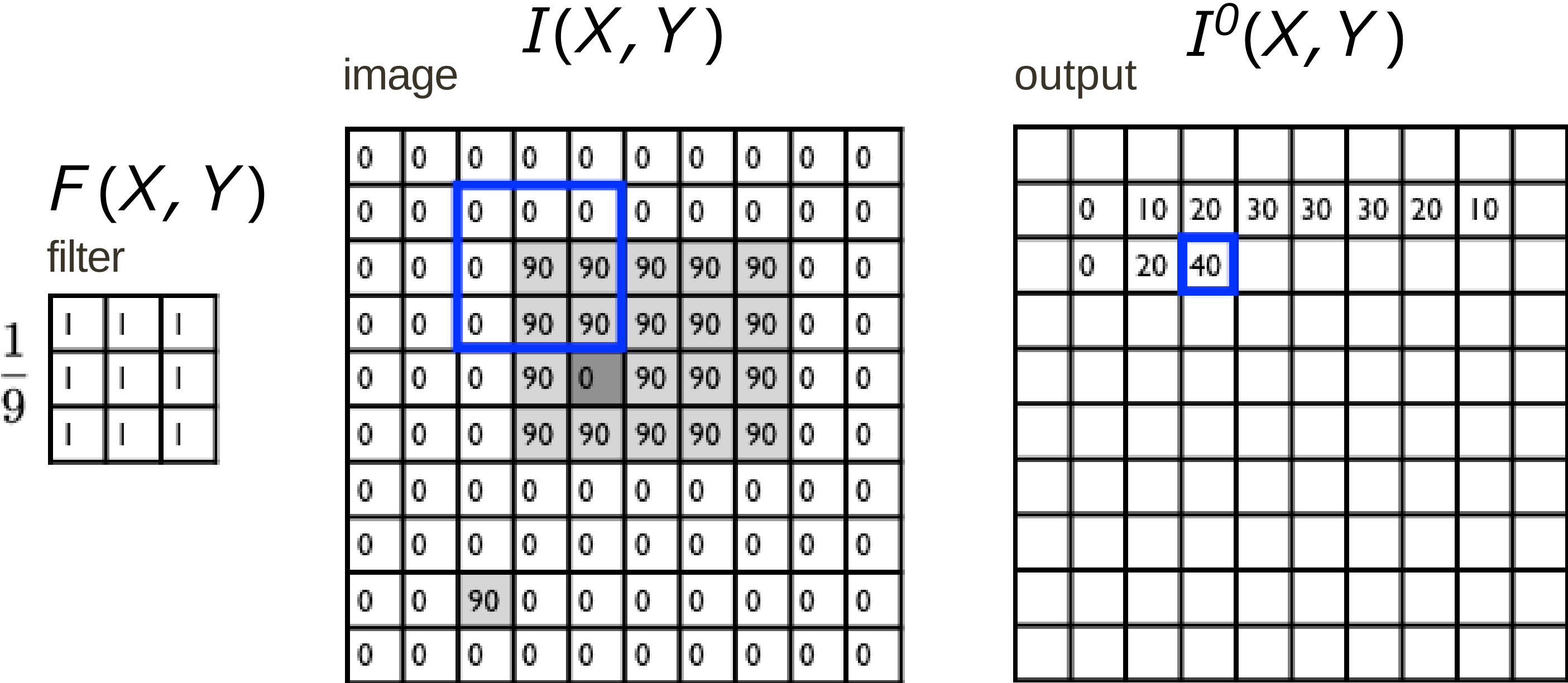
$F(I, J)$

filtre

$I(X + i, Y + j)$

image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

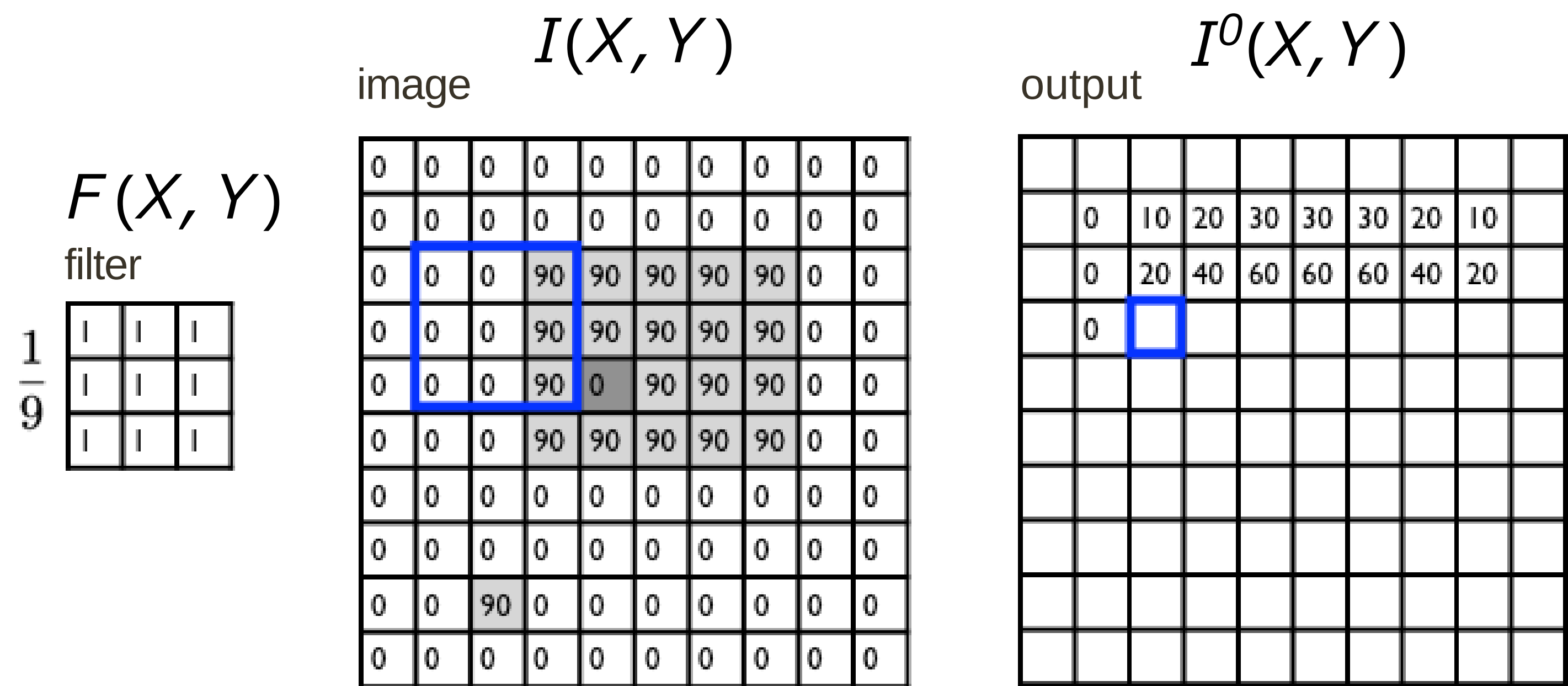
 $=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

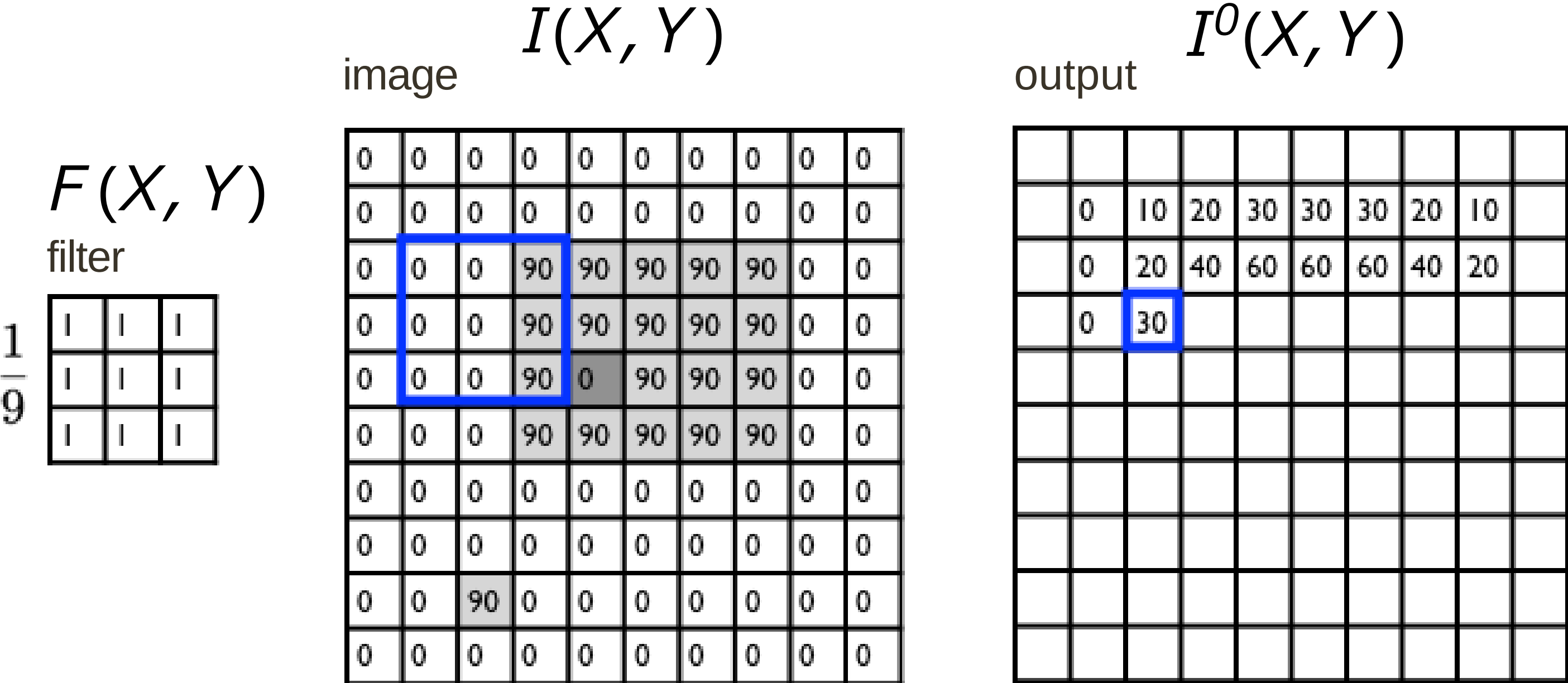
$=$

$\sum_{j = -k}^k \sum_{i = -k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

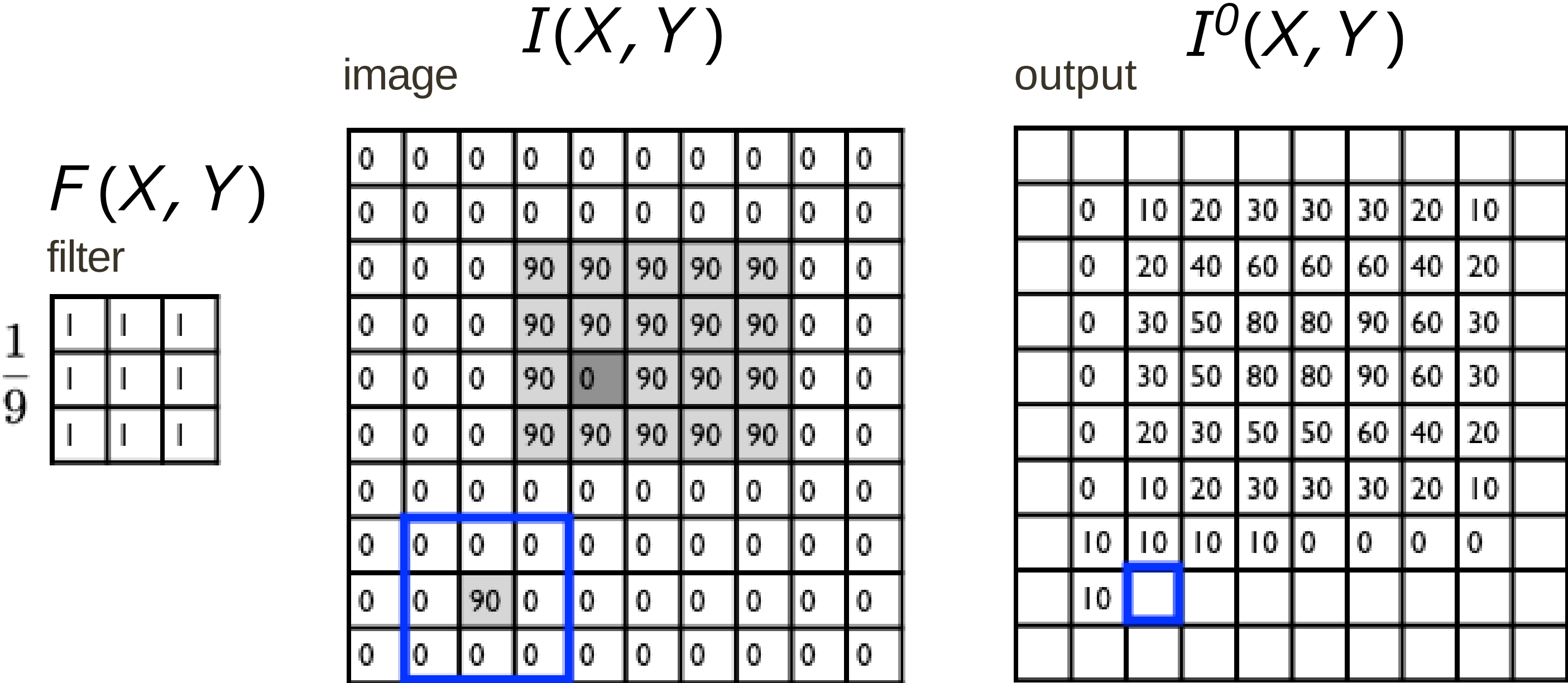
 $=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

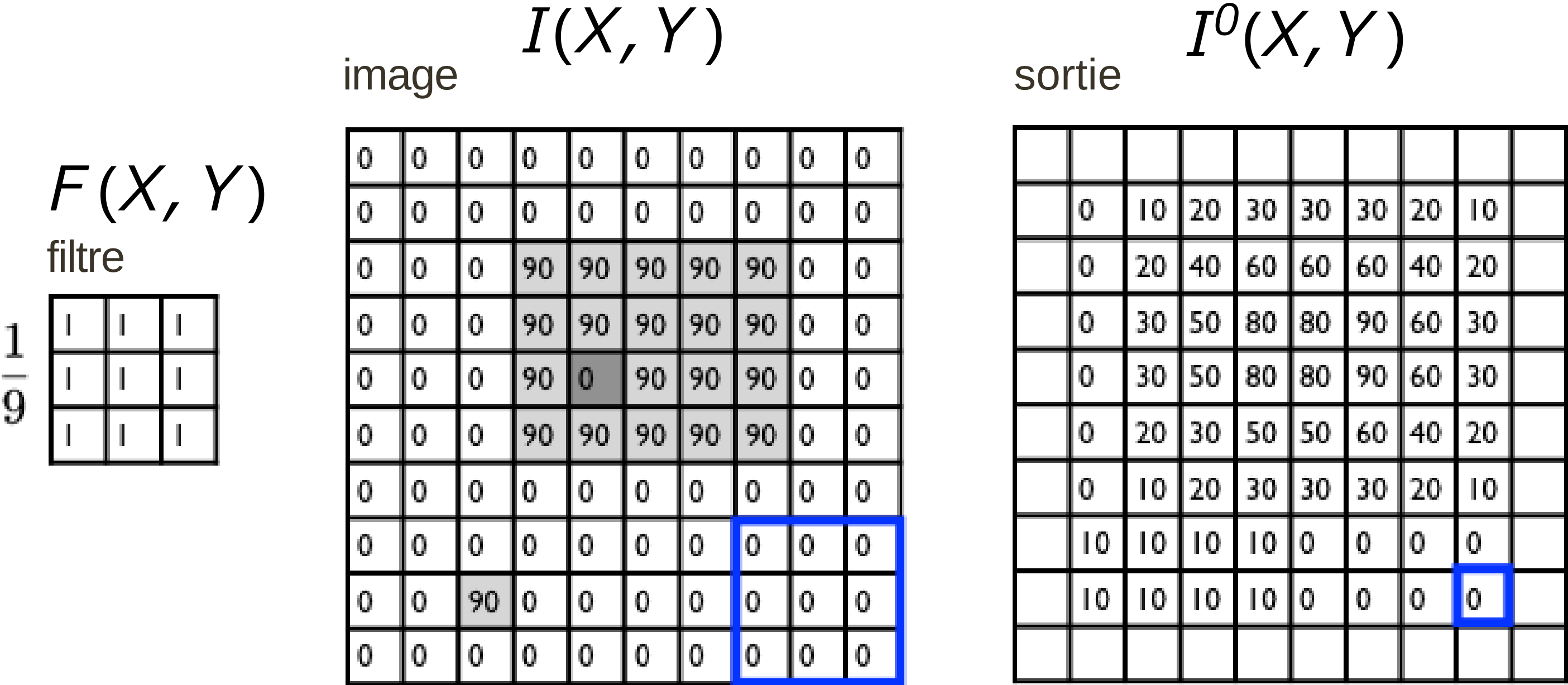
 $=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

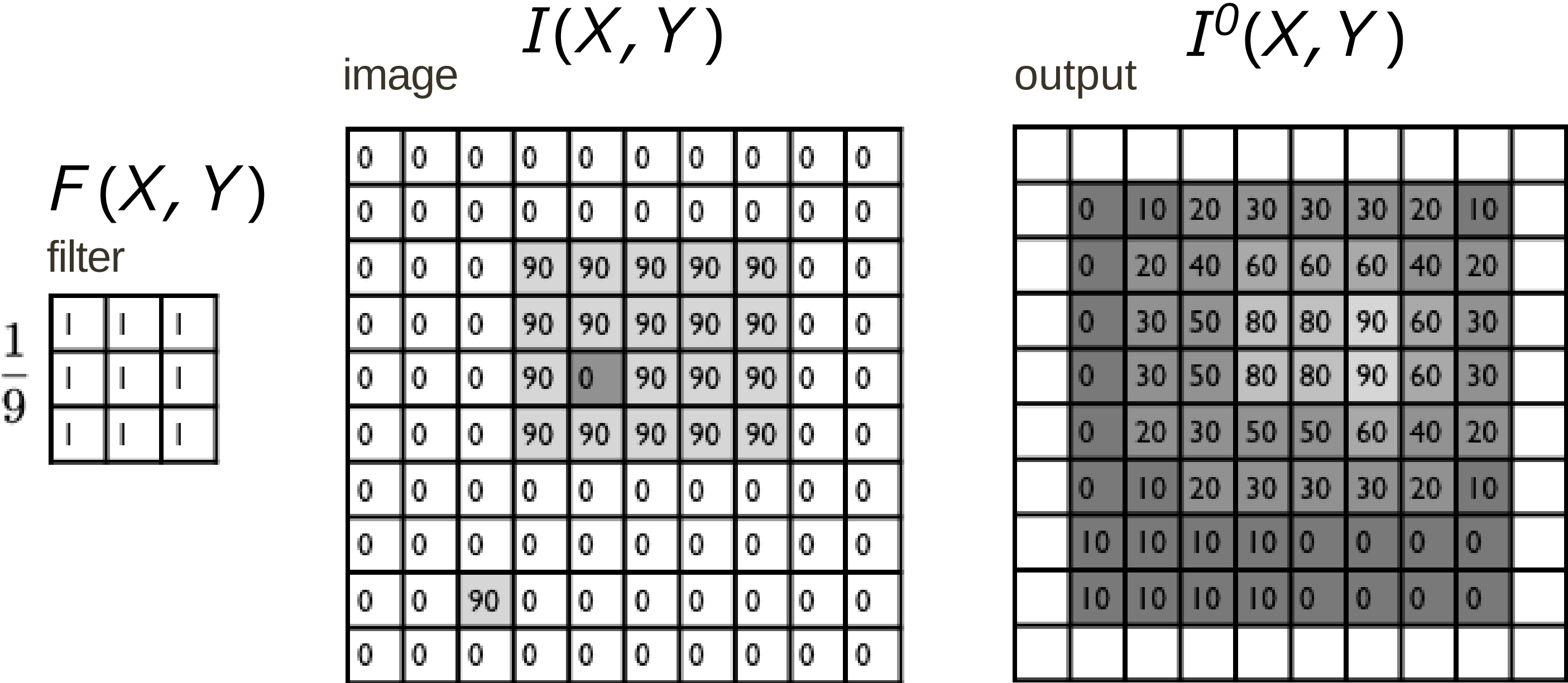
$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Exemple de Filtre Linéaire



$I^0(X, Y)$
sortie

$=$

$\sum_{j=-k}^k \sum_{i=-k}^k$

$F(I, J)$
filtre

$I(X + i, Y + j)$
image (signal)

Motivation: débruitage d'image

- Comment peut-on réduire le bruit dans une photo?



Moyenne mobile

- remplace chaque pixel par une moyenne pondérée de son voisinage
- Les poids sont appelés : *noyau du filtre*
- Quels sont les poids pour la moyenne d'un voisinage 3x3 ?

 $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

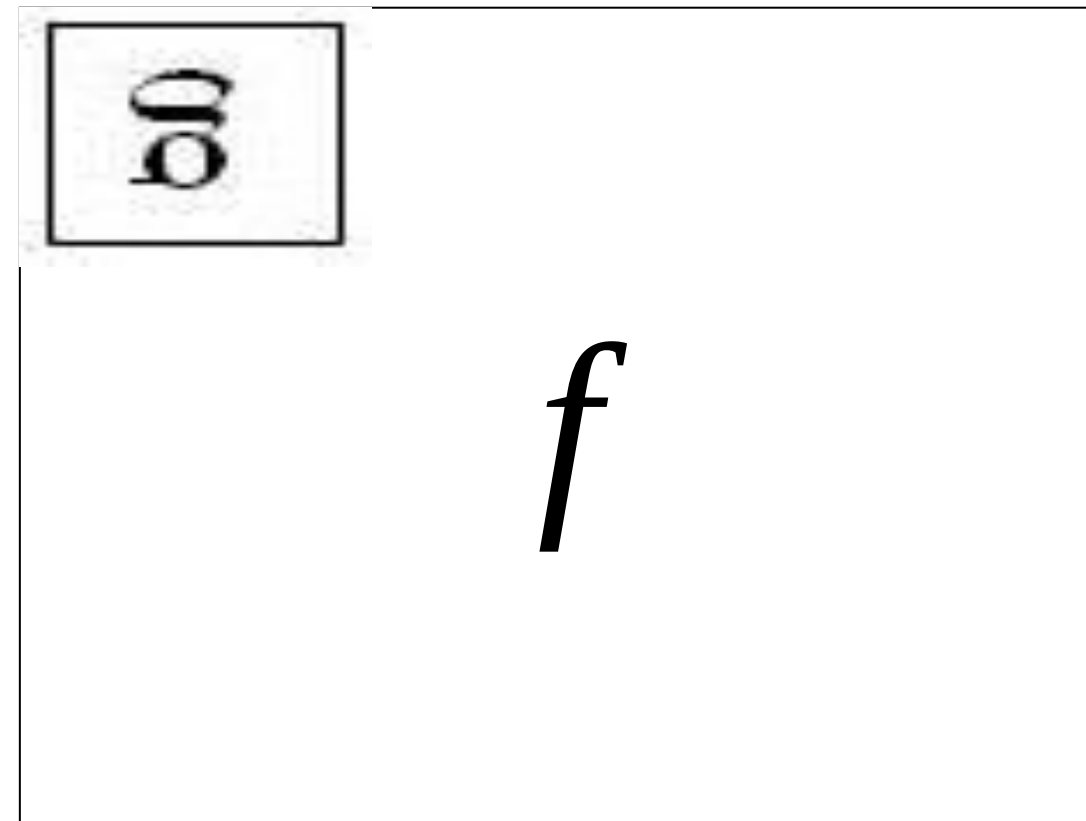
“box filter”

Définir la convolution

- Soit f une image et g un noyau. La sortie de f convoluée avec g est notée $f * g$.

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] g[k, l]$$

Convention:
noyau est “inversée”



- MATLAB functions: [conv2](#), [filter2](#), [imfilter](#)

Principales propriétés

- **Linéarité:** $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
- **Invariance spatiale:** ne dépend pas de la position du pixel :
 $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
- Résultat théorique : tout opérateur linéaire invariant peut être représentée par une convolution

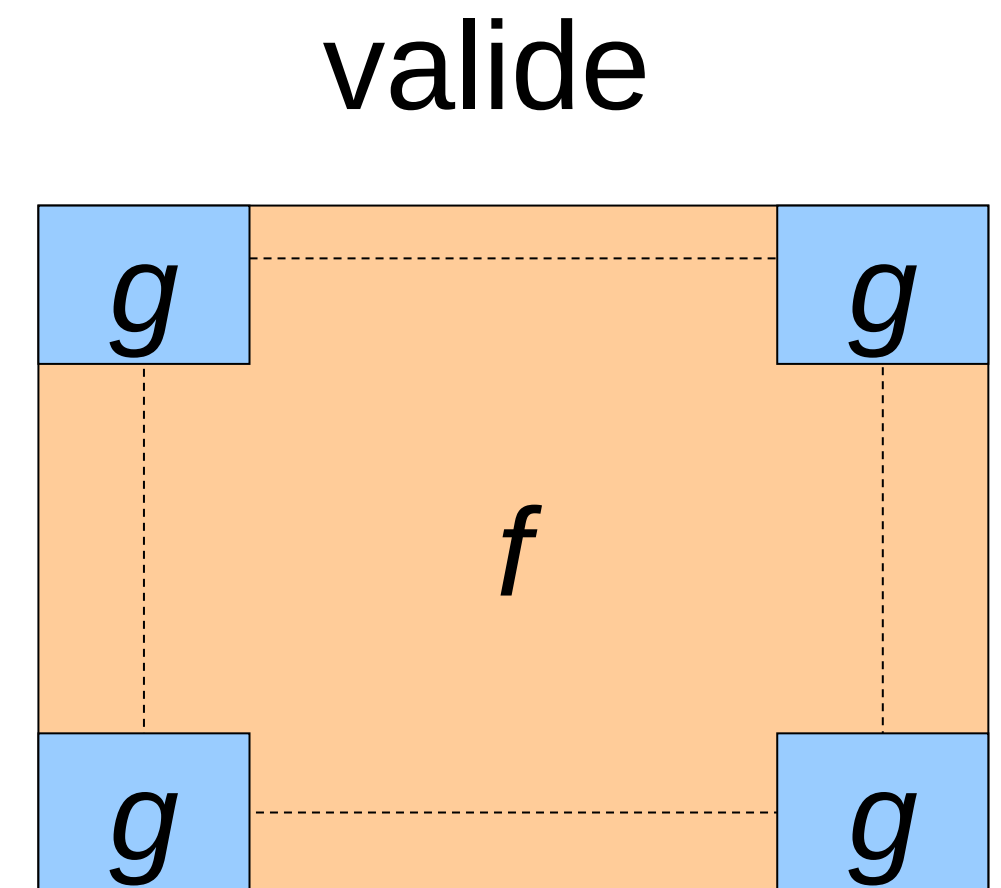
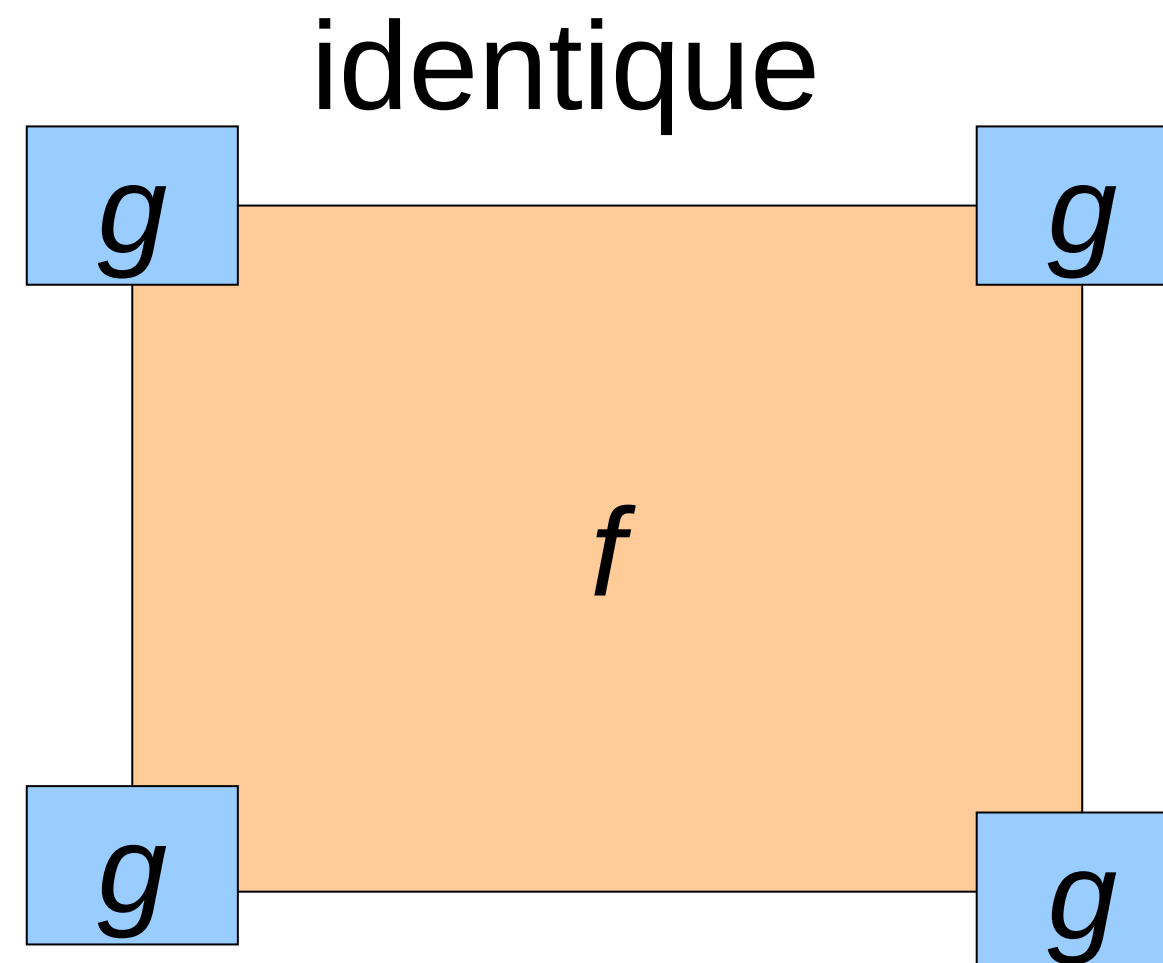
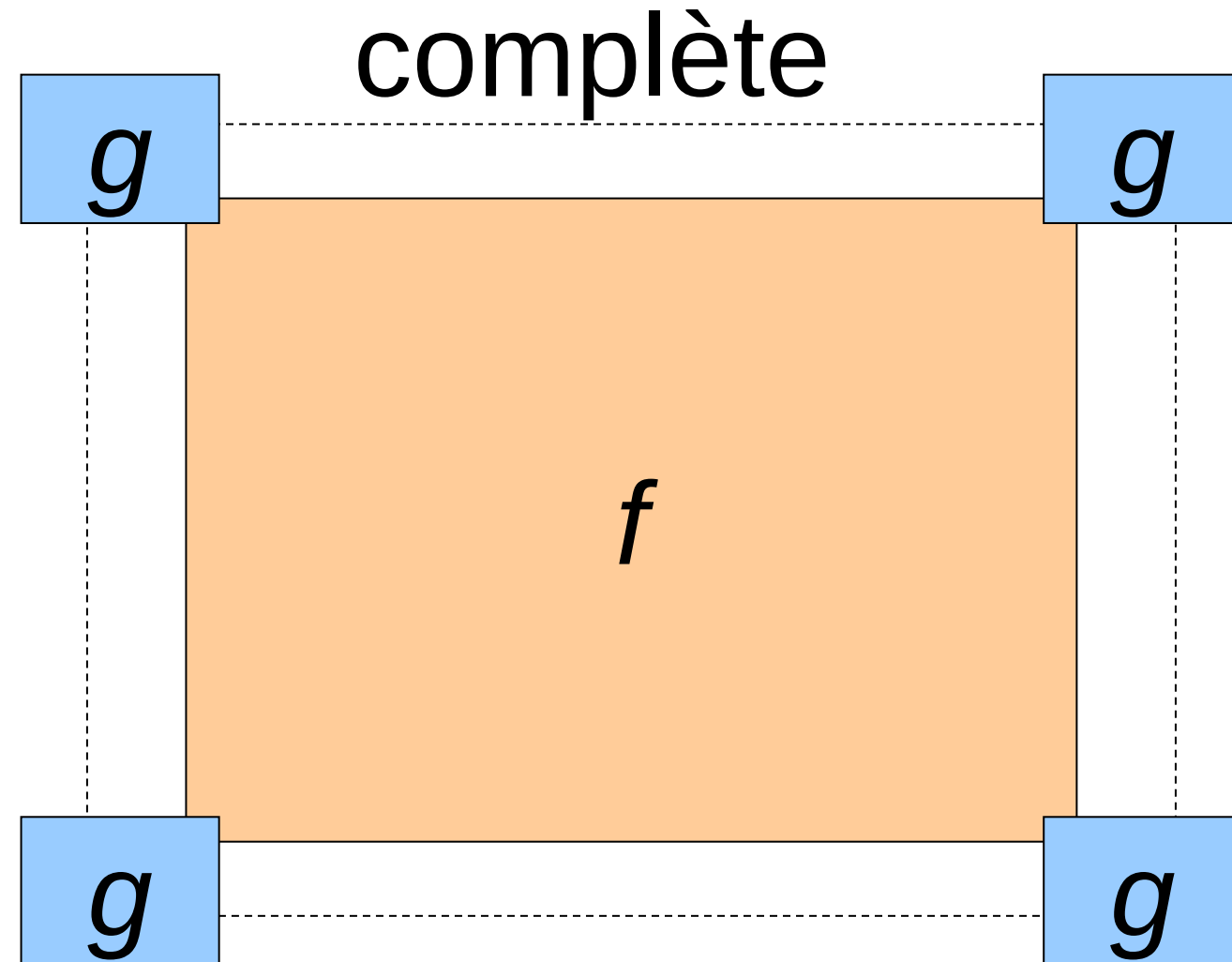
Propriétés

- Commutative: $a * b = b * a$
 - Pas de différence entre le filtre et le signal du point de vue concept
- Associative: $a * (b * c) = (a * b) * c$
 - Souvent plusieurs filtres appliqués l'un après l'autre: $((a * b_1) * b_2) * b_3$
 - Ceci équivalent à appliquer un seul filtre: $a * (b_1 * b_2 * b_3)$
- Distributive avec addition: $a * (b + c) = (a * b) + (a * c)$
- Multiplication par Scalaires : $ka * b = a * kb = k(a * b)$
- Identité: impulsion de Dirac $e = [..., 0, 0, 1, 0, 0, ...]$
 $a * e = a$

Détails de l'effet de bord

Quelle est la taille de l'image de sortie?

- MATLAB: `filter2(g, f, shape)`
 - *shape* = 'full': output size is sum of sizes of f and g
 - *shape* = 'same': output size is same as f
 - *shape* = 'valid': output size is difference of sizes of f and g



Détails Ennuyeux

Les effets de bordure?

- La fenêtre du filtre dépasse le bord de l'image
- besoin d'extrapoler
- méthodes:
 - couper les bords (clip filter)
 - replier les deux bords
 - copier le bord
 - réflexion du bord



Effets de bord sous Matlab

Les effets de bord?

- La fenêtre du filtre dépasse le bord de l'image
- besoin d'extrapoler
- méthodes:
 - clip filter (black): `imfilter(f, g, 0)`
 - replier les deux bords: `imfilter(f, g, 'circular')`
 - copier le bord: `imfilter(f, g, 'replicate')`
 - réflexion du bord: `imfilter(f, g, 'symmetric')`

Pratique avec les filtres linéaires



Originale

0	0	0
0	1	0
0	0	0

?

Pratique avec les filtres linéaires



Originale

0	0	0
0	1	0
0	0	0



Filtrée (pas de changement)

Pratique avec les filtres linéaires



Originale

0	0	0
0	0	1
0	0	0

?

Pratique avec les filtres linéaires



Originale

0	0	0
0	0	1
0	0	0



Décalée vers la gauche
par 1 pixel

Pratique avec les filtres linéaires



Originale

 $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

?

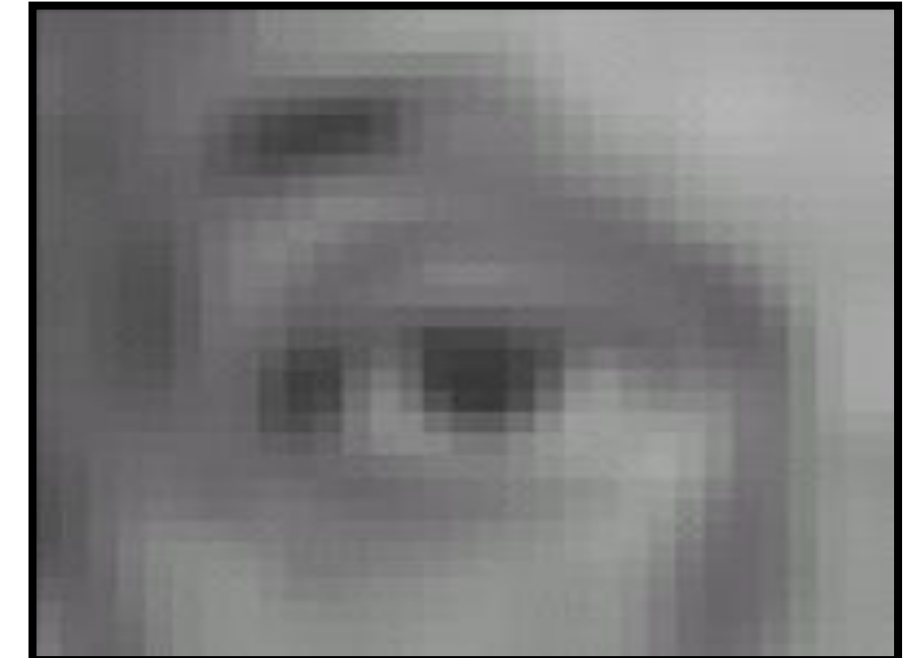
Pratique avec les filtres linéaires



Originale

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



Floue (avec un box filtre)

Pratique avec les filtres linéaires



Originale

0	0	0
0	2	0
0	0	0

—

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

?

(Noter que la somme des poids vaut 1)

Pratique avec les filtres linéaires



Originale

0	0	0
0	2	0
0	0	0

−

$\frac{1}{9}$

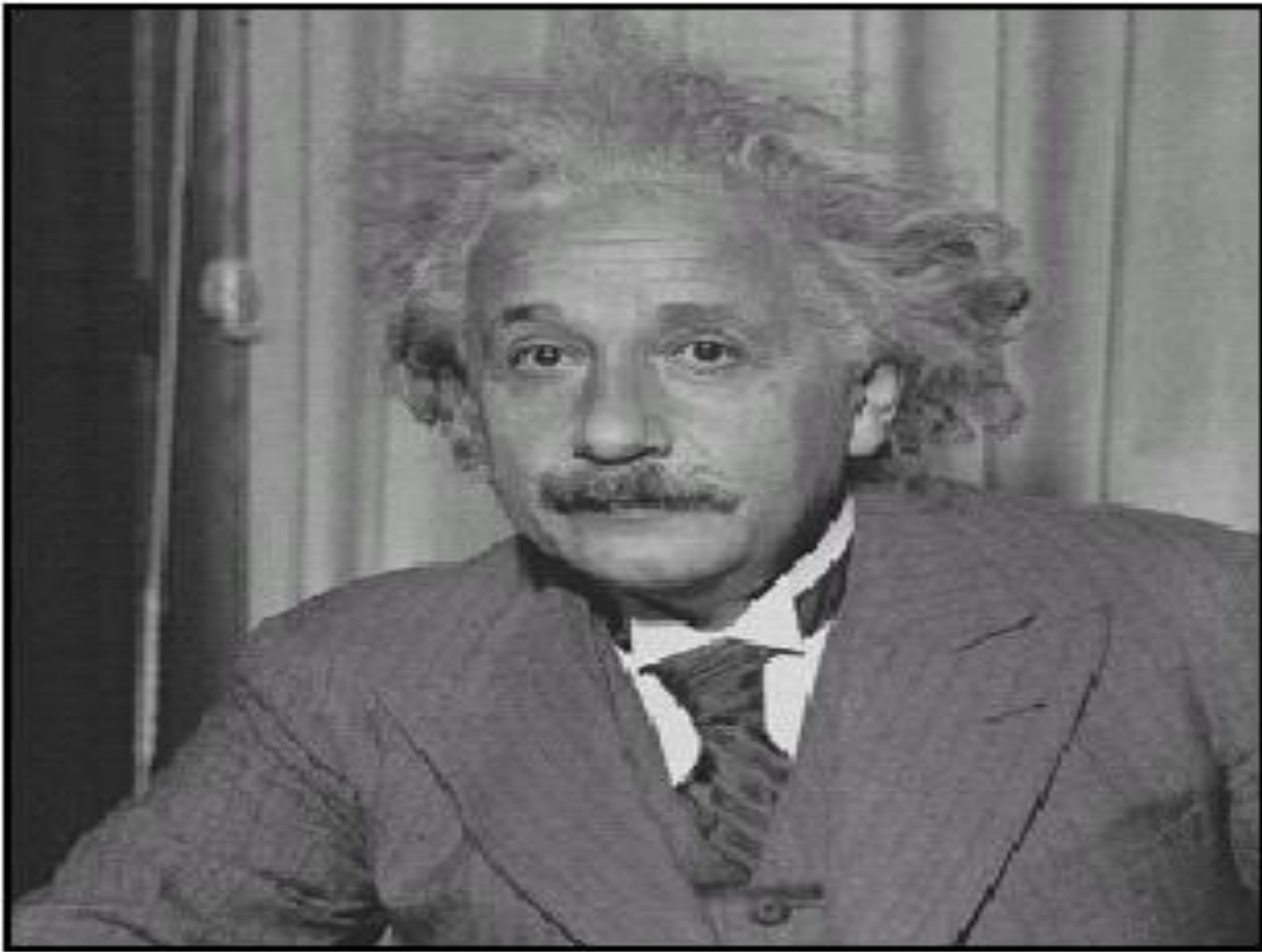
1	1	1
1	1	1
1	1	1



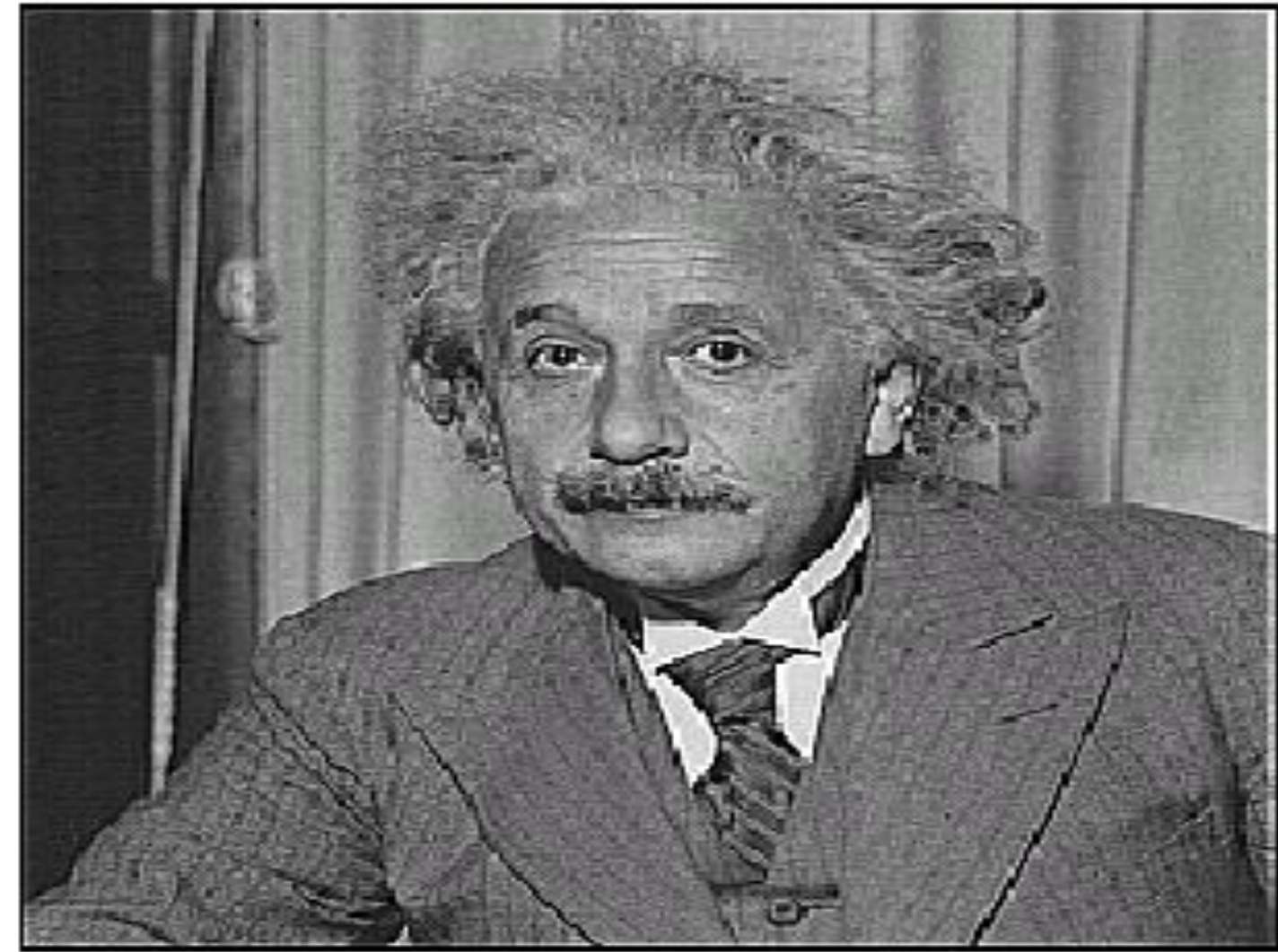
Filtre d'accentuation

- Accentuer les différences avec la moyenne locale

Accentuation (Sharpening)



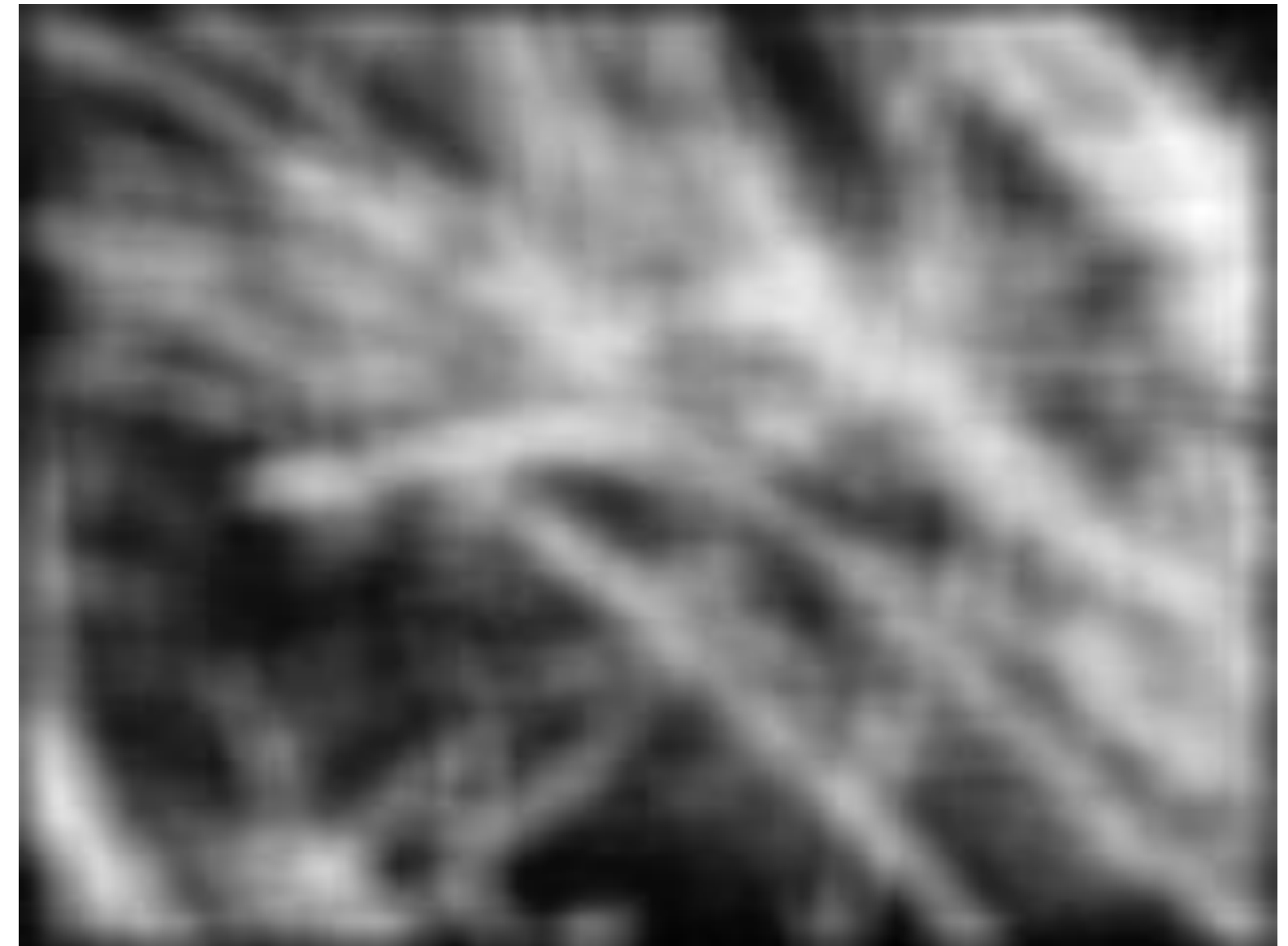
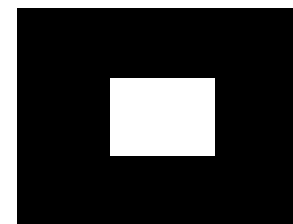
Avant



Après

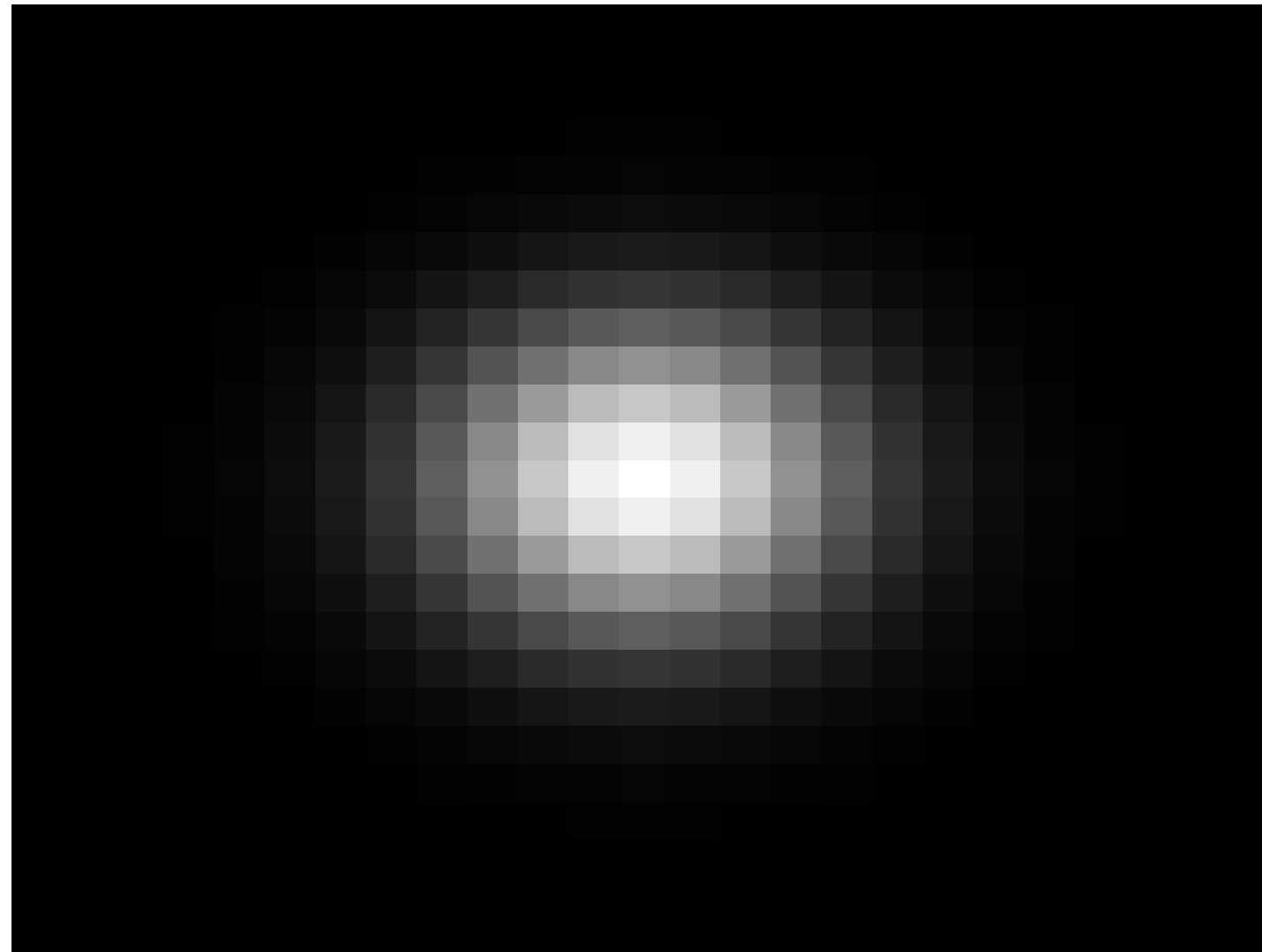
Lissage avec le filtre boîte (box filter)

- Effet de bord dans cette photo?
- Quelle est la solution?



Lissage avec le filtre boîte (box filter)

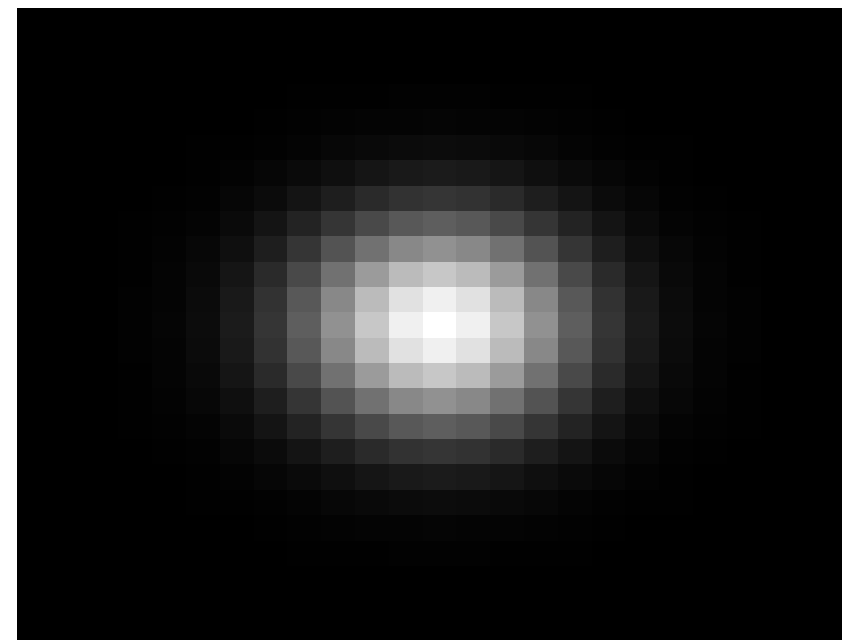
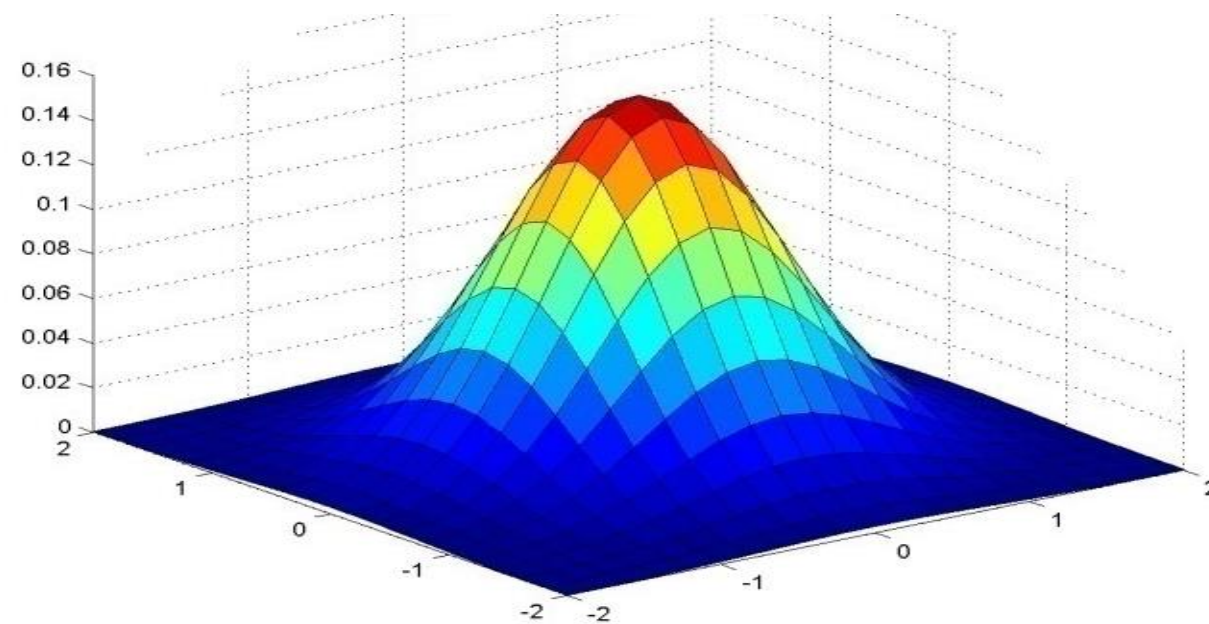
- Pour éliminer les effets du bord, pondérer la contribution des pixels voisins selon leur proximité du pixel central



“bosse flou”

Filtre ou Noyau Gaussien

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



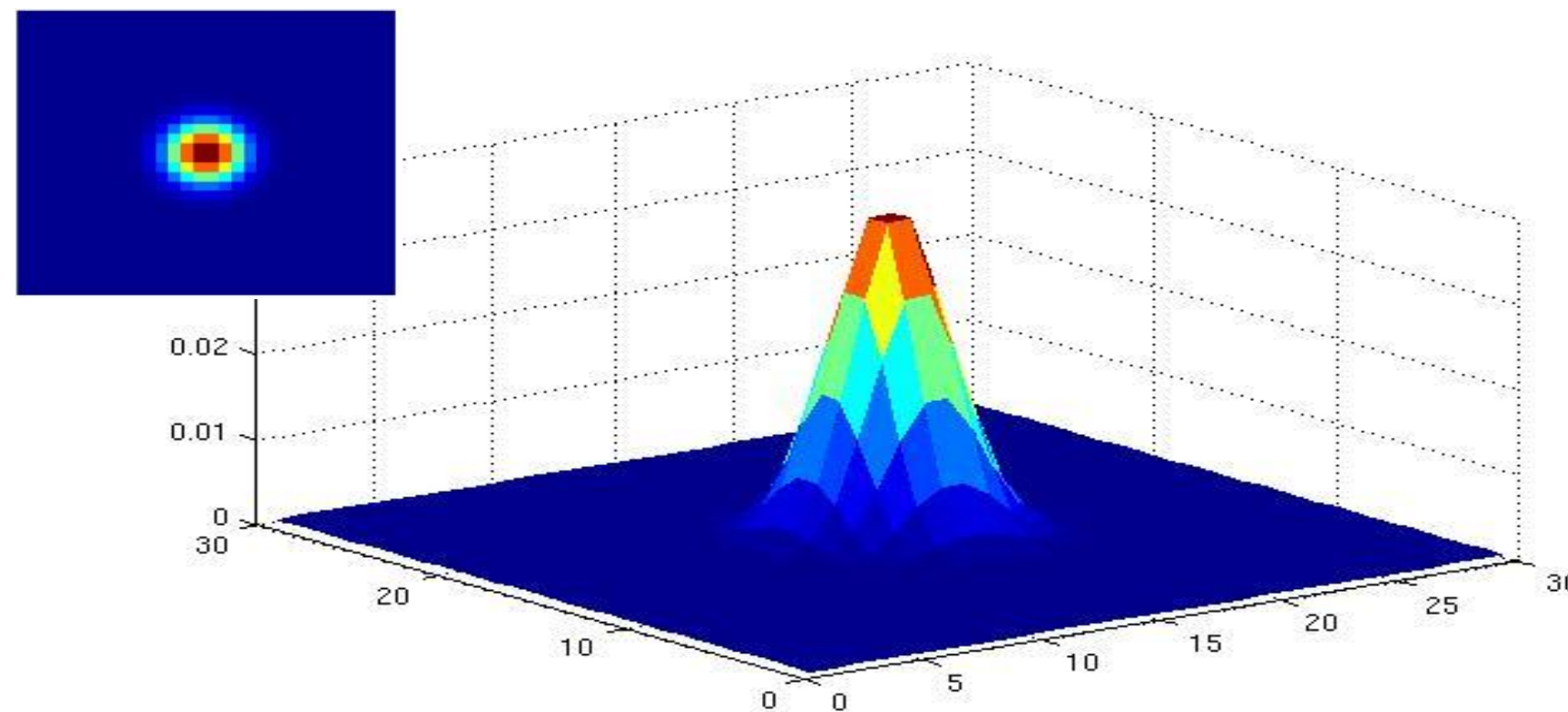
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

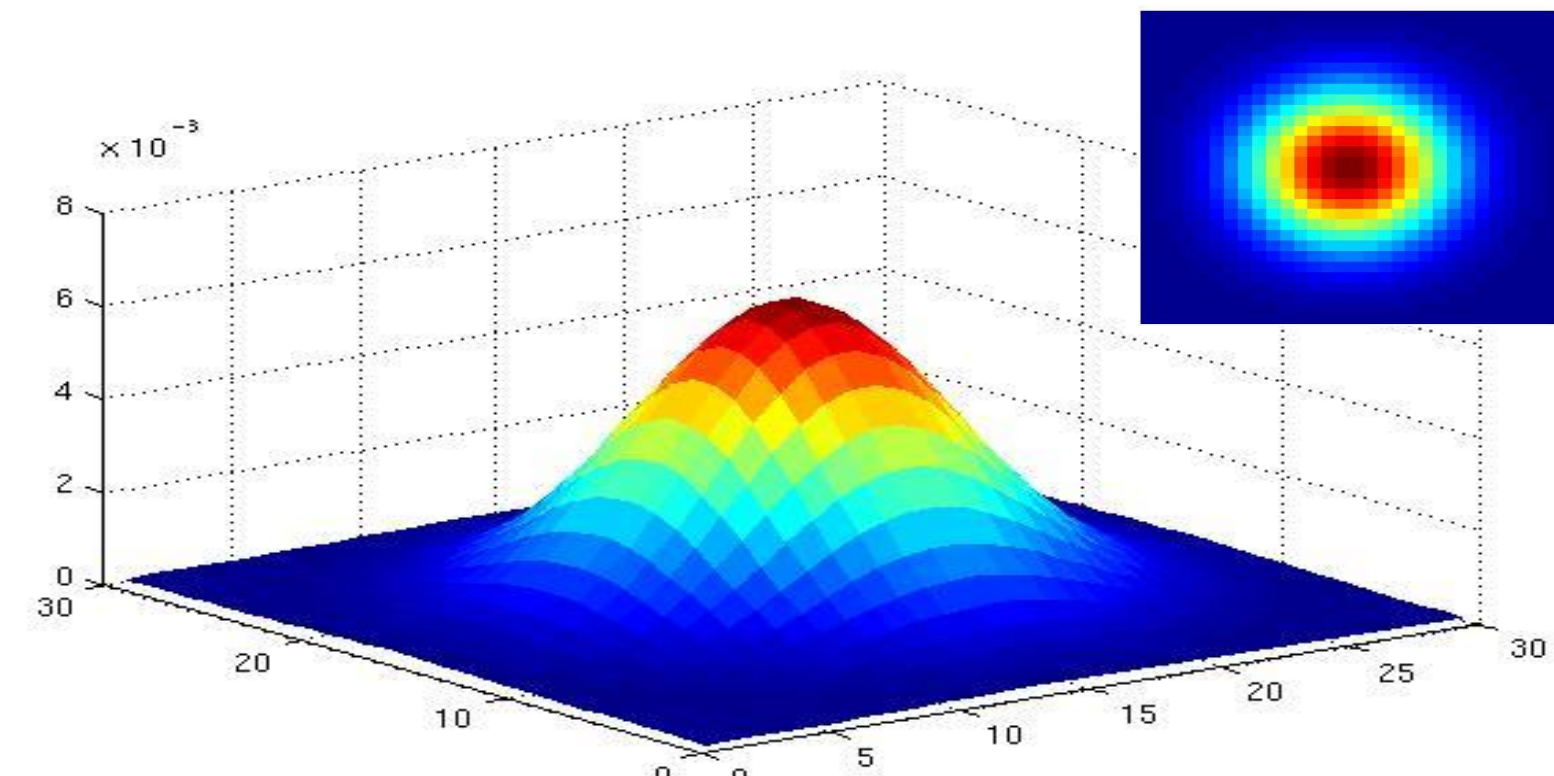
- Facteur constant fait que le volume vaut 1 (à ignorer lors du calcul des valeurs du filtre , puisque il faut renormaliser les poids de façon que leur somme vaut 1)

Filtre ou Noyau Gaussien

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



$\sigma = 2$ avec noyau de
30 x 30

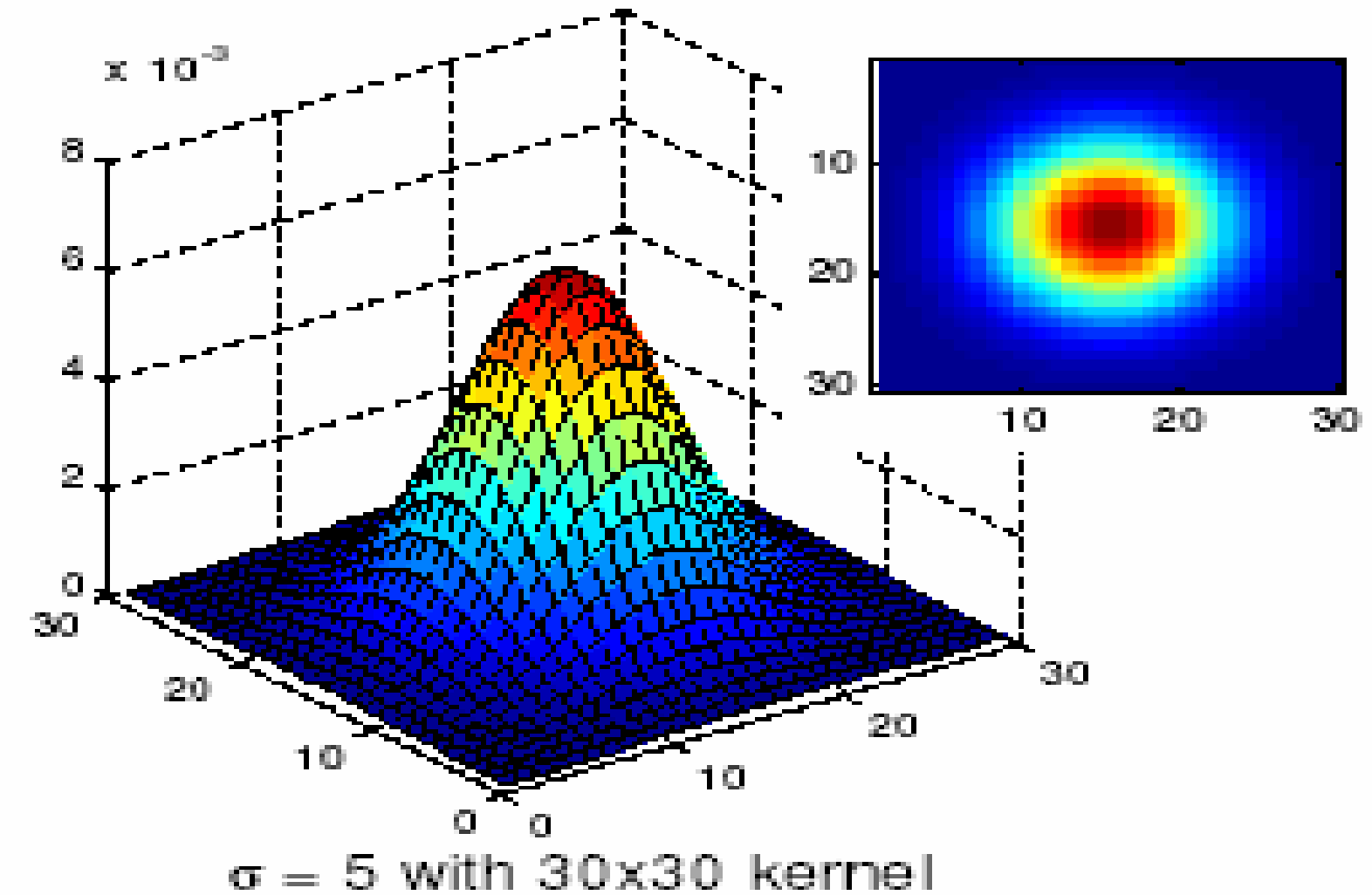
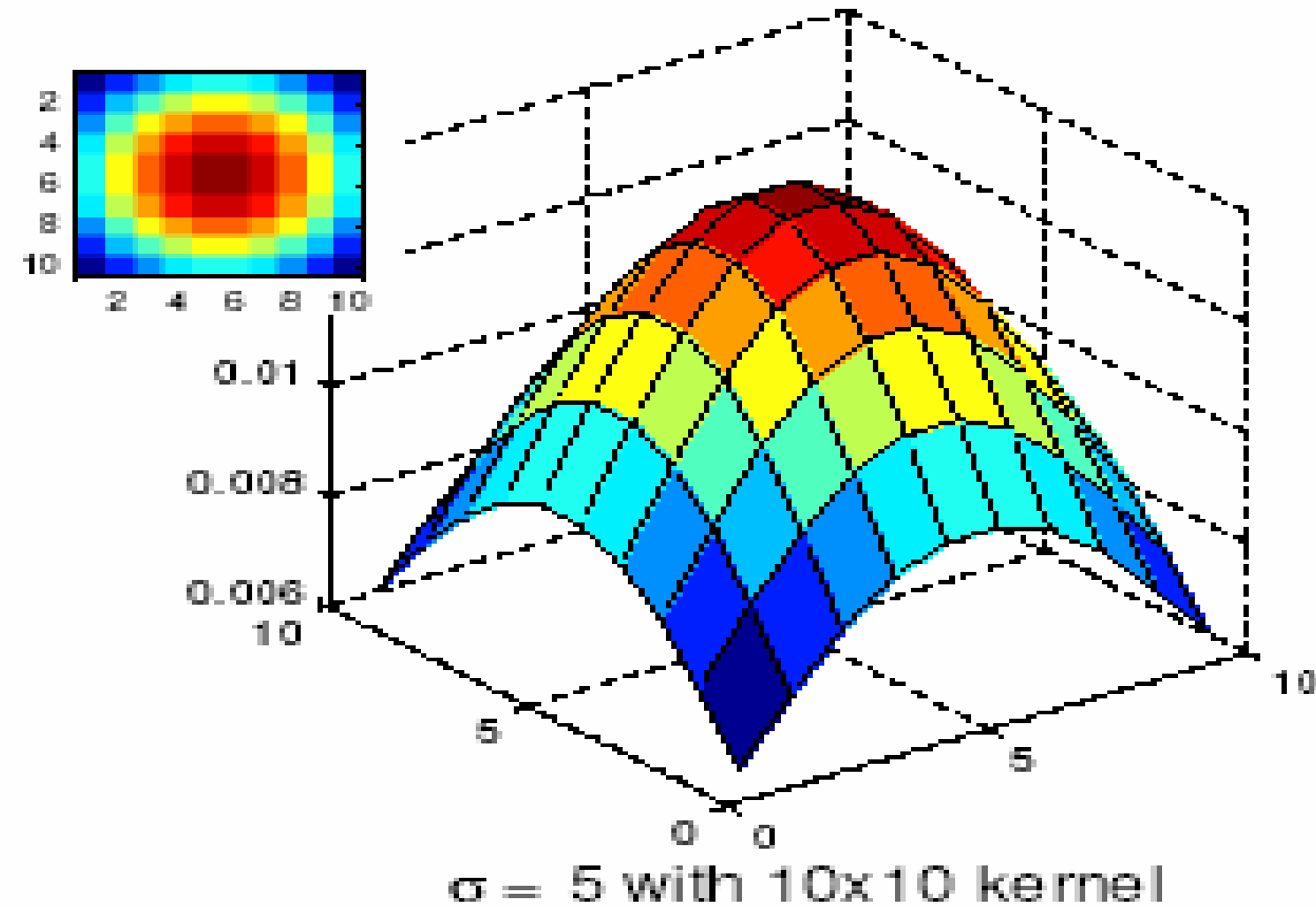


$\sigma = 5$ avec noyau de 30
x 30

- Ecart-type σ : determine l'étendue du lissage

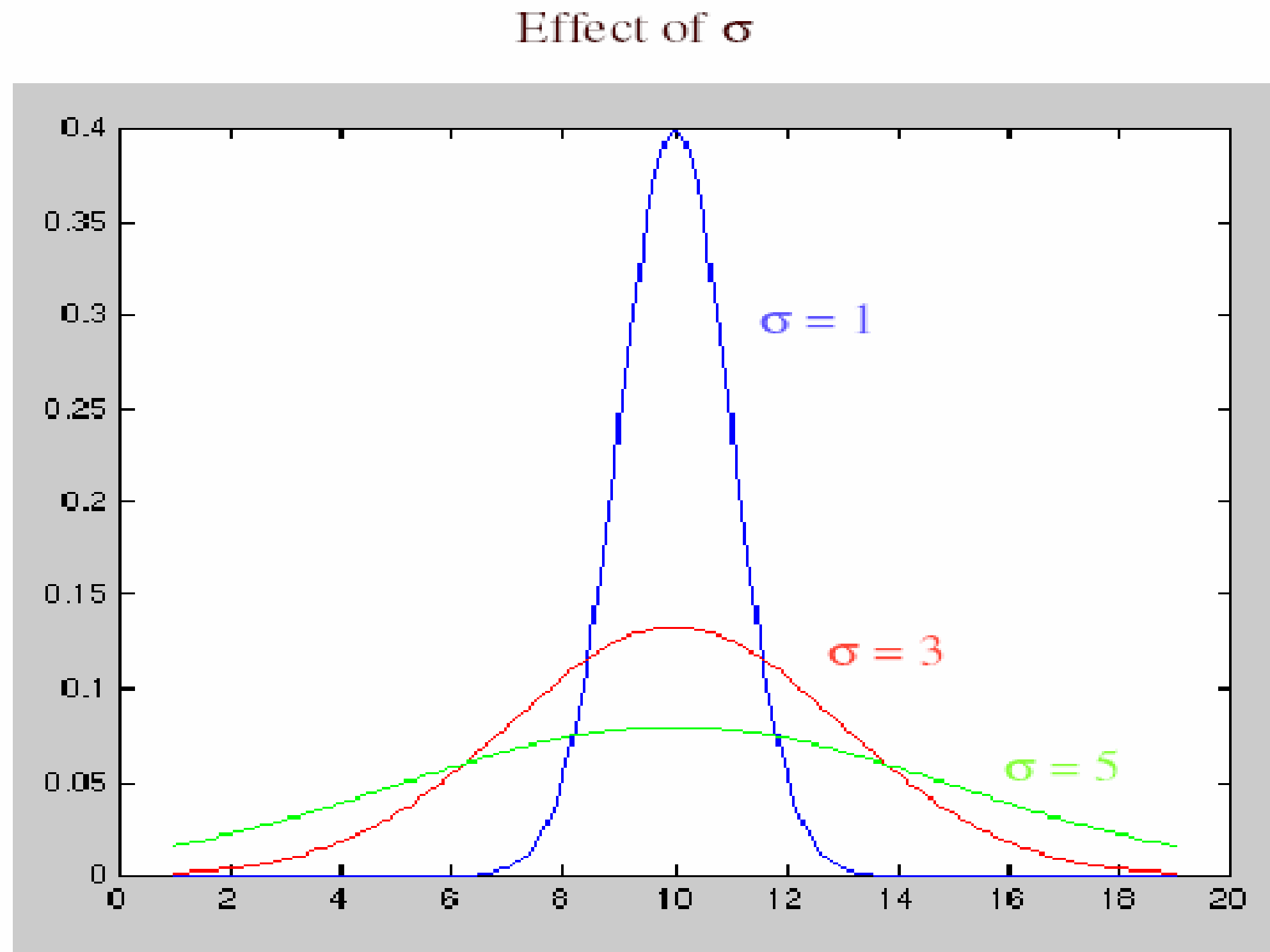
Choix de la taille du noyau

- La fonction Gaussienne est à support infini, mais les filtres discrets utilise des noyaux finis

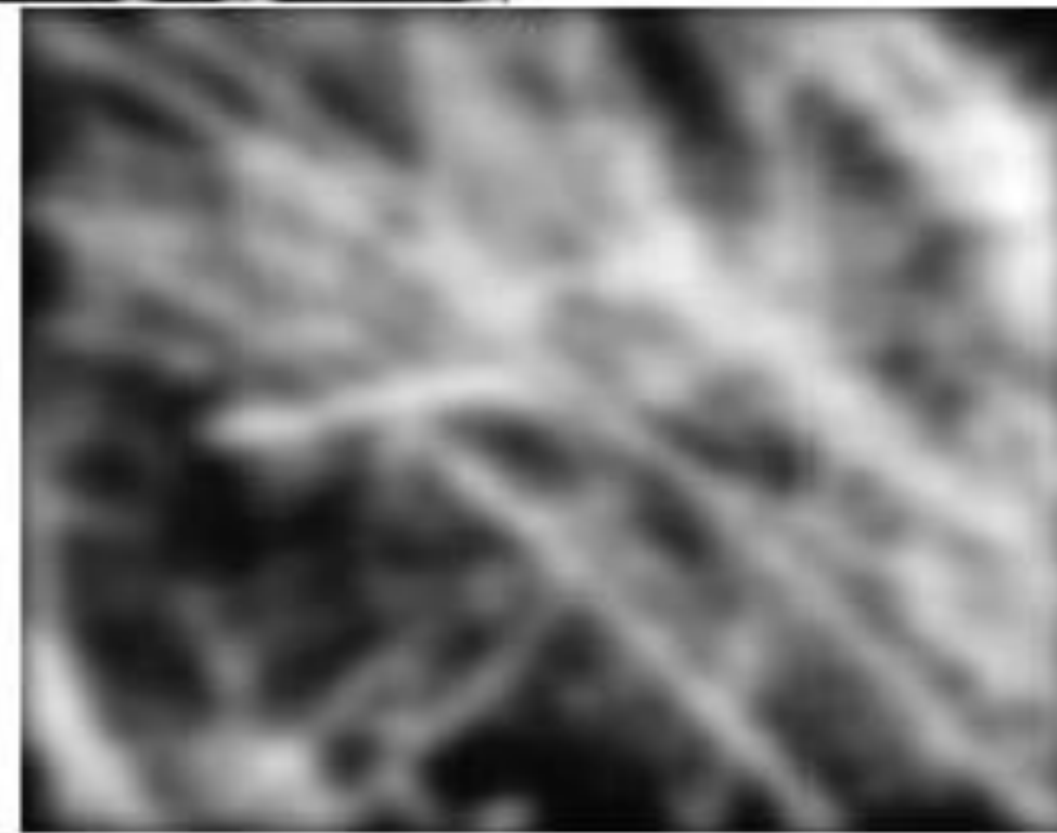


Choix de la taille du noyau

- Règle: mettre la demi-largeur du filtre égale à 3σ



Gaussien vs. filtrage par 'box'



Filtres Gaussiens

- Elimine les composantes “hautes-fréquences” de l’image (filtre passe-bas)
- Convoluer avec lui-même est un autre Gaussien
 - Ainsi on peut lisser avec des noyaux à petit- σ , répéter, et obtenir le même résultat d’un noyau à large- σ ,
 - Convoluer 2 fois avec un noyau Gaussien avec ecart-type. σ est équivalent à convoluer avec 1 noyau avec ecart-type $\sigma\sqrt{2}$
- *Noyau Separable* : Equivalent à un produit de 2 Gaussiens 1D

Séparabilité du filtre Gaussien

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

Gaussien 2D est exprimé sous forme de produit de 2 Gaussiens 1D , une fonction de x et l'autre de y

Exemple de Séparabilité

convolution 2D
(pixel central)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix}$$

Le filtre factorisé
en produit de
filters 1D:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Appliquer convolution
suivant les lignes:

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

suivi par une convolution
selon la colonne:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix} = \begin{bmatrix} & & \\ & 65 & \\ & & \end{bmatrix}$$

Pourquoi la séparabilité est utile?

- Quelle est la complexité pour filtrer une image $n \times n$ avec un noyau $m \times m$?
 - $O(n^2 m^2)$
- Par contre si le noyau est séparable?
 - $O(n^2 m)$

Bruit



Original



Salt and pepper noise



Impulse noise



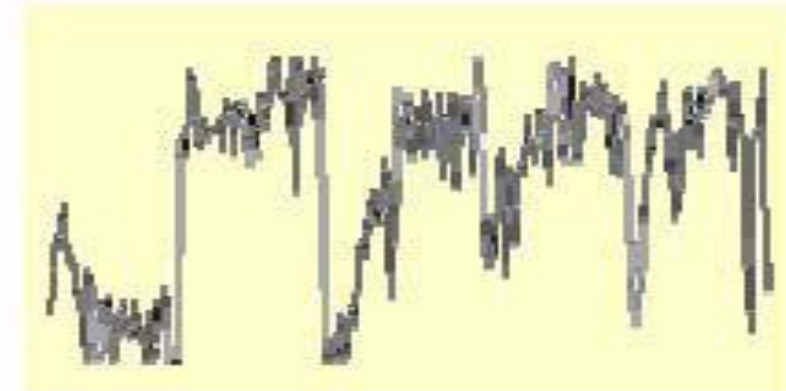
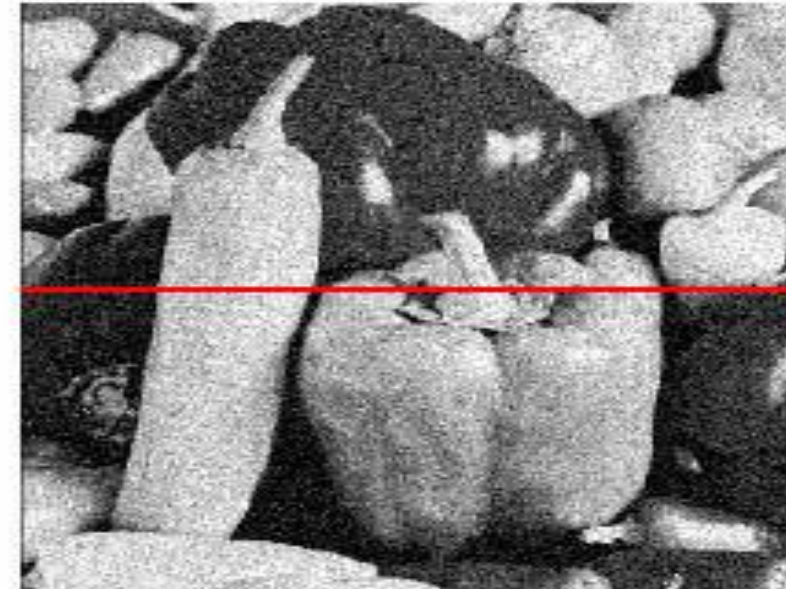
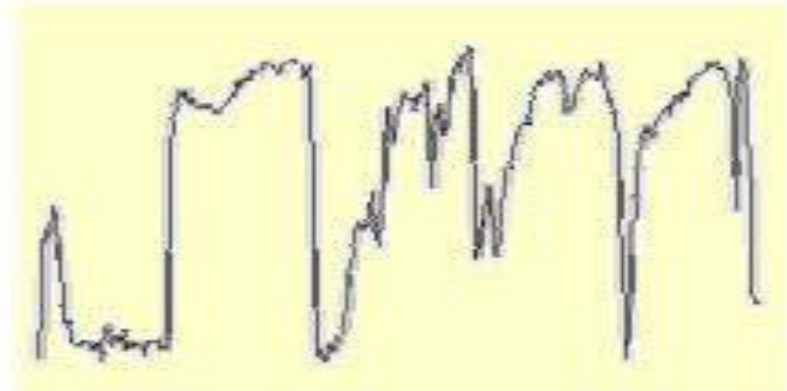
Gaussian noise

- **Bruit Salt and pepper :** contient des pixels blancs et noirs
- **Bruit Impulsion:** contient des pixels blancs aléatoires
- **Bruit Gaussien:** variations de l'intensité selon une distribution Gaussienne normale

Bruit Gaussien

- Modèle Mathématique : somme de plusieurs facteurs indépendants
- Bon pour des écart-types faibles
- Hypothèse: indépendant, bruit moyenne nulle

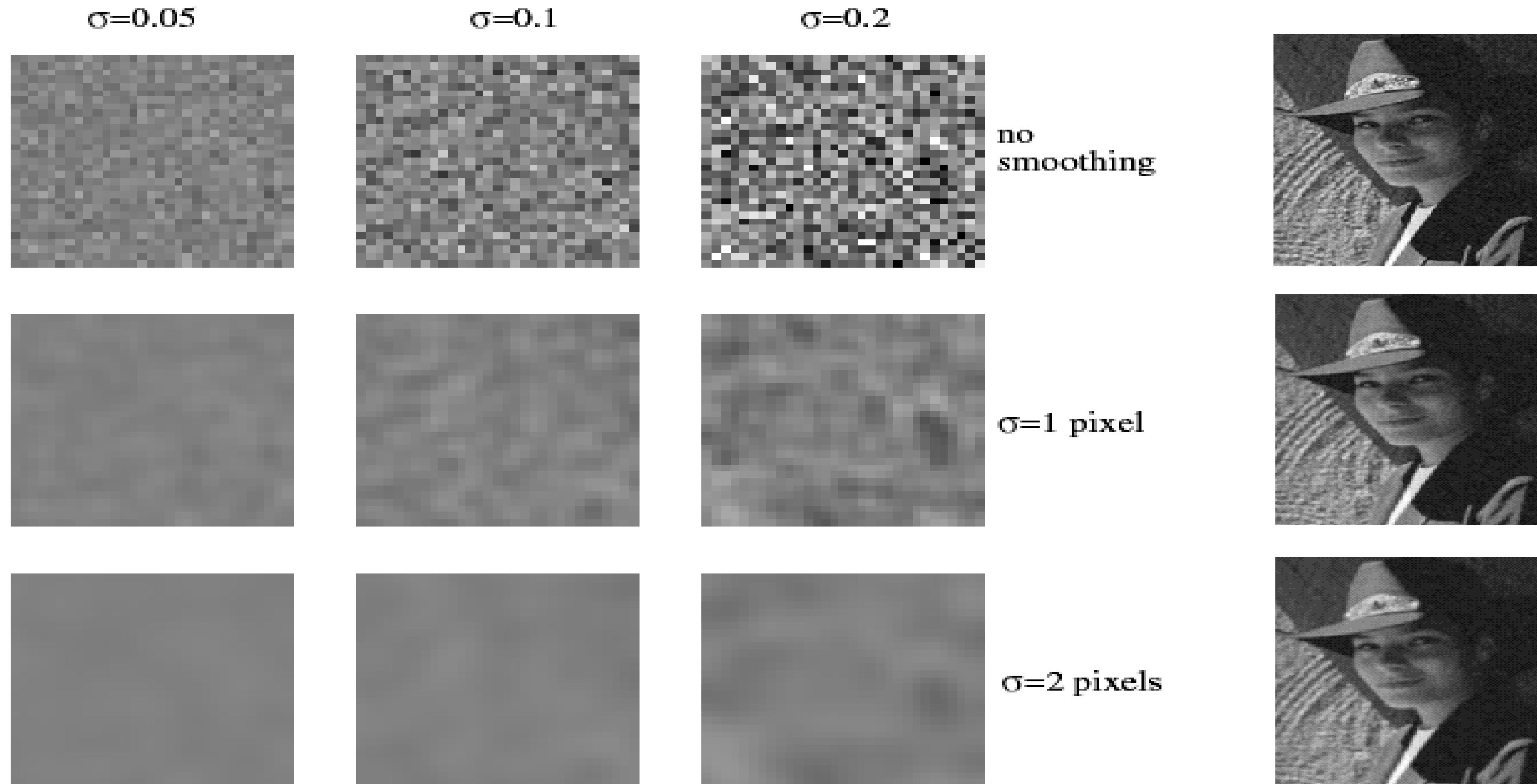
Image
Noise



$$f(x, y) = \overbrace{\hat{f}(x, y)}^{\text{Ideal Image}} + \overbrace{\eta(x, y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
 $\eta(x, y) \sim \mathcal{N}(\mu, \sigma)$

Réduire le bruit Gaussien



Lissage avec des écart-types plus larges supprime le bruit, mais aussi rend l'image plus floue

Réduire le bruit salt-and-pepper

3x3



5x5



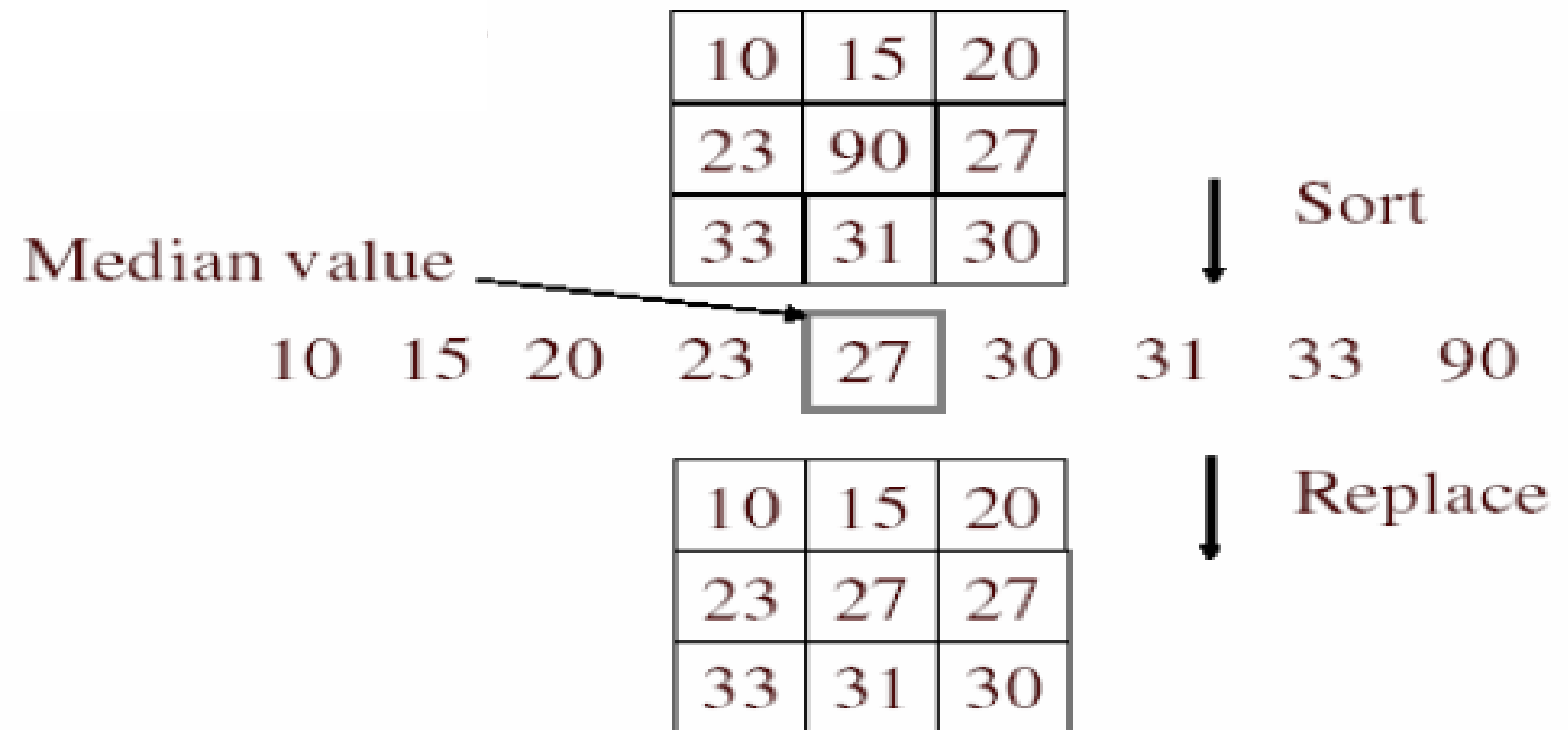
7x7



Comparer les résultats?

Une solution alternative : filtrage Médian

- Un **filtre median** opère sur une fenêtre en sélectionnant l'intensité médiane dans la fenêtre

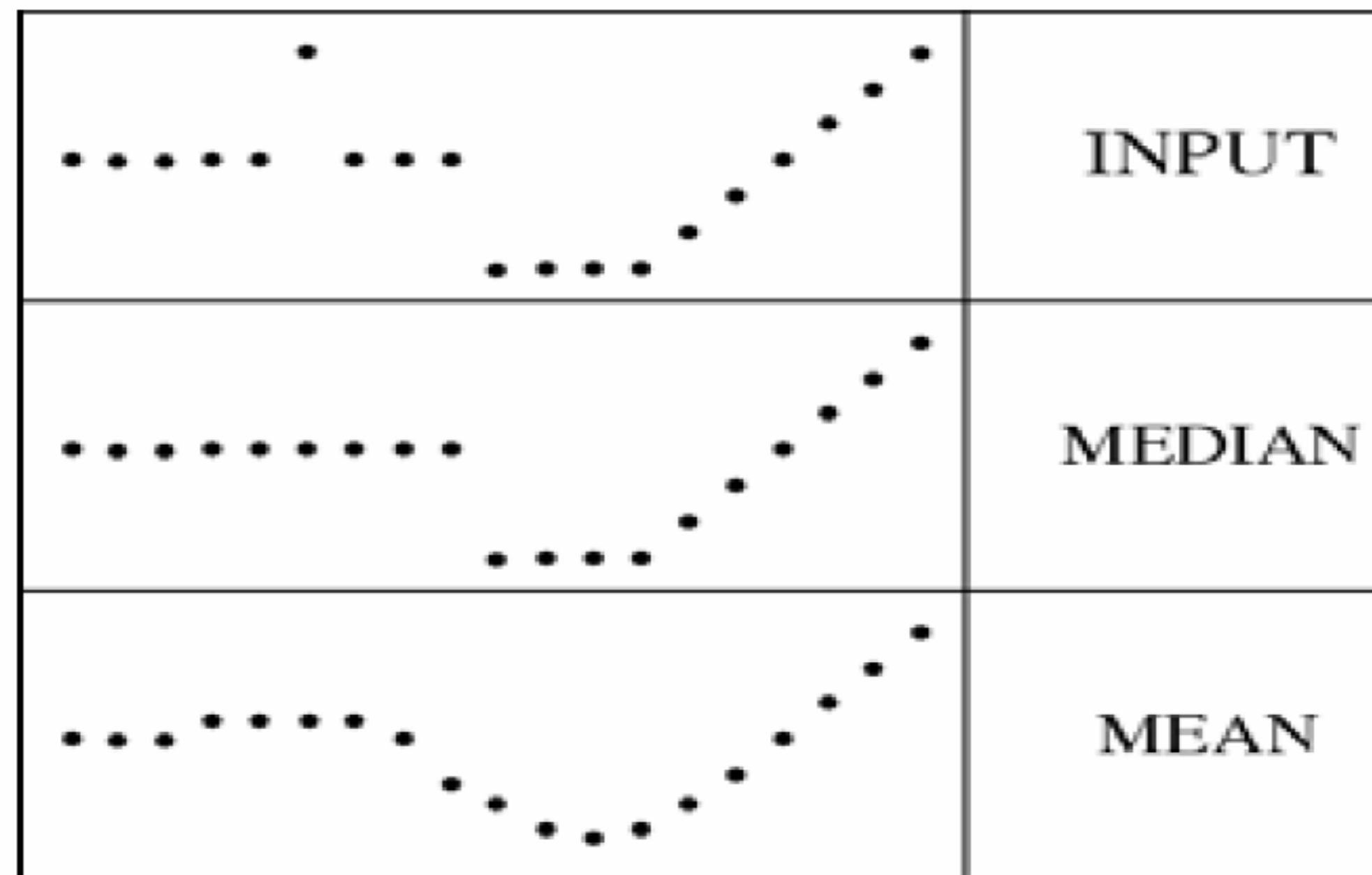


- le filtre médian est-il linéaire?

Filtre Médian

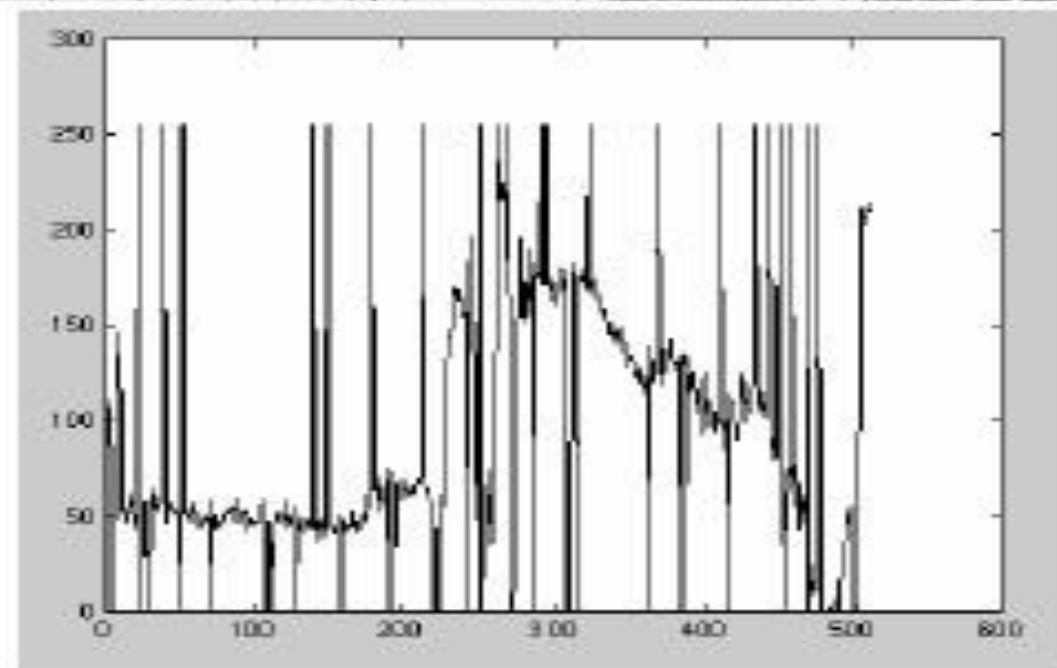
- Quel est l'avantage du filtrage médian sur le filtrage Gaussien ?
 - Robustesse au bruit d'impulsion (outliers)

filters have width 5 :

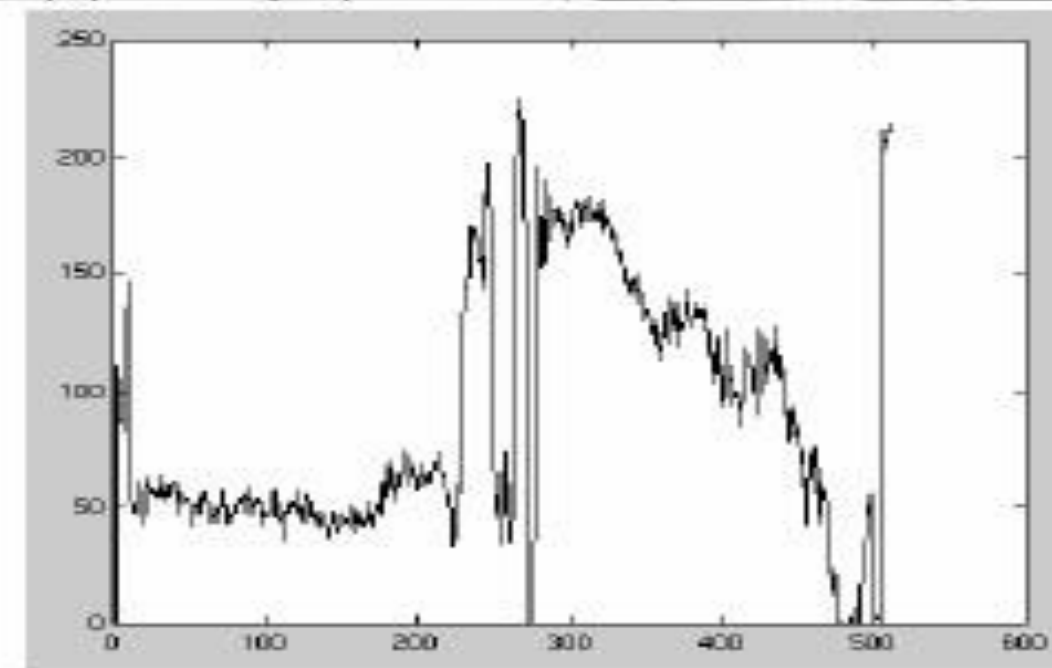


Filtre Médian

Bruit Salt-and-pepper noise



Filtrage Médian



MATLAB: `medfilt2(image, [h w])`

Filtrage gaussien vs. filtrage médian

3x3

5x5

7x7

Gaussien



Médian



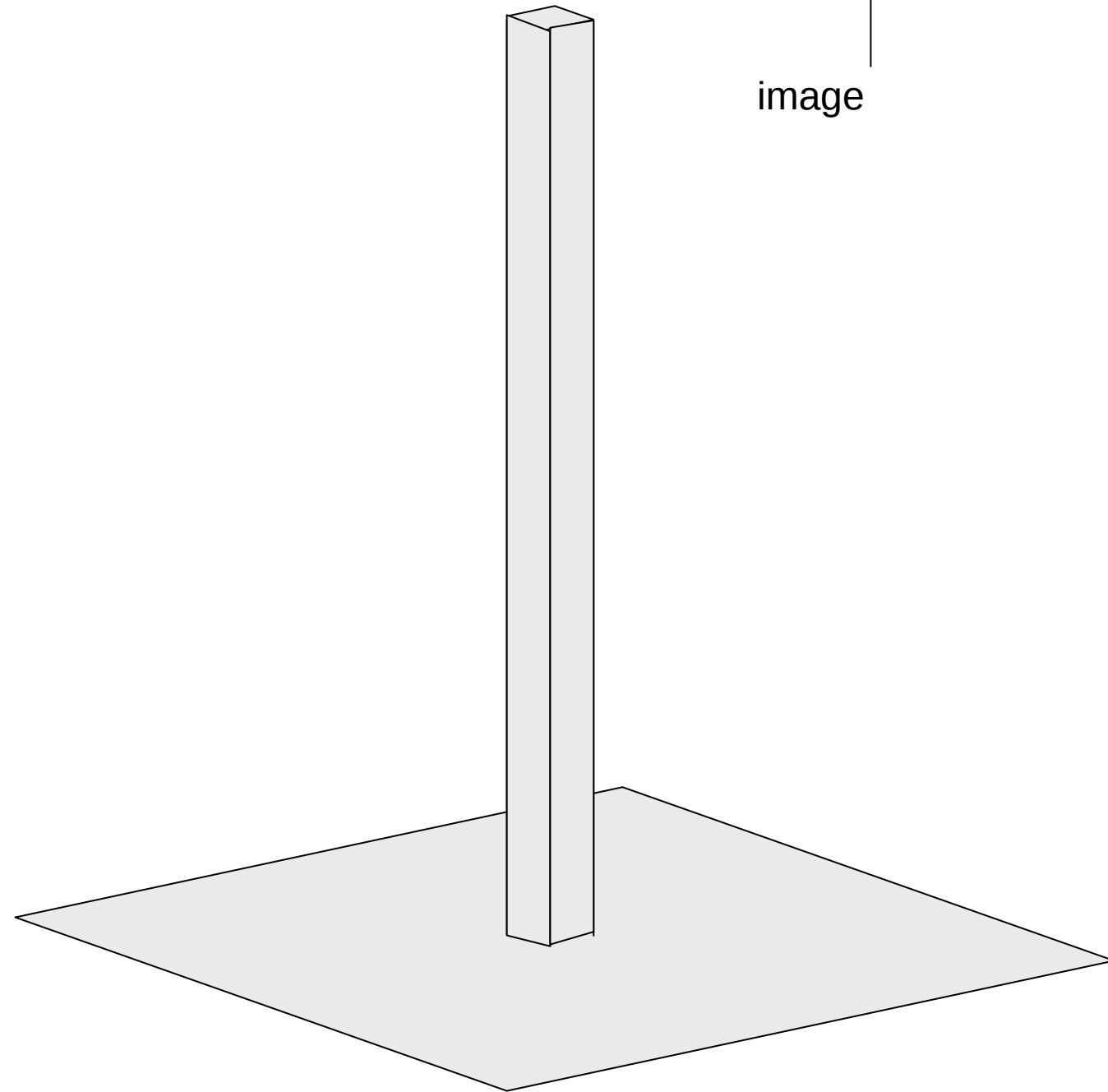
Filtre dé-accentuateur

$$f + \alpha(f - f * g) = (1 + \alpha)f - \alpha f * g = f * ((1 + \alpha)e - g)$$

image

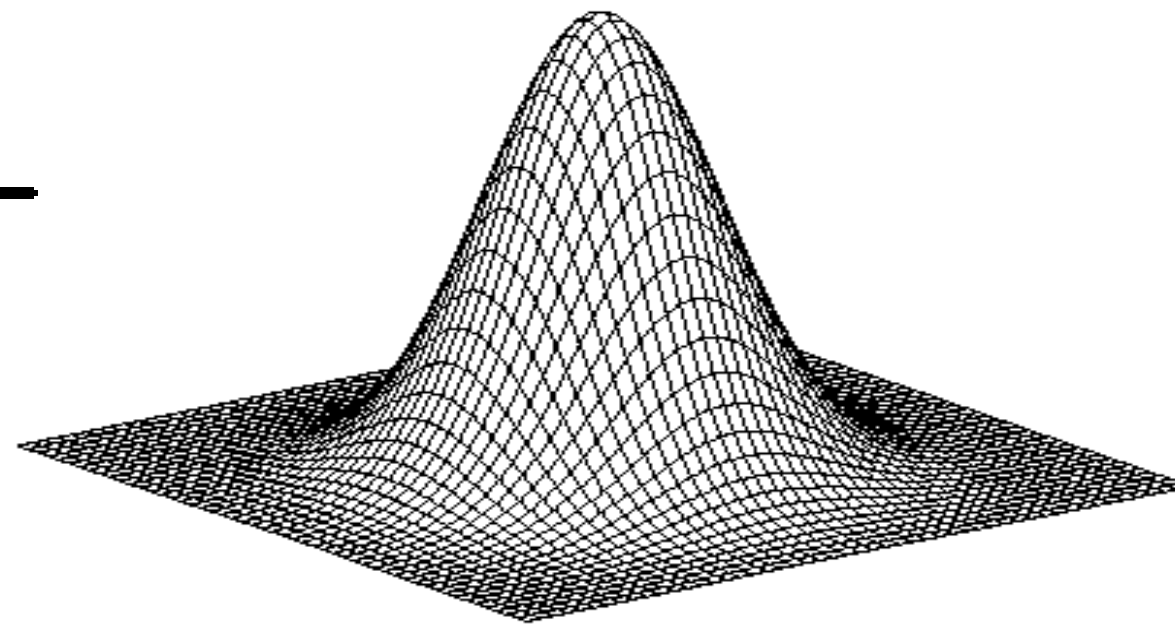
Image floue

Impulse unité
(identité)



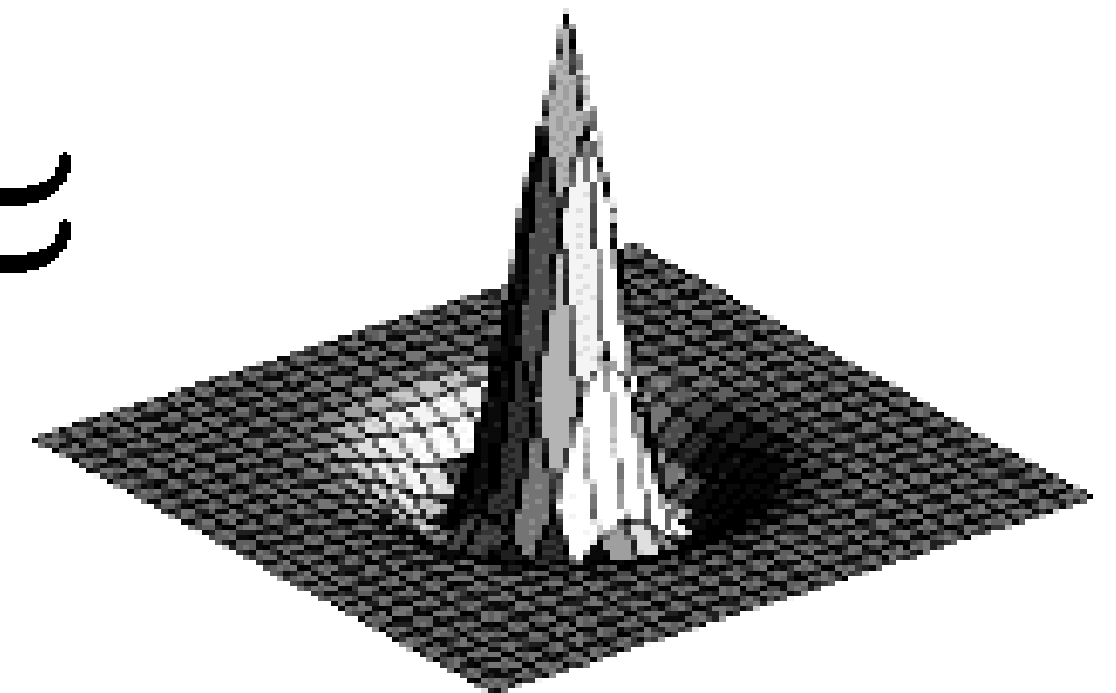
impulsion unité

—



Gaussien

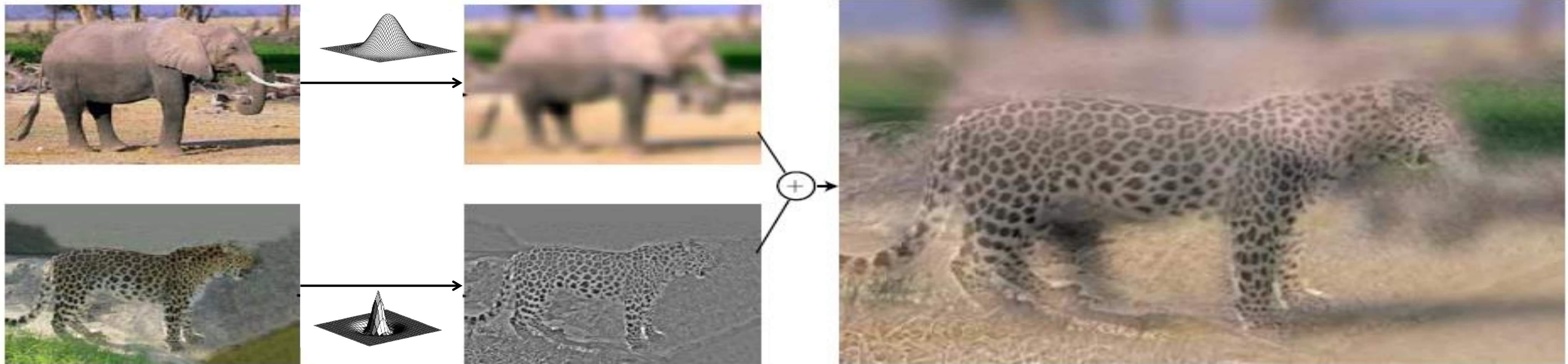
$\approx$



Laplacien d'un Gaussien

Application: Images Hybrides

Filtre Gaussien



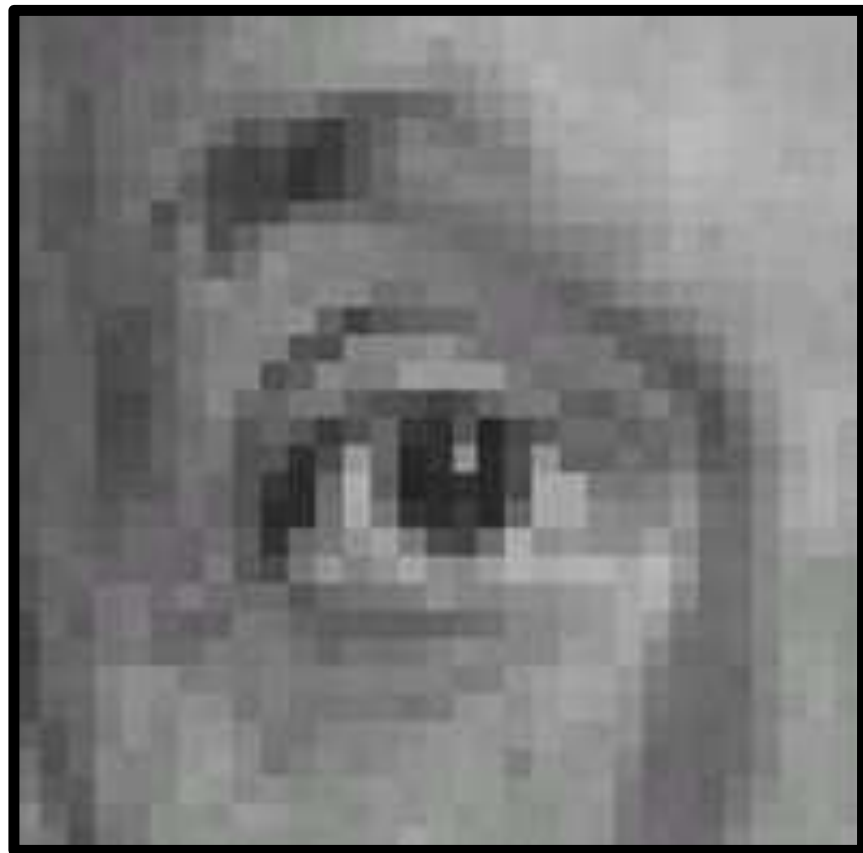
Filtre Laplacien

Voir : A. Oliva, A. Torralba, P.G. Schyns,
["Hybrid Images,"](#) SIGGRAPH 2006



Exercices ...

Exemple 1: pour s'échauffer



Original

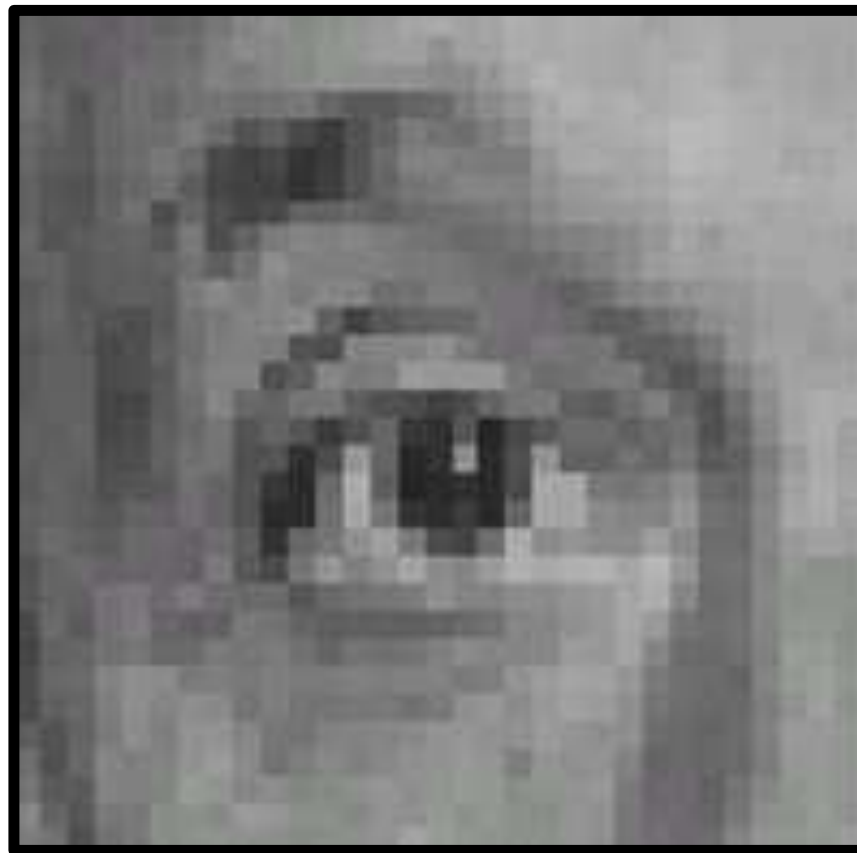
0	0	0
0	1	0
0	0	0

Filtre1



Résultat

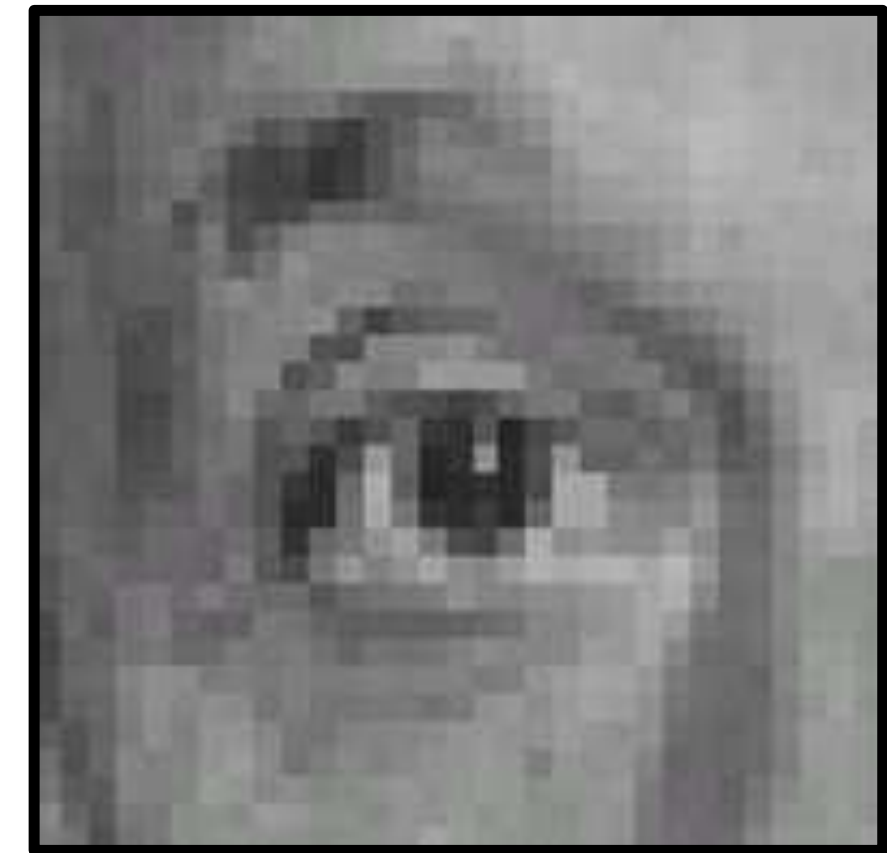
Exemple 1: résultat



Original

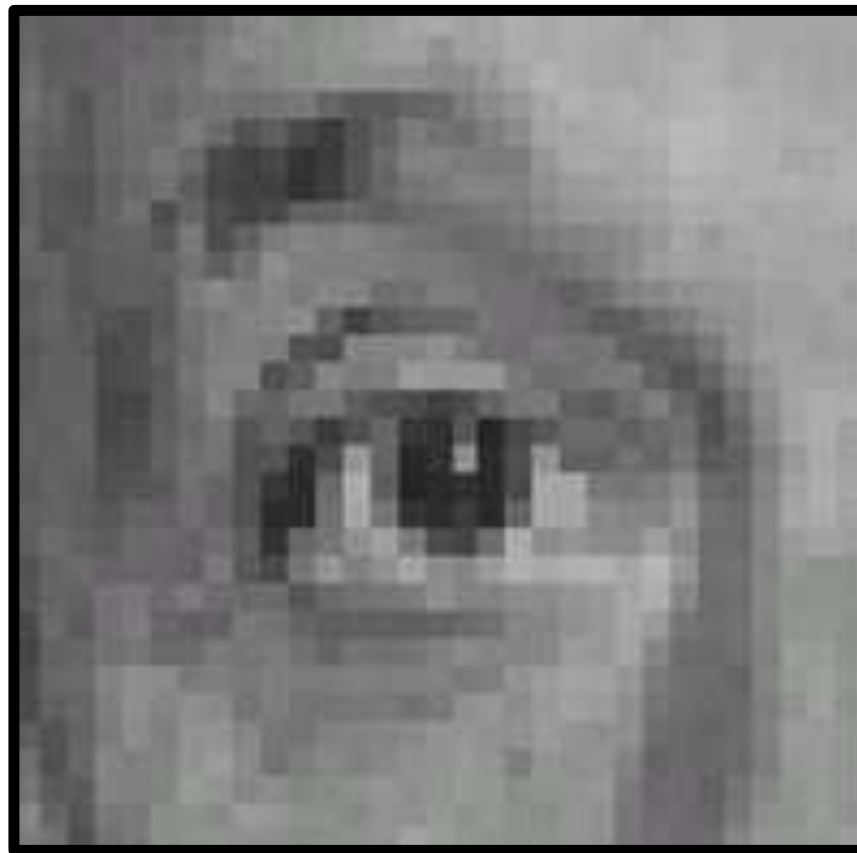
0	0	0
0	1	0
0	0	0

Filtre



Résultat
(pas de changement)

Exemple 2:



Original

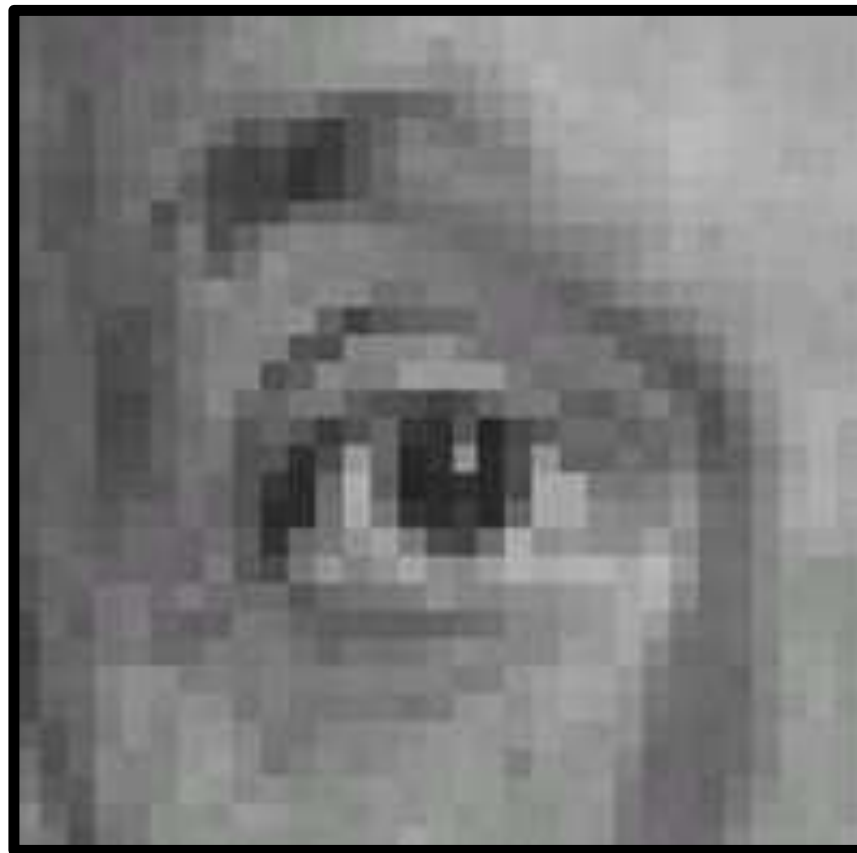
0	0	0
0	0	1
0	0	0

Filtre



Résultat

Exemple 2:



Original

0	0	0
0	0	1
0	0	0

Filtre



Résultat
(décalage à gauche
d'1 pixel)

Exemple 3:



Original

$\frac{1}{9}$

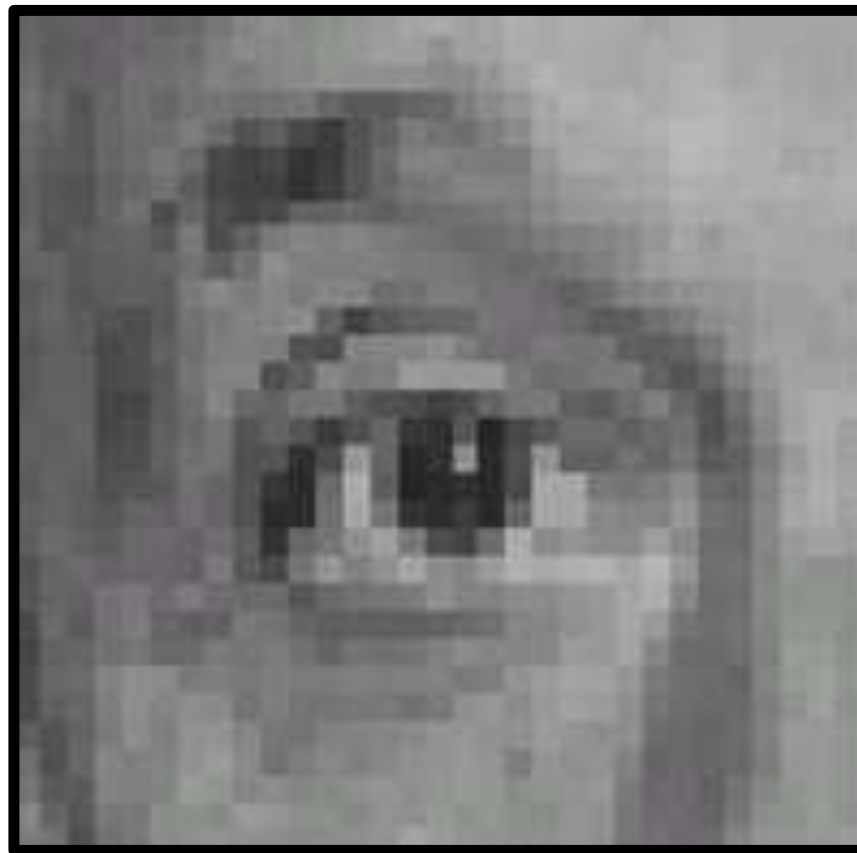
1	1	1
1	1	1
1	1	1

Filtre
(total=1)



Résultat

Exemple 3:



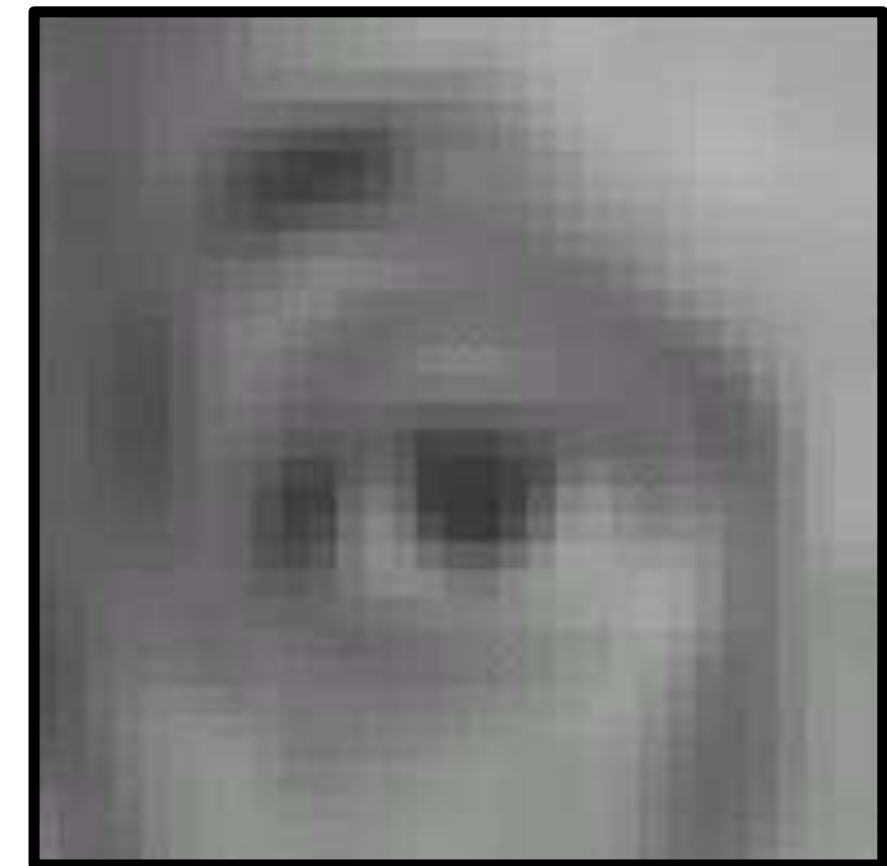
Original

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Filtre

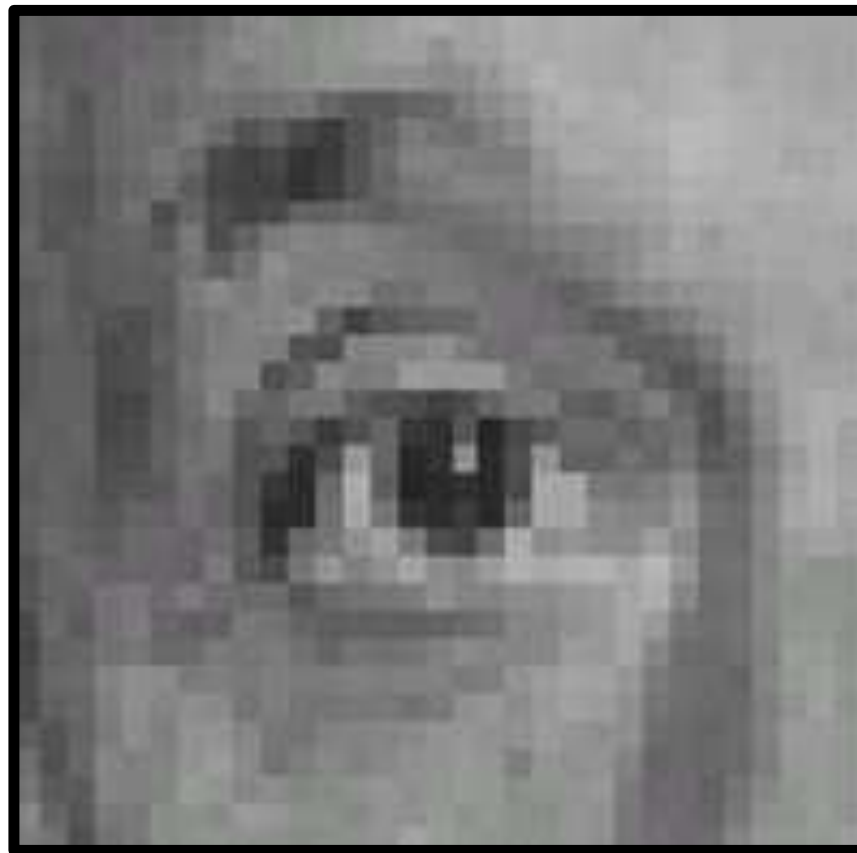
(somme des
poids = 1)



Résultat

(flou dû au filtre box)

Exemple 4:



Original

0	0	0
0	2	0
0	0	0

— $\frac{1}{9}$

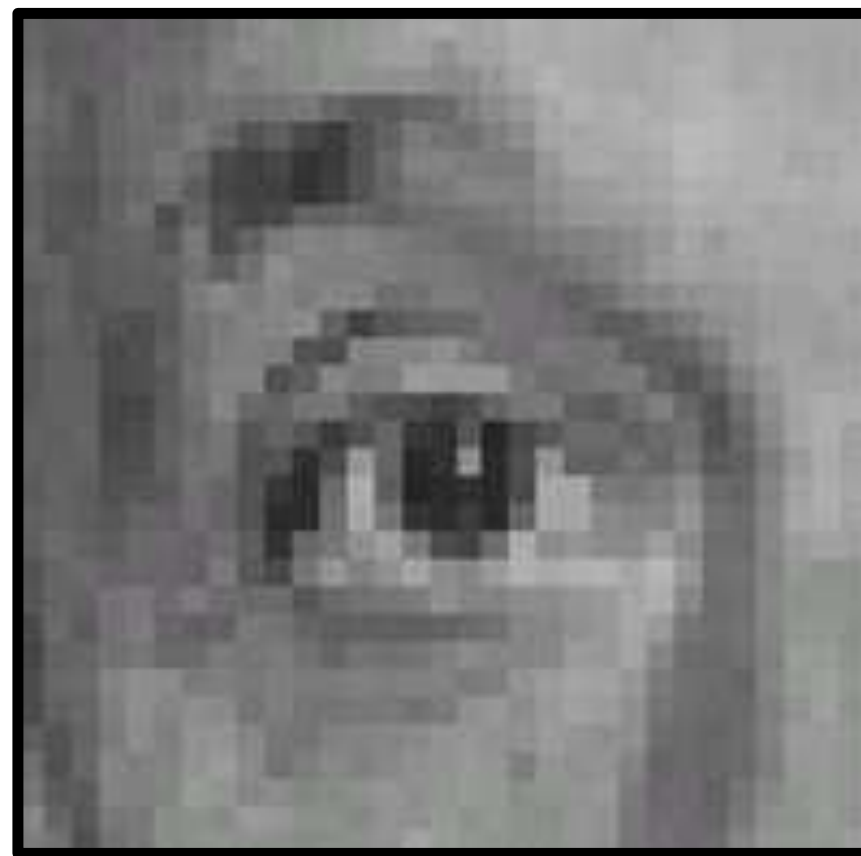
1	1	1
1	1	1
1	1	1

Filtre
(total=1)

?

Résultat

Exemple 4:



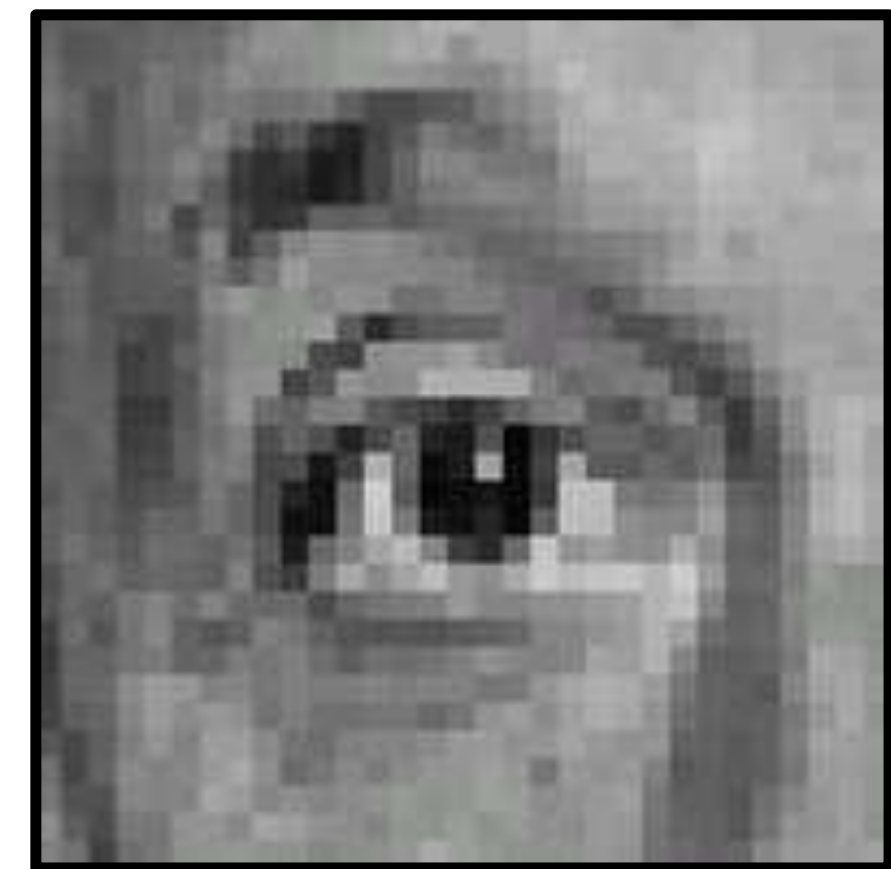
Original

0	0	0
0	2	0
0	0	0

 $- \frac{1}{9}$

1	1	1
1	1	1
1	1	1

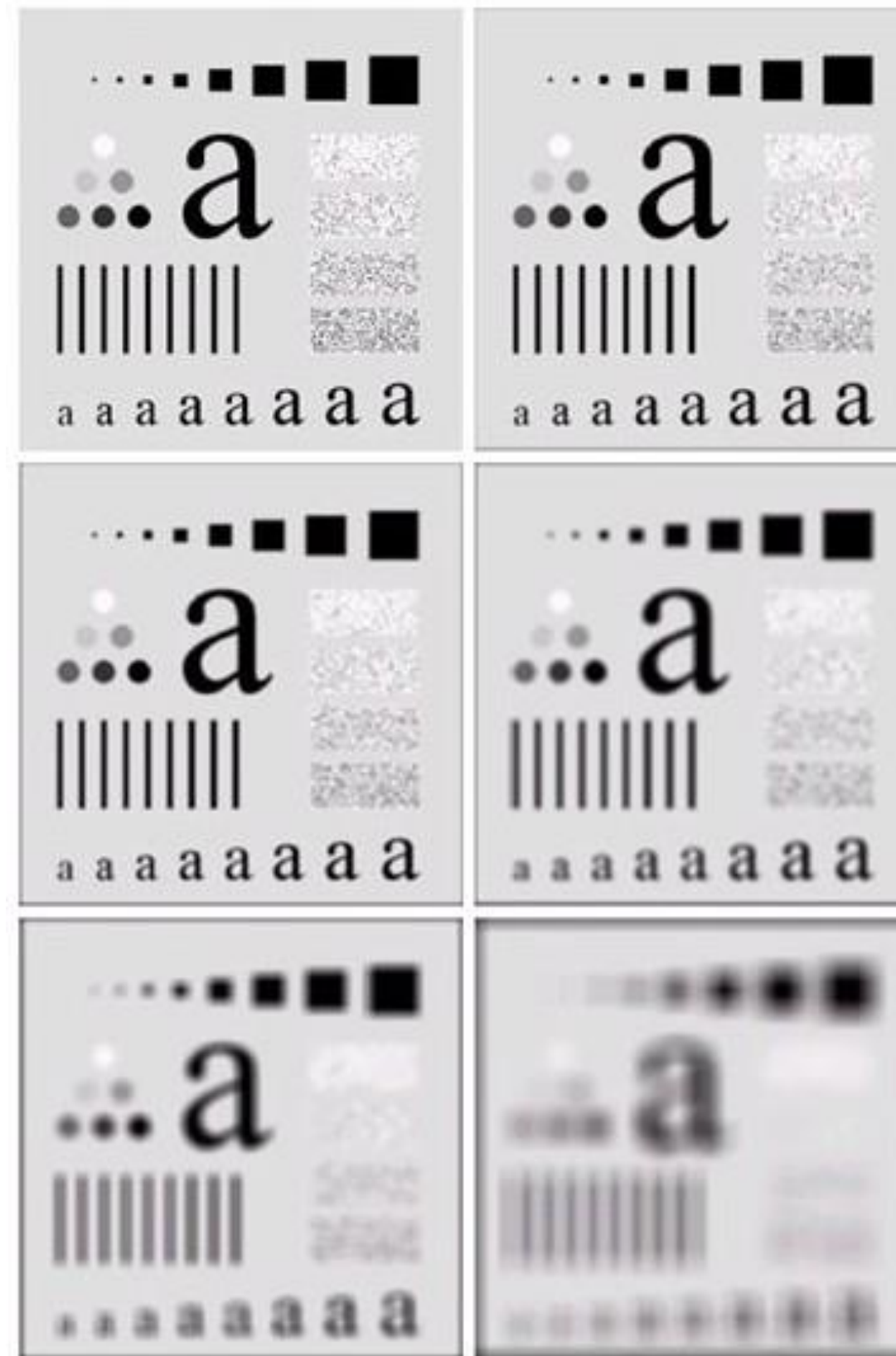
Filtre
(total=1)



Résultat
(accentuer)

Exemple 5: Lissage avec un Filtre Box

Original



3x3

9x9

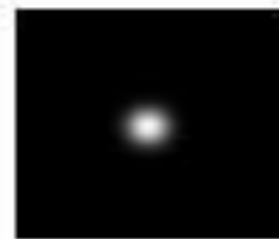
15x15

35x35

Exemple 5: Filtres Gaussiens



$\sigma = 1$ pixel



$\sigma = 5$ pixels



$\sigma = 10$ pixels



$\sigma = 30$ pixels