Traitement d’images

Pré-traitement

ÉCOLE D’INGÉNIEURS DU LITTORAL

Université Littoral Côte d’Opale

Plan

- Restauration d’images
- Amélioration d’images
- Compression d’images

Restauration d'images

- Définition
  - La restauration d'images a pour objet la réduction, voire l'élimination des distorsions introduites (bruits) par le système ayant servi à acquérir l'image.
  - Son but est d'obtenir une image qui soit la plus proche possible de l'image idéale qui aurait été obtenue si le système d'acquisition était parfait.
  - Différentes approches :
    - Le filtrage (temporel)
    - Le filtrage (fréquentiel)
    - Le filtrage non linéaire

Restauration d'images

- Bruit
  - Bruit lié au contexte de l'acquisition
    - Bougé, dérive lumineuse, flou, poussière, …
  - Bruit lié au capteur
    - Distorsion de la gamme des niveaux de gris, distorsion géométrique, mauvaise mise au point, …
  - Bruit lié à la numérisation
    - Codage, quantification, échantillonnage (moiré, effet poivre et sel), …

Restauration d'images

- Exemples

Image source

Flou de mise au point

Restauration d'images

- Exemples

Image source

Flou de bougé

Restauration d'images

- Exemples

Bruit uniforme (gaussien)

Bruit aléatoire (impulsionnelle)

- Filtrage
  - Convolution discrète à 2 dimensions
    - Cas continu

$$g(x,y) = f(x,y) * h(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-u, y-v)\cdot h(u,v)\,du\,dv$$

    - Cas discret

$$g(p,q) = f(p,q) * h(p,q) = \sum_i \sum_j f(p-i, q-j)\cdot h(i,j)$$

    - Remarque : en traitement d'images, on utilise plutôt l'opérateur de corrélation à condition d'avoir fait pivoter le noyau (ou masque) du filtre de 180°.

$$g(p,q) = f(p,q) \otimes h_\pi(p,q) = \sum_i \sum_j f(p+i, q+j)\cdot h_\pi(i,j)$$

Restauration d'images

- Rotation d'un filtre de taille 3×3

$h(-1,-1)$	$h(0,-1)$	$h(1,-1)$
$h(-1,0)$	$h(0,0)$	$h(1,0)$
$h(-1,1)$	$h(0,1)$	$h(1,1)$

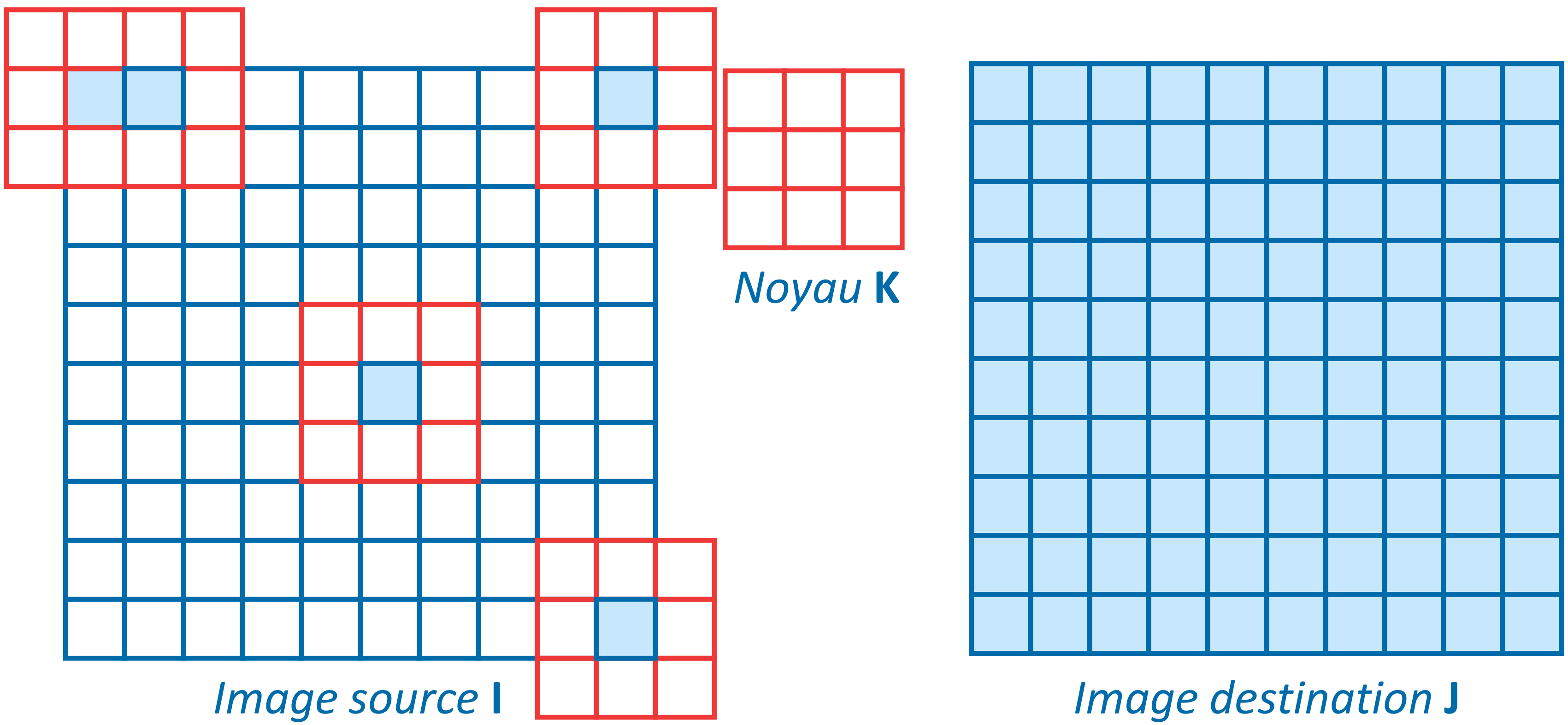
Filtre h de taille 3×3

$h_\pi(-1,-1) = h(1,1)$	$h_\pi(0,-1) = h(0,1)$	$h_\pi(1,-1) = h(-1,1)$
$h_\pi(-1,0) = h(1,0)$	$h_\pi(0,0) = h(0,0)$	$h_\pi(1,0) = h(-1,0)$
$h_\pi(-1,1) = h(1,-1)$	$h_\pi(0,1) = h(0,-1)$	$h_\pi(1,1) = h(-1,-1)$

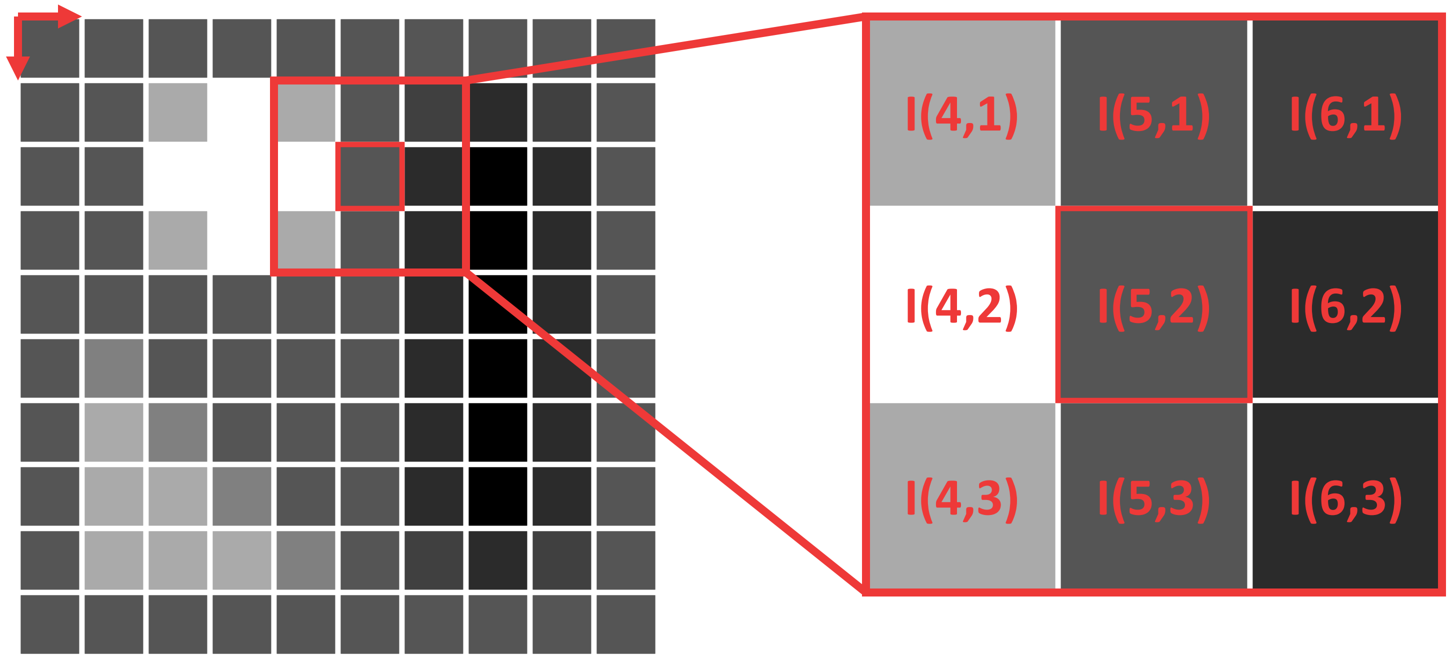
Filtre h_π de taille 3×3

Restauration d’images

- Parcours d’un noyau de convolution **K** sur une image **I**

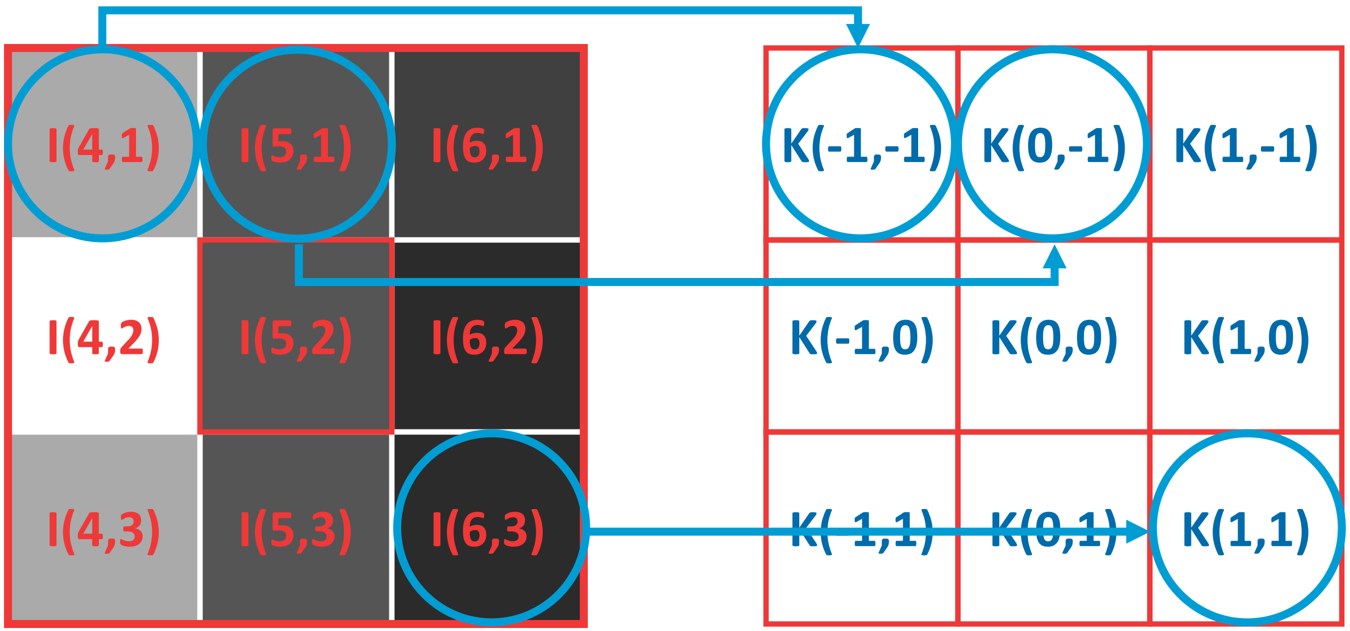


- Exemple



Restauration d’images

- Exemple



$$
\begin{aligned}
\mathbf{J}(5,2) \quad &= \mathbf{I}(4,1)\mathbf{K}(-1,-1) + \mathbf{I}(5,1)\mathbf{K}(0,-1) + \mathbf{I}(6,1)\mathbf{K}(1,-1) \\
&+ \mathbf{I}(4,2)\mathbf{K}(-1,0) + \mathbf{I}(5,2)\mathbf{K}(0,0) + \mathbf{I}(6,2)\mathbf{K}(1,0) \\
&+ \mathbf{I}(4,3)\mathbf{K}(-1,1) + \mathbf{I}(5,3)\mathbf{K}(0,1) + \mathbf{I}(6,3)\mathbf{K}(1,1)
\end{aligned}
$$

Restauration d'images

- Notion de filtre passe-bas
  - Il est utilisé pour atténuer les valeurs de pixels aberrantes. Il filtre les hautes fréquences spatiales (variations rapides des niveaux de gris) comme les contours et le bruit.
  - Il correspond à l'estimation d'une moyenne (pondérée ou non) dans le voisinage de chaque pixel. On parle alors de lissage ou de moyennage.
  - Il existe différents noyaux :
    - Uniforme
    - Gaussien
    - …

Restauration d’images

- Notion de filtre passe-haut
  - Ils sont utilisés pour mettre en évidence les variations dans l'image. Ils filtrent les basses-fréquences spatiales (variations lentes des niveaux de gris) comme les régions homogènes.
  - Ils correspondent à l’approximation de dérivées premières ou secondes.
  - Il existe différents types de noyau :
    - Les gradients horizontaux ou verticaux qui sont des estimations de la dérivée première (le gradient dans un direction donnée)
    - Les filtres de Sobel, Prewitt ou Roberts qui correspondent à des lissages des approximations de dérivées premières
    - Les filtres Laplacien qui sont des estimations de dérivées secondes
  - Les filtres doivent être combinés dans différentes directions.

Restauration d'images

- Quelques filtres usuels
  - Filtre moyenneur (lissage) : c'est un filtre passe-bas défini par *h* :

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

1/10	1/10	1/10
1/10	2/10	1/10
1/10	1/10	1/10

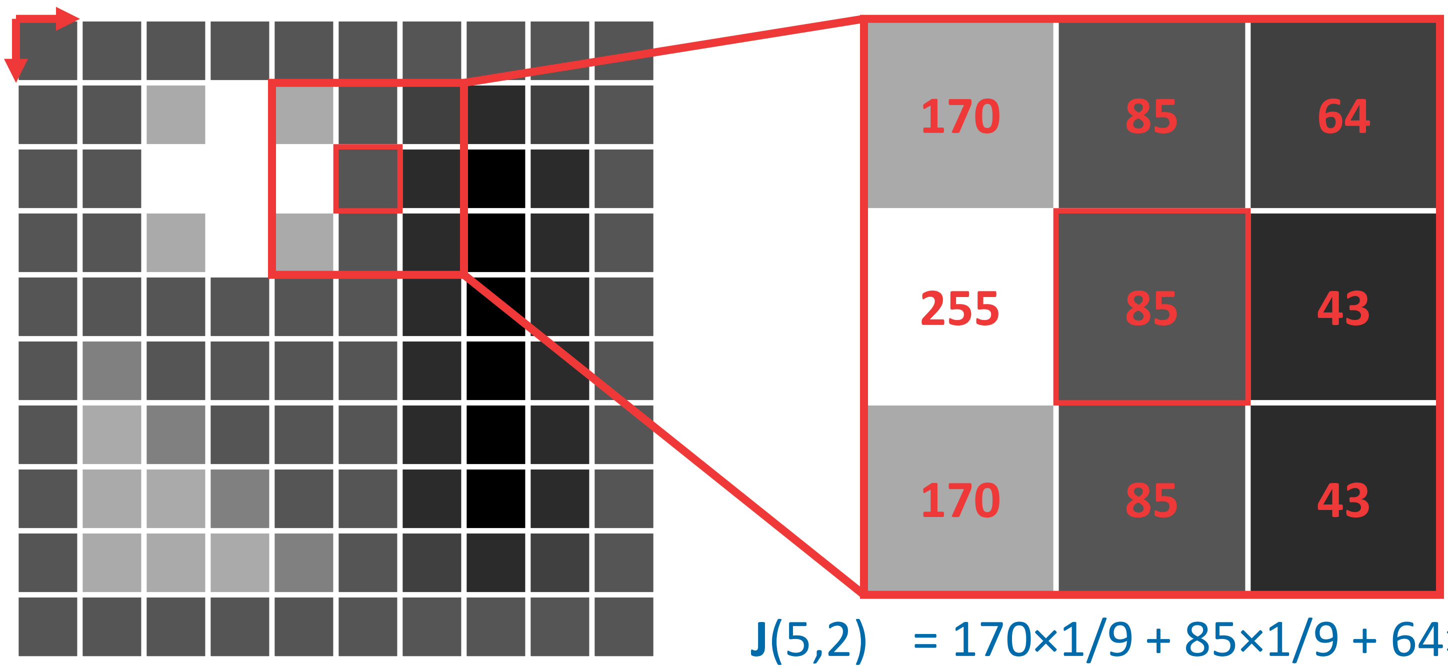
  - Filtre Gaussien : c'est un filtre passe-bas défini par *h* :

$$h(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

0,011	0,084	0,011
0,084	0,619	0,084
0,011	0,084	0,011

Restauration d’images

- Exemple avec un filtre moyenneur



$$
\begin{aligned}
J(5,2) &= 170\times 1/9 + 85\times 1/9 + 64\times 1/9 \\
&+ 255\times 1/9 + 85\times 1/9 + 43\times 1/9 \\
&+ 170\times 1/9 + 85\times 1/9 + 43\times 1/9 \\
&= 1000/9 = 111{,}11
\end{aligned}
$$

Restauration d'images

- Filtre Laplacien : c'est un filtre passe-haut défini par h_π :

-1	-1	-1
-1	8	-1
-1	-1	-1

0	-1	0
-1	4	-1
0	-1	0

- Filtre rehausseur

-1	-1	-1
-1	9	-1
-1	-1	-1

Restauration d'images

- Filtre séparable : $h(x,y) = h_1(x)*h_2(y)$ (exemple du filtre de Prewitt)

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Restauration d'images

- Application

Filtre Gaussien

Filtre moyenneur

Restauration d'images

- Application

Filtre laplacien

Filtre rehausseur

Restauration d’images

- Application

Filtre de Prewitt vertical

Filtre de Sobel horizontal

Restauration d'images

- Effet de bord
  - Taille de l'image de taille $X \times Y$ après convolution par un filtre de taille $P \times Q$: $(X + P - 1) \times (Y + Q - 1)$.
  - En pratique la taille est de $X \times Y$.
  - Bord non traité (mis a zéro, recopie ou aucun traitement).
  - Zero-padding : les valeurs du signal en dehors de l'image sont égales à zéro.
  - Périodisation : le signal image est périodisé.
  - Symétrie : les valeurs du signal en dehors de l'image sont obtenues par symétrie (effet miroir).

- Transformée de Fourier discrète à 2 dimensions
  - Transformée de Fourier bidimensionnelle

$$X(f,g) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x(u,v)\cdot \mathrm{e}^{-j2\pi(fu+gv)}\,dudv$$

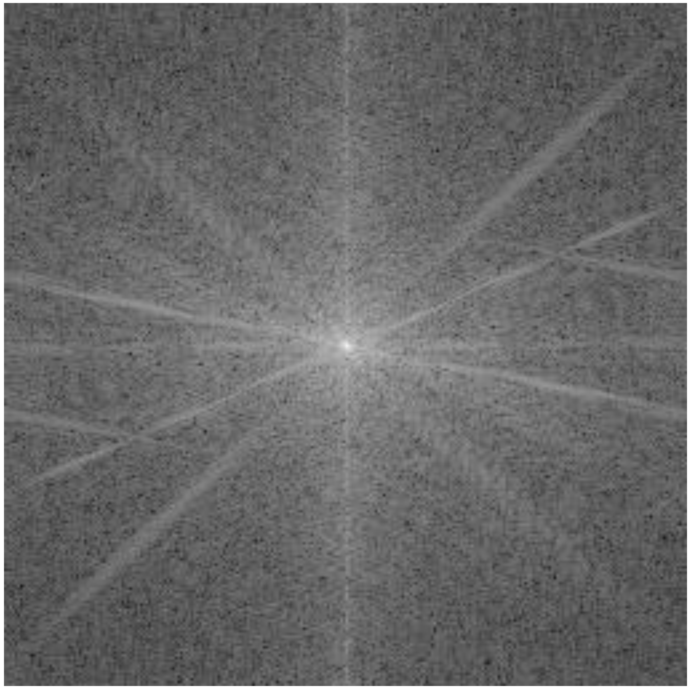
  - Transformée de Fourier discrète bidimensionnelle
    - La transformée de Fourier d'un signal numérique à support borné ($X\times Y$ échantillons) est un spectre échantillonné à support borné dont les $X\times Y$ échantillons sont définis par la transformée de Fourier discrète bidimensionnelle :

$$X_{m,n} = \sum_p \sum_q x_{p,q}\cdot \mathrm{e}^{-j2\pi\left(\frac{mp}{X}+\frac{nq}{Y}\right)}$$

Restauration d'images

- FFT (Fast Fourier Transform) d'une image

Image **I**



FFT de **I**

Restauration d'images

- Fréquence spatiale (image 256 × 256)

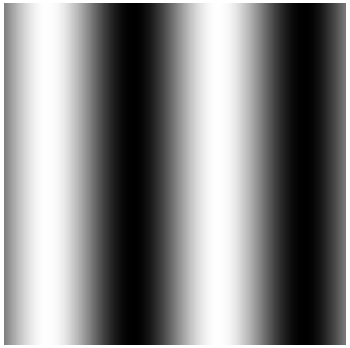


Image sinus (période 128)

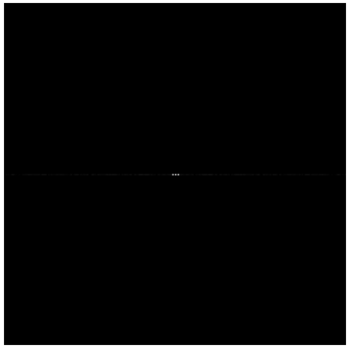
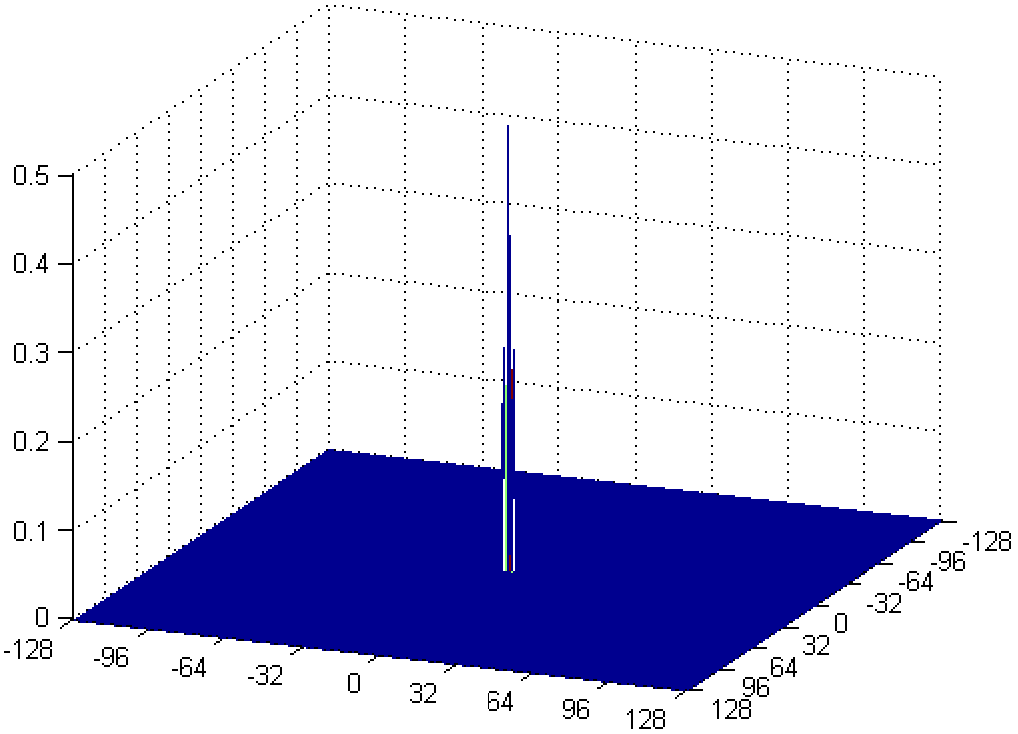


Image de la FFT (fréquence 2)



Module de la transformée de Fourier

Restauration d'images

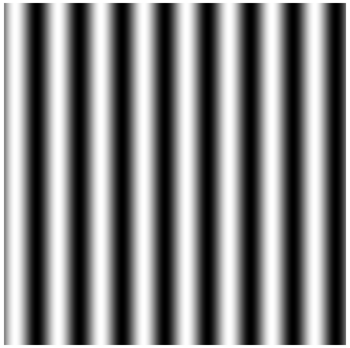


Image sinus (période 32)

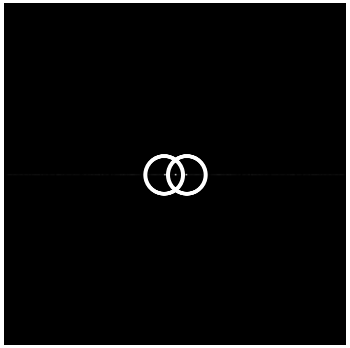
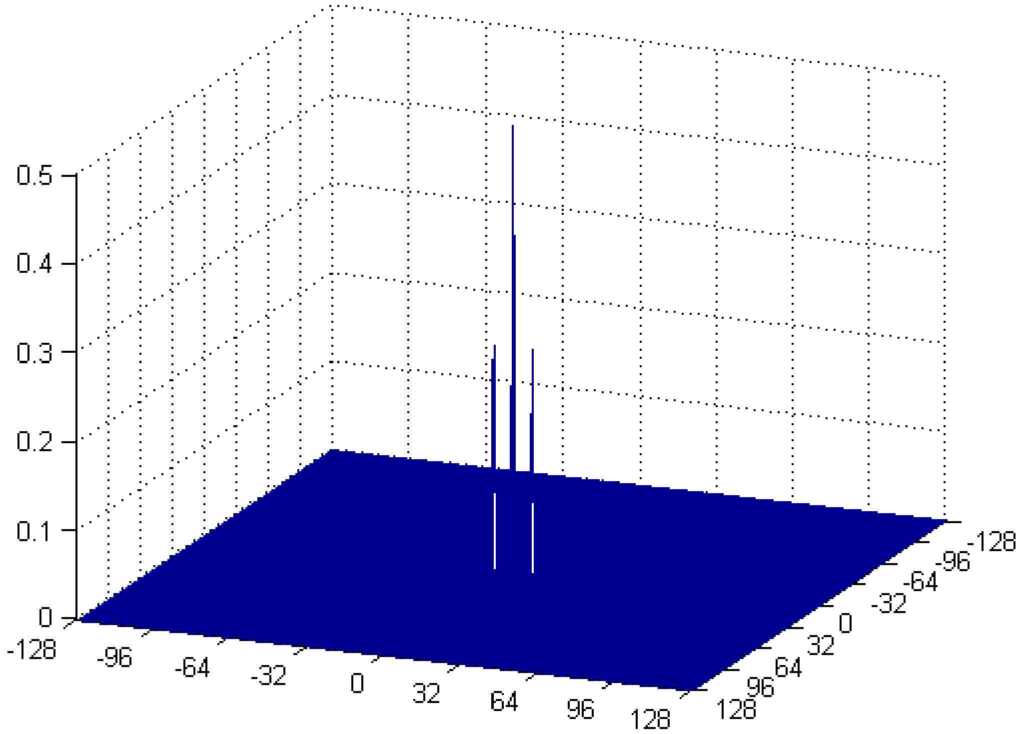


Image de la FFT (fréquence 8)



Module de la transformée de Fourier

Restauration d’images

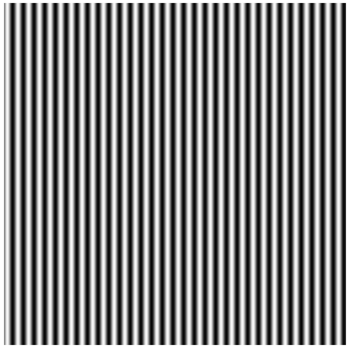


Image sinus (période 8)

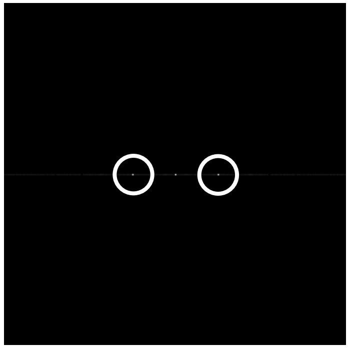
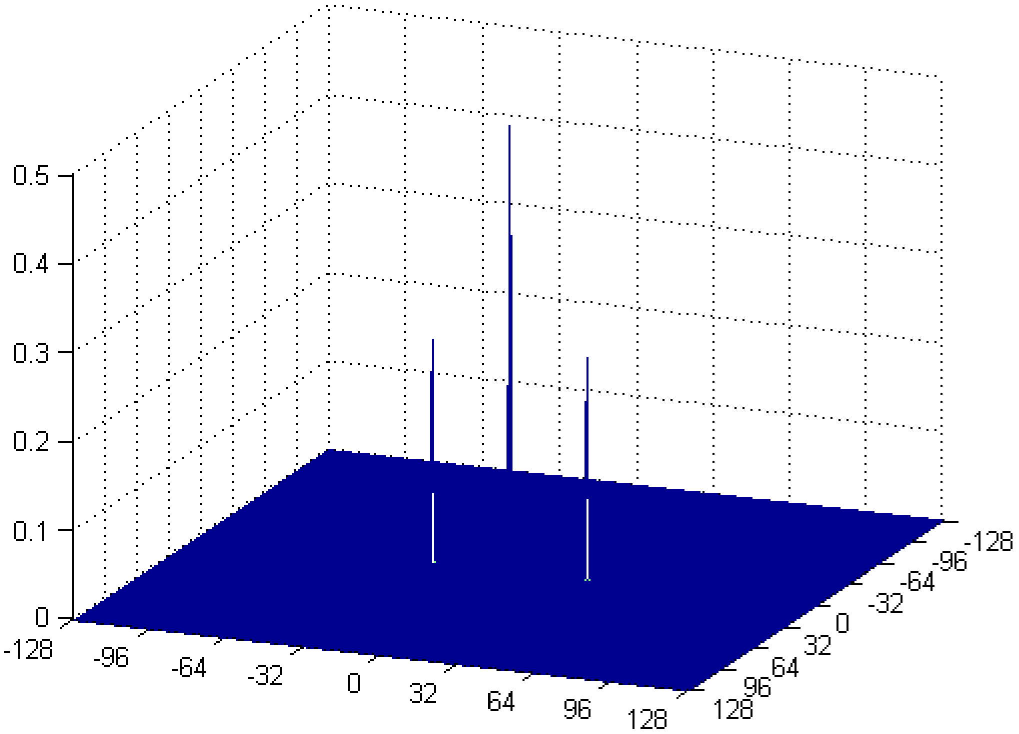
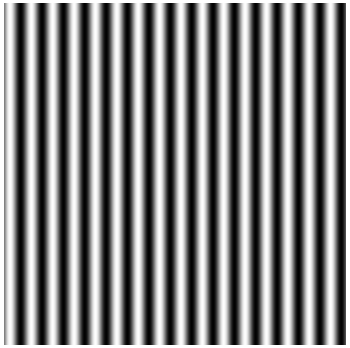


Image de la FFT (fréquence 32)



Module de la transformée de Fourier

- Orientation



Sinus horizontal (période 16)

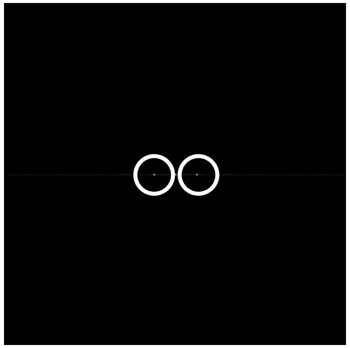
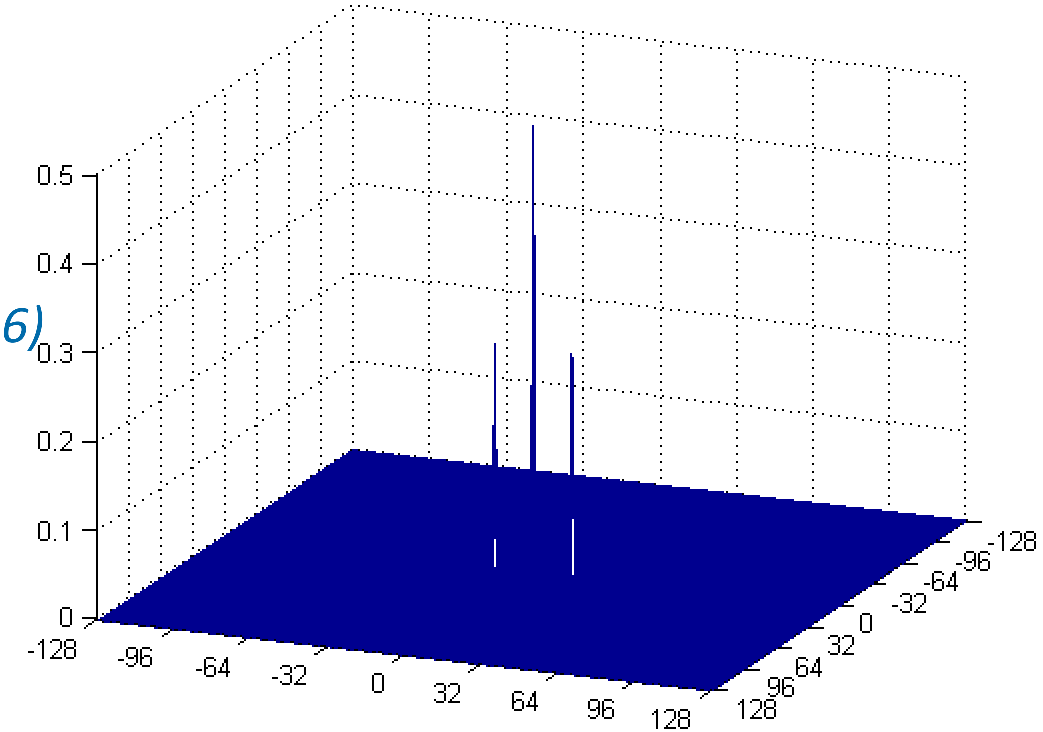
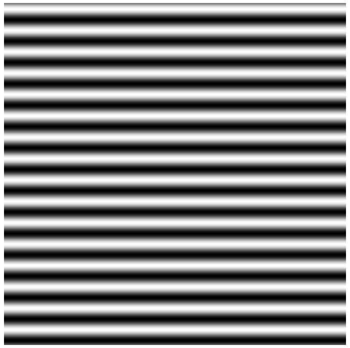


Image de la FFT (fréquence 16)



Module de la transformée de Fourier

Restauration d'images



Sinus vertical (période 16)

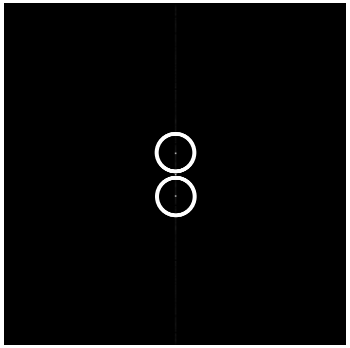
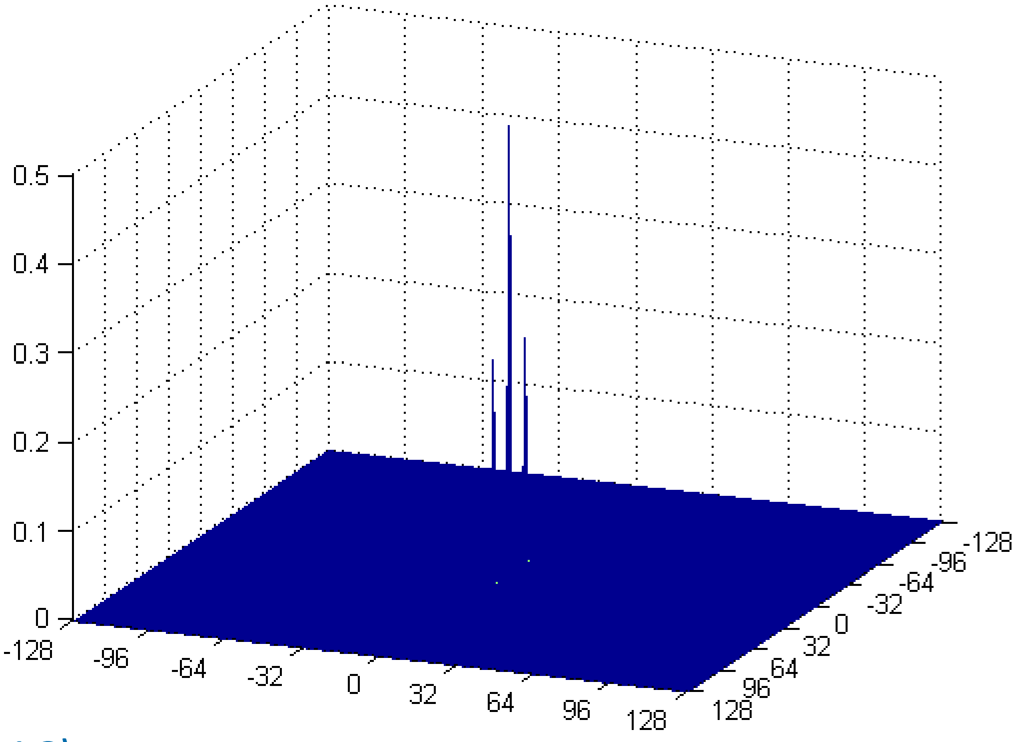


Image de la FFT (fréquence 16)



Module de la transformée de Fourier

Restauration d'images

Sinus diagonal (période 16)

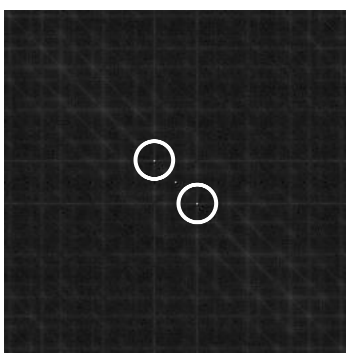
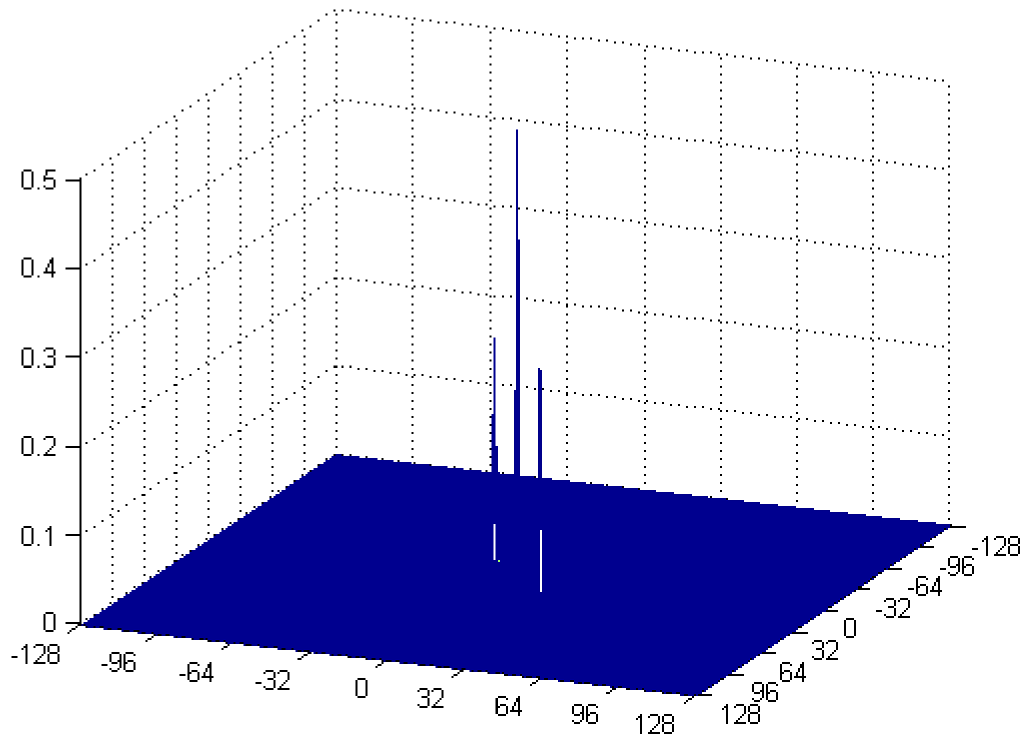


Image de la FFT (fréquence 16)



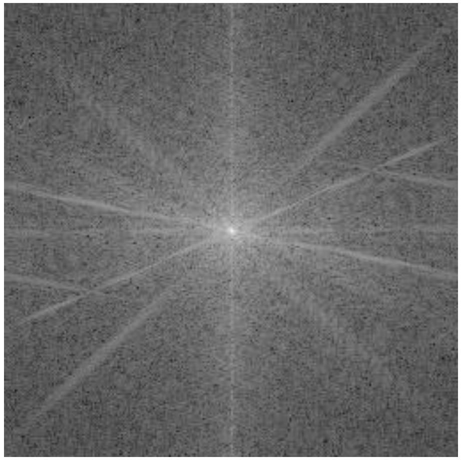
Module de la transformée de Fourier

- Application
  - Passe-bas : ils mettent en évidence les zones homogènes dans l'image où il y a peu de variation des niveaux, soit par élimination des hautes fréquences, soit par lissage (moyenneur, gaussien, …).
  - Passe-haut : ils mettent en évidence les zones hétérogènes où il y a des variations locales importantes des niveaux, soit par élimination des basses fréquences, soit par approximation des dérivées (Prewitt, Sobel, Roberts, Laplacien, …).
  - Passe-bande et coupe-bande : ils sélectionnent une fréquence particulière.
  - Filtre récursif (ou itératif) : application de convolution successives sur l'image.
  - Synthèse de filtres : le filtre est défini en fonction du modèle souhaité.

Restauration d’images

Image **I**

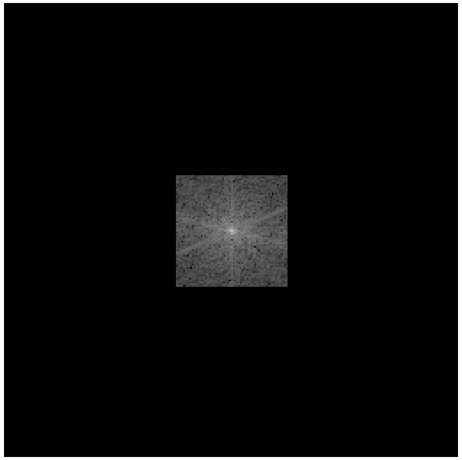
FFT^{-1} de **J**



Filtre passe-bas

FFT de **I**

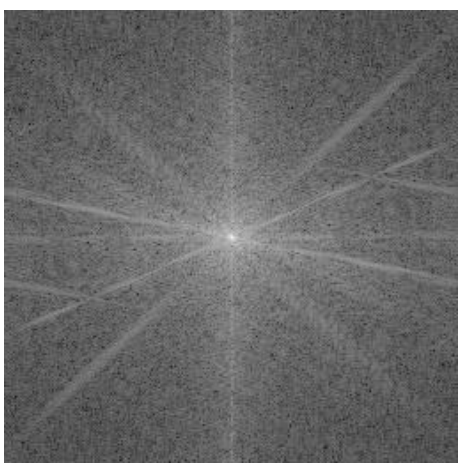
Image **J**



Restauration d’images

Image **I**

FFT de **I**



Filtre passe-haut

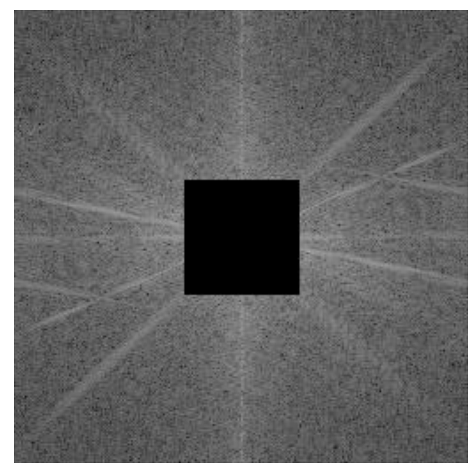
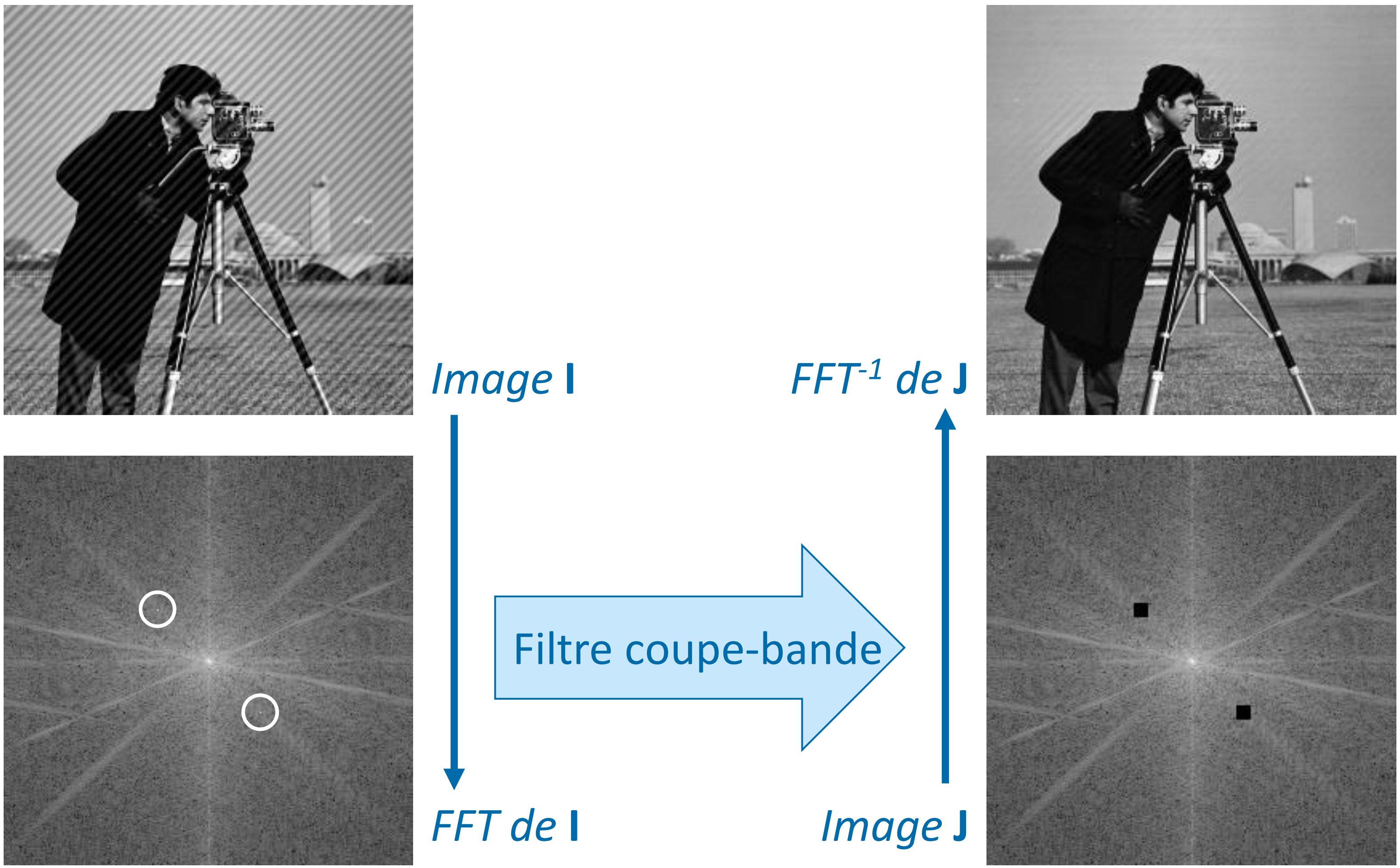


Image **J**

FFT^{-1} de **J**

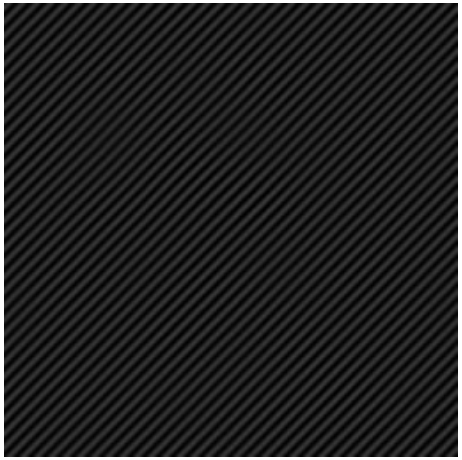
Restauration d'images

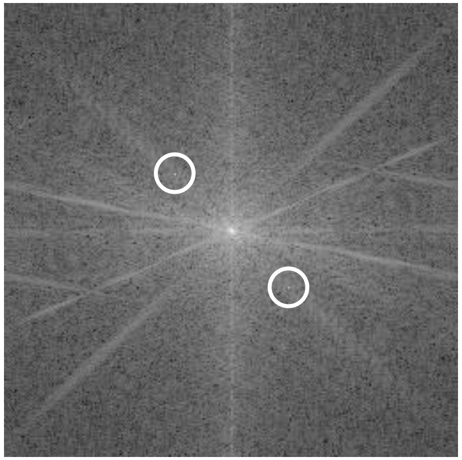


Restauration d'images

Image **I**

FFT^{-1} de **J**

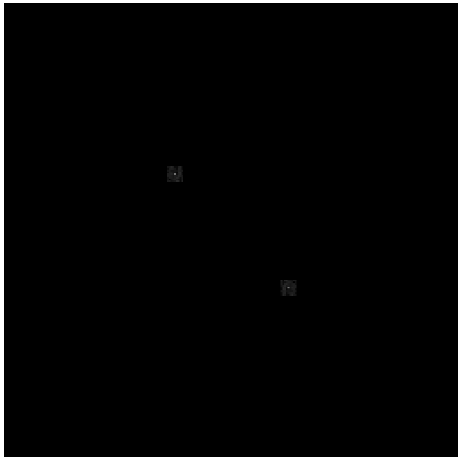




Filtre passe-bande

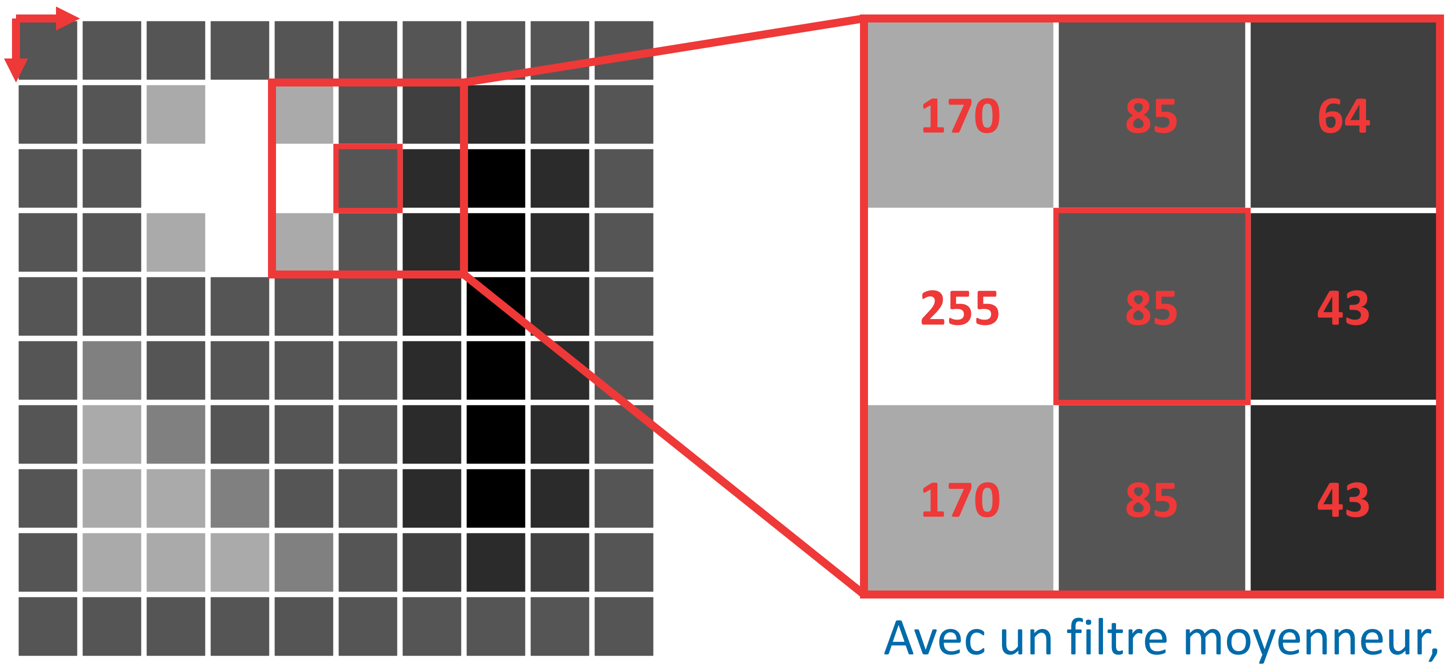
FFT de **I**

Image **J**



Restauration d’images

- Filtrage non linéaire
  - Filtre médian



Avec un filtre moyenneur,
on obtient **J**(5,2) = 111

Restauration d'images

170	85	64
255	85	43
170	85	43

43	43	64	85	85	85	170	170	255

Avec un filtre médian,
on obtient **J**(5,2) = 85

Restauration d'images

- Lissage robuste

170	85	64
255	85	43
170	85	43

255	170	170	85	85	85	64	43	43

255 | 170 | 170 | 85 | 85 → 170

170 | 170 | 85 | 85 | 85 → 85

170 | 85 | 85 | 85 | 64 → 106

85 | 85 | 85 | 64 | 43 → 42

85 | 85 | 64 | 43 | 43

Demi intervalle le plus compact

85 – 43 = 42

J(5,2) = 85

Restauration d’images

- Lissage min/max

170	85	64
255	85	43
170	85	43

Valeur minimale, m = 43

Valeur maximale, M = 255

Si **J**(5,2) < (M – m)/2 alors **J**(5,2) = m

Si **J**(5,2) > (M – m)/2 alors **J**(5,2) = M

Ici (M – m)/2 = 106, donc **J**(5,2) = 43

Restauration d'images

- Lissage SNN (Symetric Nearest Neighbor)

170	85	64
255	85	43
170	85	43

	85	64
	85	43
		43

Après avoir déterminé la valeur la plus proche de celle du pixel central pour chaque paire de points symétriques par rapport au pixel central, il faut calculer la moyenne sur les valeurs restantes.

Ici **J**(5,2) = 64.

Restauration d'images

ÉCOLE D'INGÉNIEURS DU LITTORAL

- Lissage adaptatif

170	85	64
255	85	43
170	85	43

Les coefficients du filtre de convolution s'adapte automatiquement à l'image

?	?	?
?	?	?
?	?	?

170-85 = 85	85-85 = 0	85-64 = 21
255-85 = 170	0	85-43 = 42
170-85 =85	85-85 =0	85-43 = 42

$$\mathbf{K}(x,y) = 1/\Sigma \times \begin{bmatrix} 1/85 & 0 & 1/21 \\ 1/170 & 0 & 1/42 \\ 1/85 & 0 & 1/42 \end{bmatrix}$$

$$\Sigma = 1/85 + 1/21 + 1/170 + 1/42 + 1/85 + 1/42$$

- Application

Bruit impulsionnelle

Bruit uniforme

Restauration d'images

- Filtre moyenneur

Restauration d’images

- Filtre médian

Restauration d'images

- Lissage robuste

Restauration d'images

- Lissage min/max

Restauration d'images

- Lissage SNN

Restauration d'images

- Lissage adaptatif

Restauration d'images

- Autres filtres
  - Autres filtres d'ordre
  - Filtres homomorphiques
  - Filtres morphologiques
  - Autres filtres adaptatifs
- Démo Matlab
  - Restauration d'images : nrfiltdemo
  - Filtrage : firdemo

Amélioration d'images

- Définition
  - L'amélioration a pour but de satisfaire l'oeil de l'observateur humain.
  - L'oeil humain est essentiellement sensible aux forts contrastes. C'est pourquoi les techniques d'amélioration tentent d'augmenter ceux-ci dans le but d'accroître la séparabilité des régions composant une scène.
  - Différentes approches :
    - La modification d'histogramme
    - Le filtrage (fréquentiel)
    - Autres techniques

Amélioration d'images

- Contraste
  - Variance des niveaux des pixels de l'image

$$\frac{1}{N}\sum_{n=1}^{N}(I(i,j)-Moy)^2$$

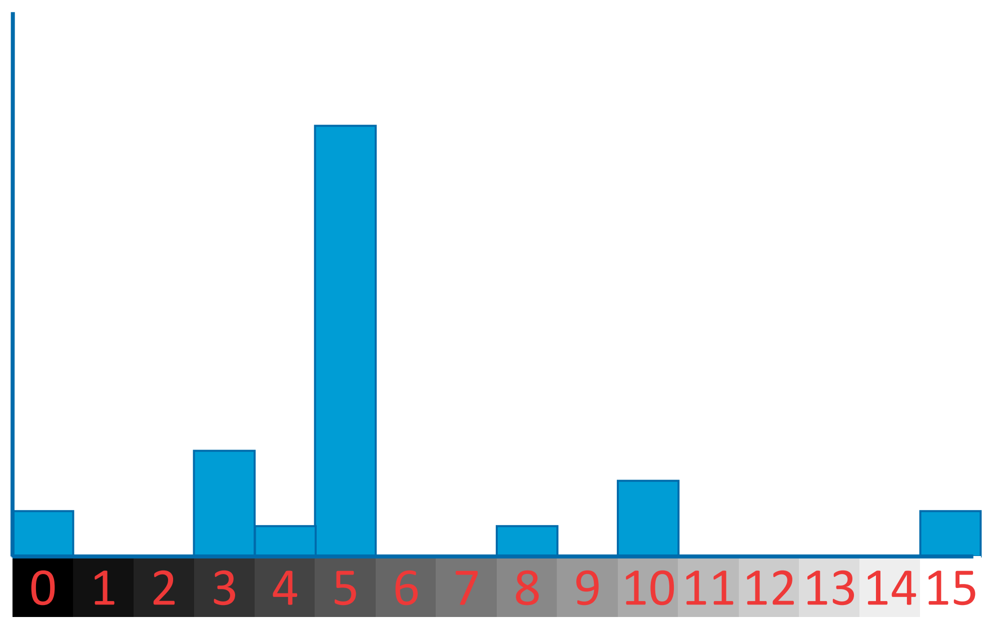
  - Variation entre les niveaux minimum et maximum

$$\frac{max[I(i,j)]-min[I(i,j)]}{max[I(i,j)]+min[I(i,j)]}$$

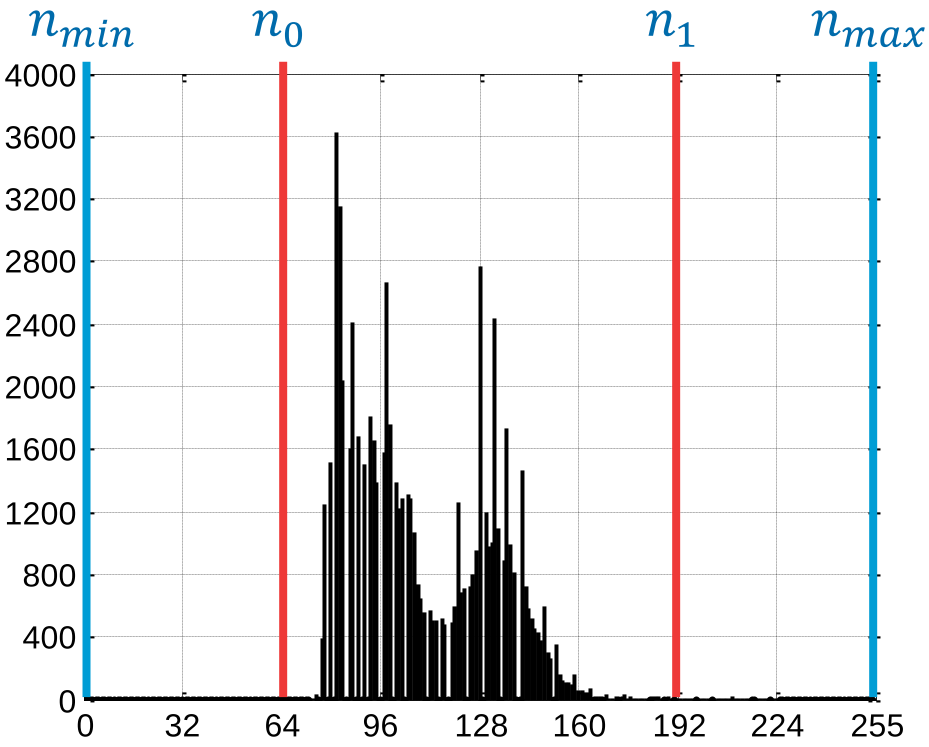
Amélioration d'images

- La modification d'histogramme
  - L'histogramme

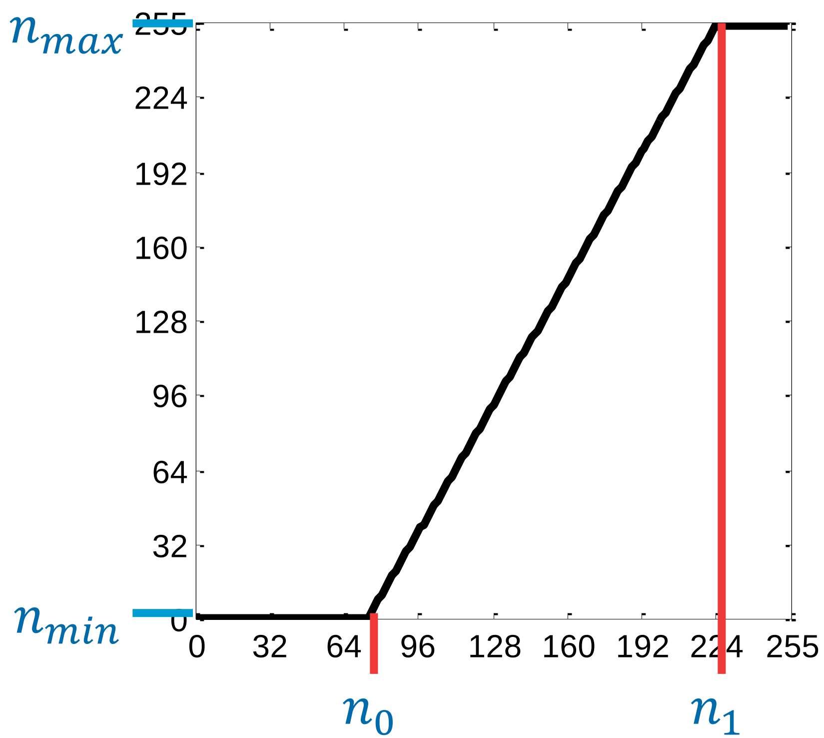
5	5	5	5	5	5	5	5	5	5
5	5	10	15	10	5	4	3	4	5
5	5	15	15	15	5	3	0	3	5
5	5	10	15	10	5	3	0	3	5
5	5	5	5	5	5	3	0	3	5
5	8	5	5	5	5	3	0	3	5
5	10	8	5	5	5	3	0	3	5
5	10	10	8	5	5	3	0	3	5
5	10	10	10	8	5	4	3	4	5
5	5	5	5	5	5	5	5	5	5



- Recadrage dynamique



Amélioration d'images



$$n' = a \times n + b$$

$$a = \frac{n_{max} - n_{min}}{n_1 - n_0}$$

$$b = \frac{n_{min} \times n_1 - n_{max} \times n_0}{n_1 - n_0}$$

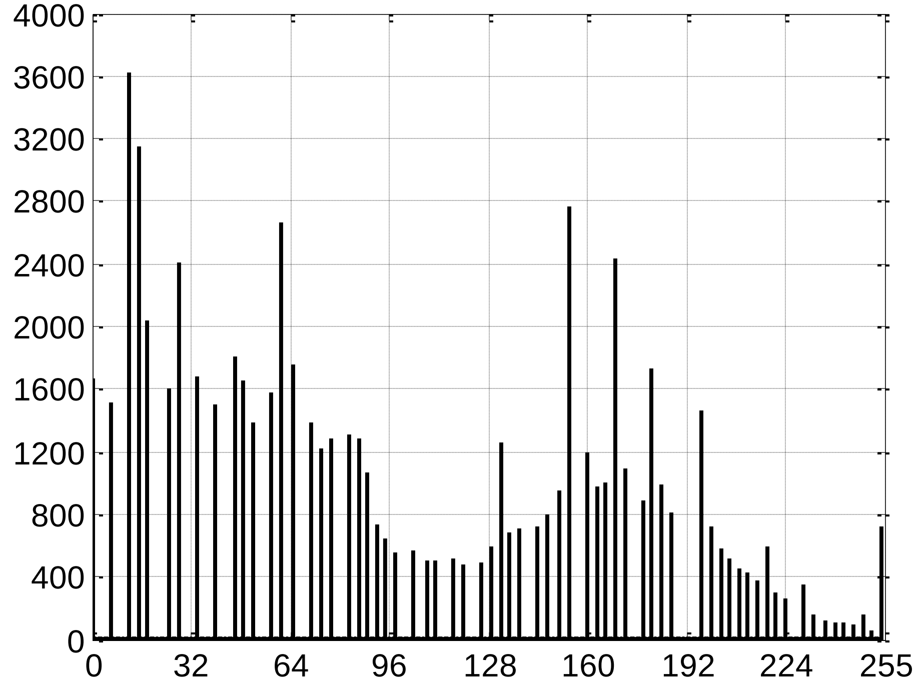
Soit $n_{max} = Q - 1$ et $Q = 256$

Pour $n_{max} = 255$ et $n_{min} = 0$:

$$\Rightarrow n' = 255 \times \frac{n - n_0}{n_1 - n_0}$$

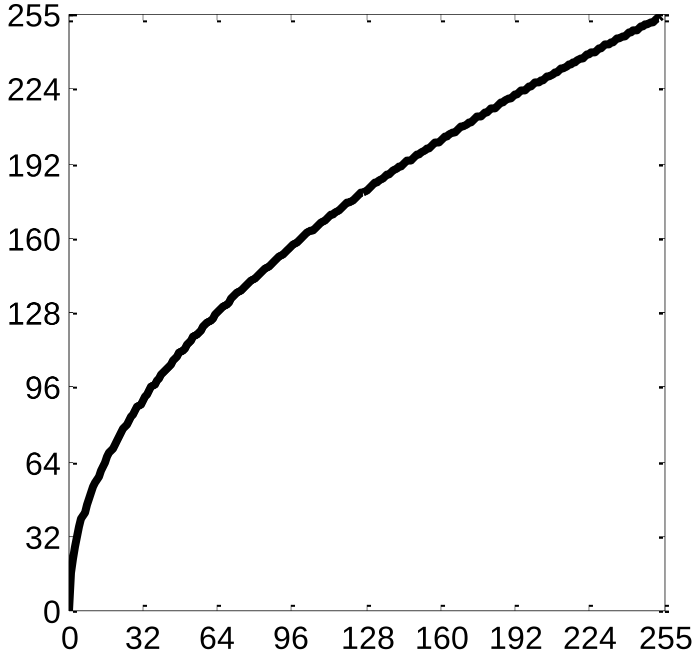
→ **Valeur entière de** n'

Amélioration d'images



Amélioration d’images

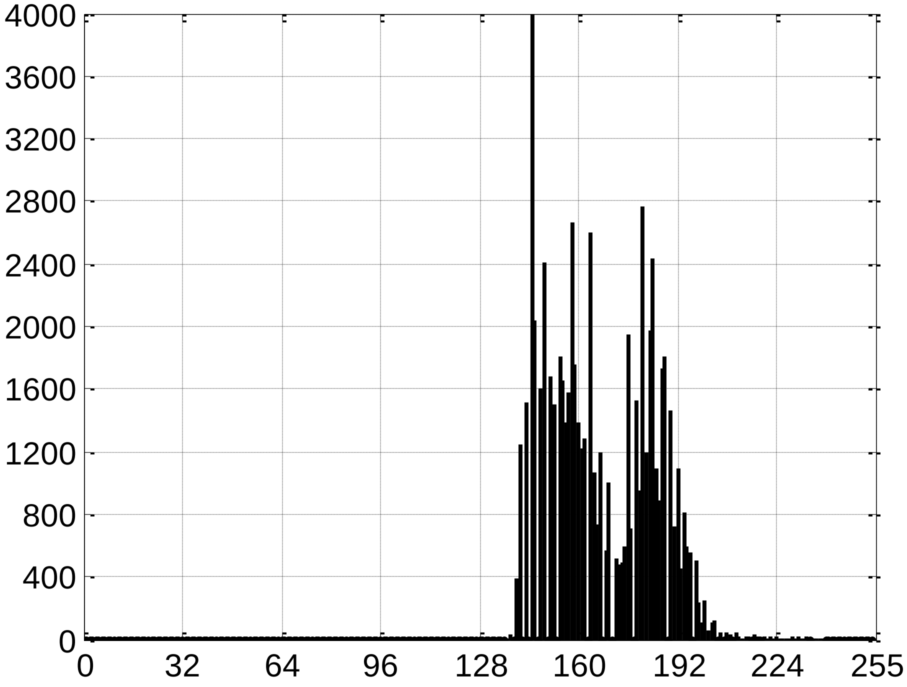
- Correction gamma



$$n' = n^{1/\gamma}$$

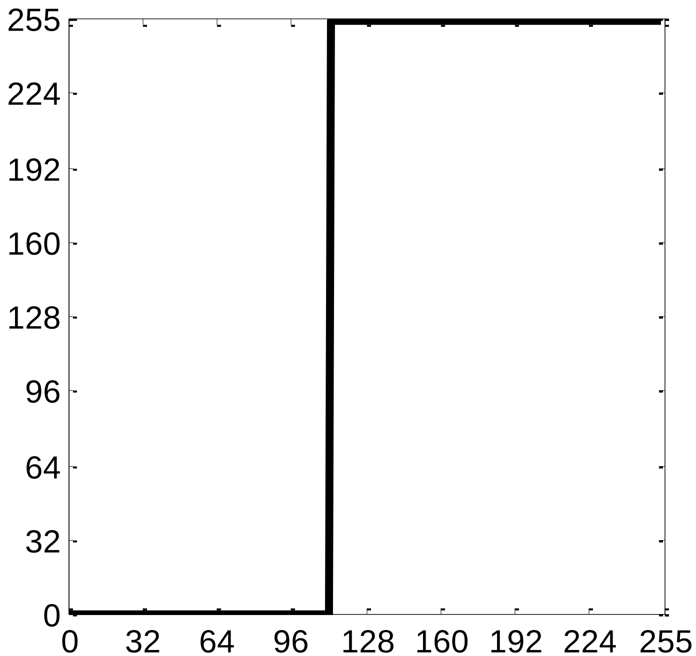
→ **Valeur entière de n'**

Amélioration d'images



Amélioration d'images

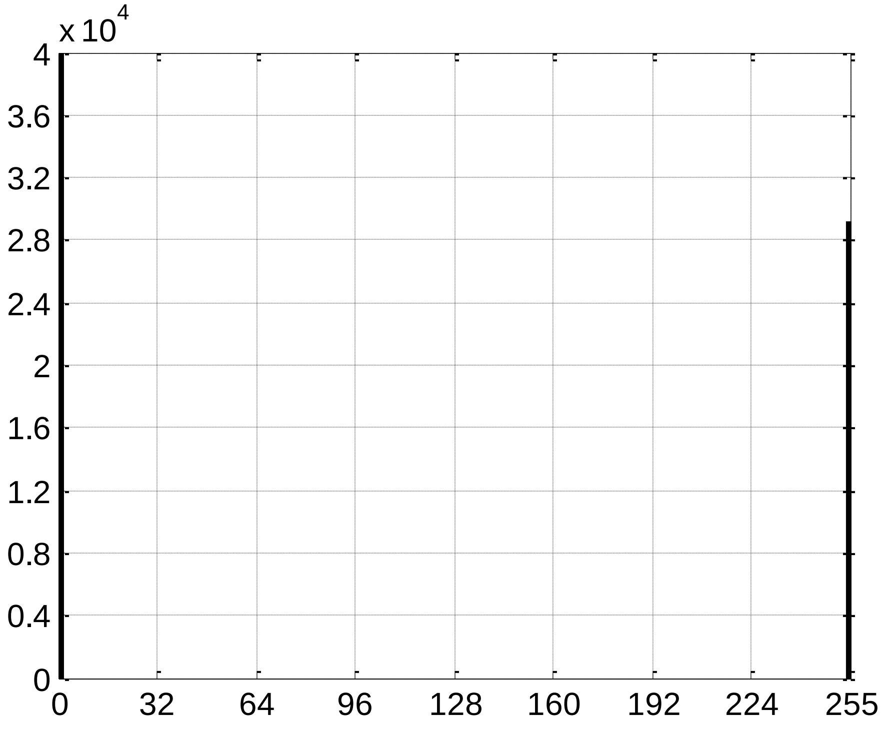
- Binarisation



$seuil$

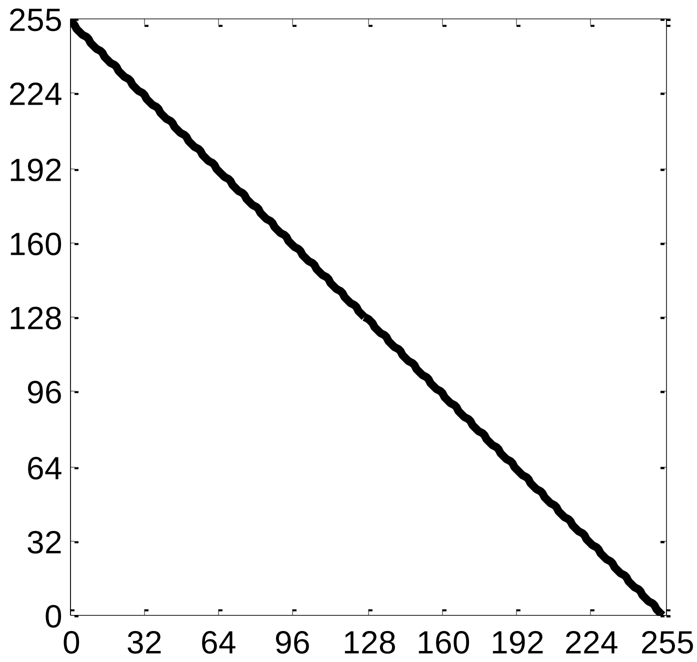
Si $n < seuil$ alors $n' = 0$
sinon $n' = 1$

Amélioration d’images



Amélioration d'images

- Inversion

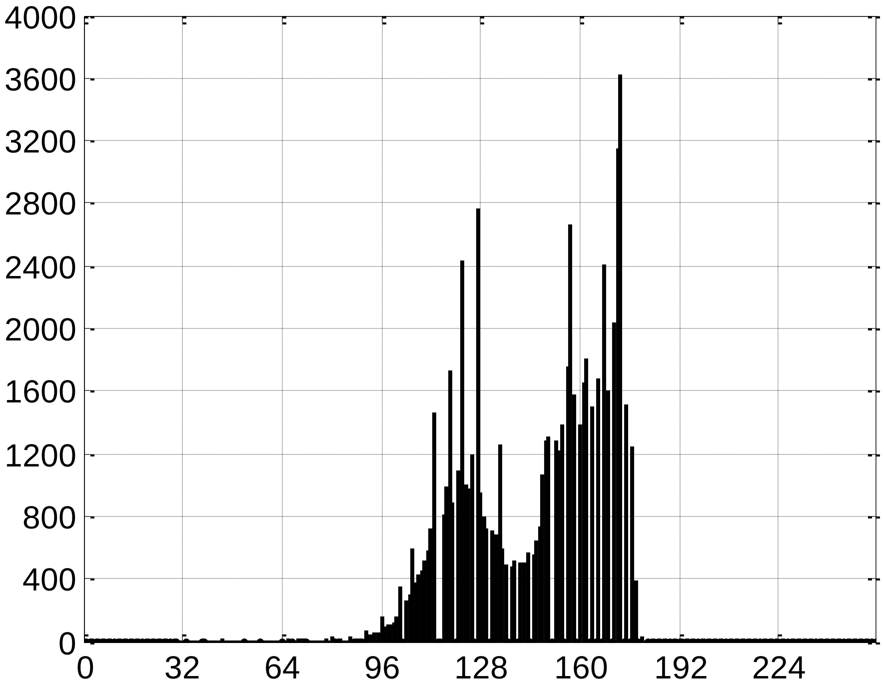


$$n' = n_{max} - n$$

$$n' = n \oplus 1$$

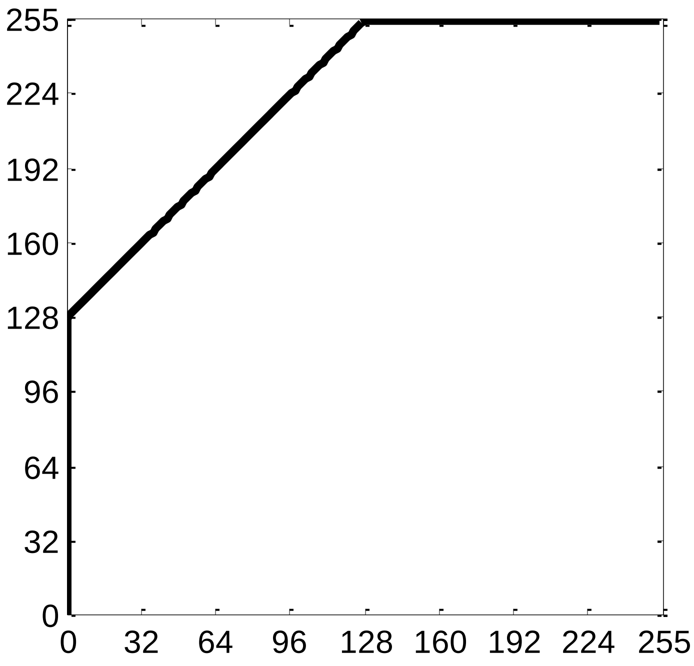
$$n' =\sim n$$

Amélioration d’images



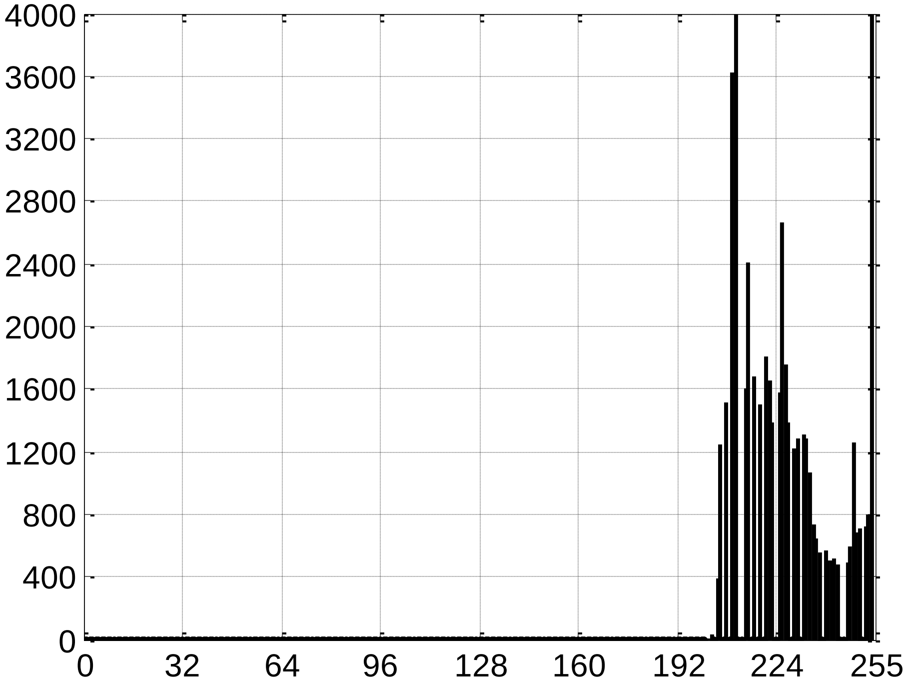
Amélioration d'images

- Offset ou décalage (addition : ajout d'une constante)



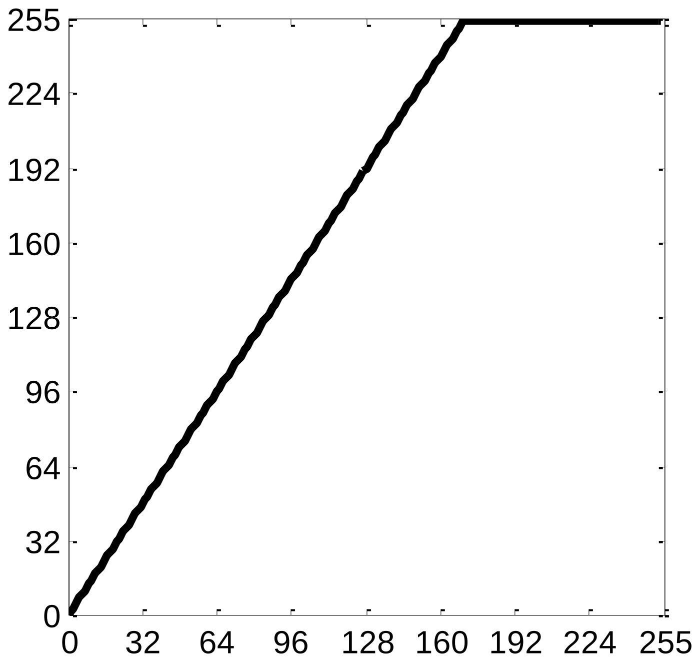
$$n' = n + offset$$

Amélioration d'images



Amélioration d'images

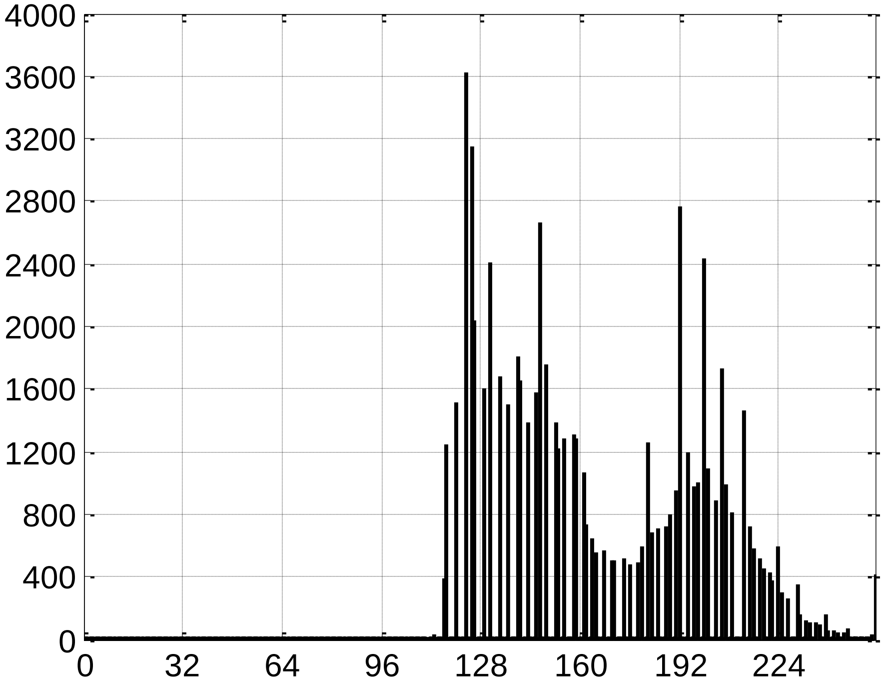
- Gain (Multiplication : produit par une constante)



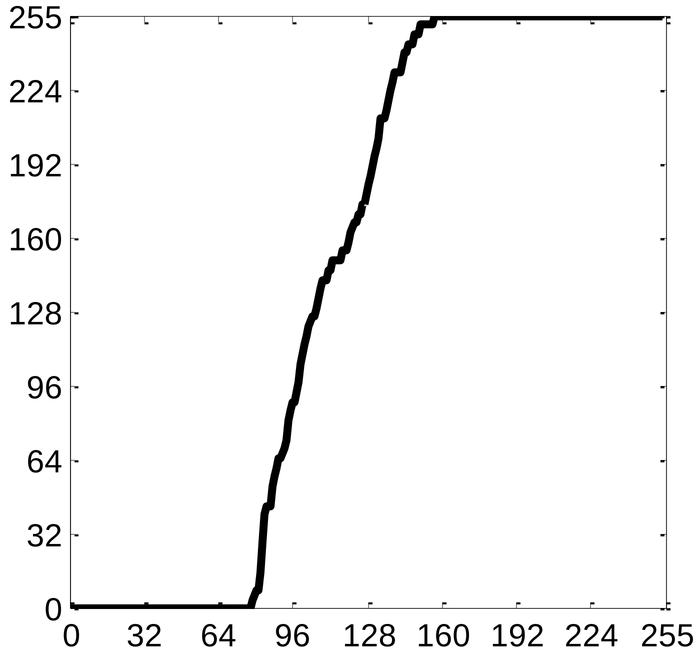
$$n' = n \times gain$$

→ **Valeur entière de n'**

Amélioration d'images



- Égalisation d'histogramme (linéarisation)



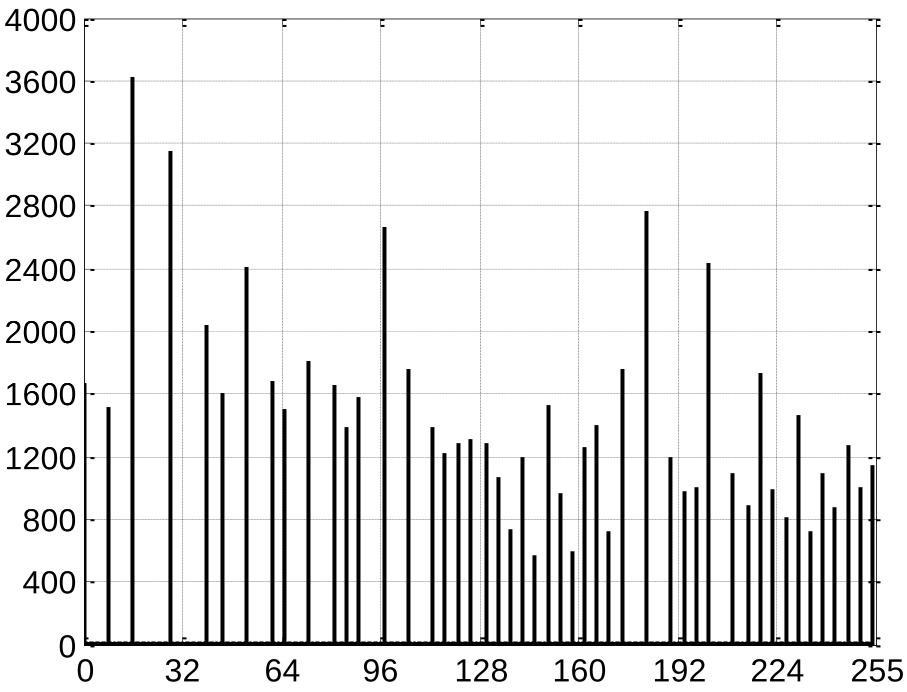
Le but est de rendre l'histogramme H aussi plat que possible.
Pour cela on utilise l'histogramme **cumulé** et **normalisé** comme fonction de transformation.

Pour $Q = 256$:

$$n' = 255 \times \sum_{j=0}^{n} \frac{H(j)}{\sum_{i=0}^{255} H(i)}$$

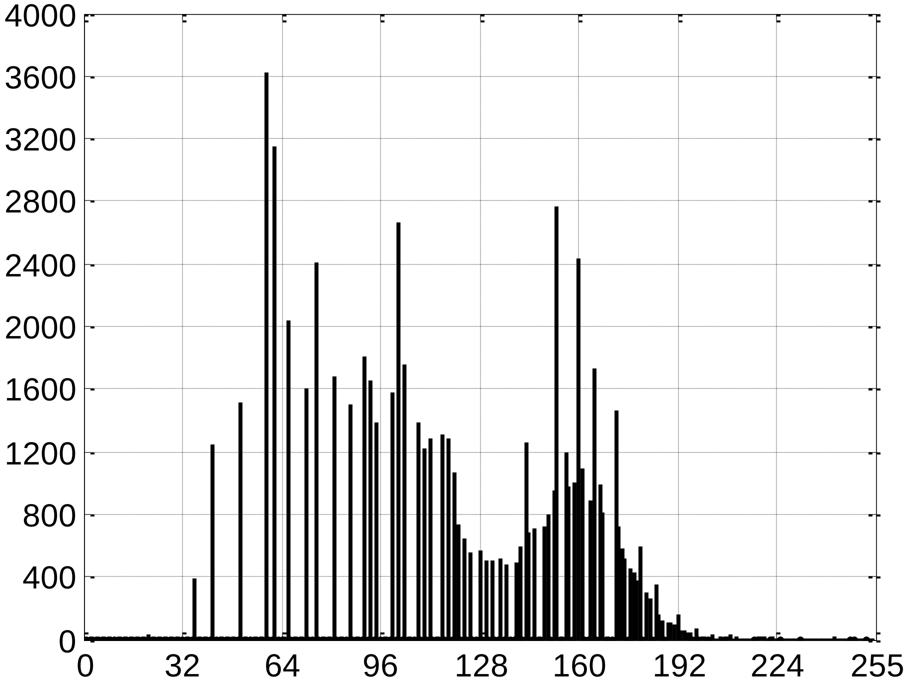
→ **Valeur entière de n'**

Amélioration d’images

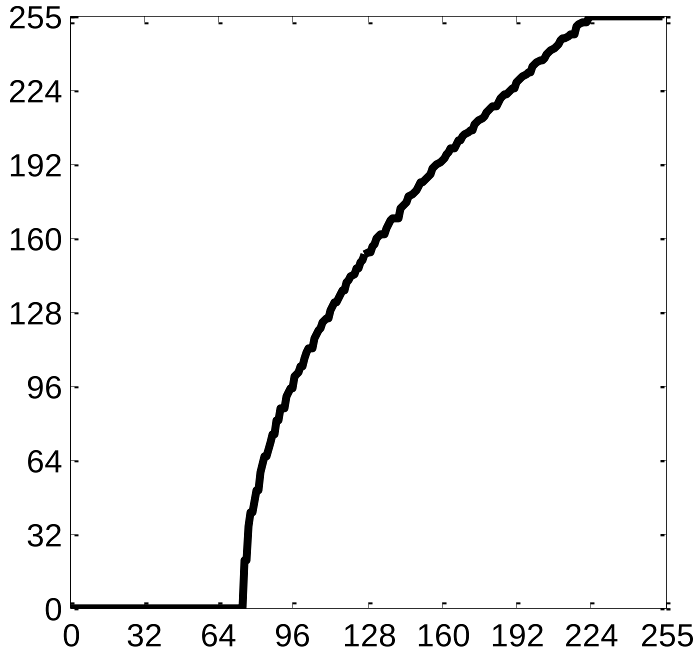


Amélioration d'images

- Spécification d'histogramme



Amélioration d'images



Le but est de faire ressembler l'histogramme à un histogramme de référence.
Pour cela on utilise l'histogramme de référence **cumulé** et **normalisé** comme fonction de transformation.

Amélioration d'images

- Le rehaussement de contraste
  - Convolution de l'image avec un filtre rehausseur

-1/6	-2/3	-1/6
-2/3	26/6	-2/3
-1/6	-2/3	-1/6

Amélioration d'images

- Le filtrage morphologique
  - Modification d'un ensemble géométrique
  - Application d'un élément de morphologie de géométrie connue appelé élément structurant centré en chaque pixel
  - Opérateur de la théorie des ensembles : union, intersection, inclusion, exclusion, complémentation de l'élément…
  - Utilisations des opérateurs *min* et *max* pour le traitement des images :
    - L'opérateur *min* appliqué dans le voisinage défini par l'élément structurant permet l'opération d'**érosion** (ou **rétrécissement**) : les zones claires de l'image se réduisent.
    - L'opérateur *max* appliqué dans le voisinage défini par l'élément structurant permet l'opération de **dilatation** (ou **expansion**) : les zones claires de l'image s'étendent.

Amélioration d'images

- Rehaussement par méthodes morphologiques
  - Érosion et dilation de l'image

Somme de l'image dilatée et érodée.
Comparaison du résultat avec l'image initiale :
Si la valeur du pixel traité est inférieur ou égale à la valeur calculée alors on garde le résultat de l'érosion sinon on garde le résultat de la dilatation.

Amélioration d'images

- Autres méthodes
  - Autres méthodes linéaires
  - Filtrage homomorphiques
  - Filtrage d'ordre adaptatifs
  - Autres méthodes morphologiques
  - Multiresolution
- Démo Matlab
  - Amélioration d'images : imadjdemo

Compression d'images

- Définition
  - Le but des algorithmes de compression est de réduire la taille des fichiers pour diminuer l'espace nécessaire à leur stockage sur le disque ou leur transfert par le réseau.
  - Ils encodent d'une manière différente les données de l'image afin de les rendre plus compactes.
  - Deux familles d'algorithmes de compression se distinguent :
    - La compression sans perte
    - La compression avec perte
  - Ratio de compression, σ :
  - Taux de compression, τ :
  - Gain de compression, γ :

$$\sigma = \frac{Taille\ image}{Taille\ image\ compressée}$$

$$\tau = \frac{Taille\ image\ compressée}{Taille\ image}$$

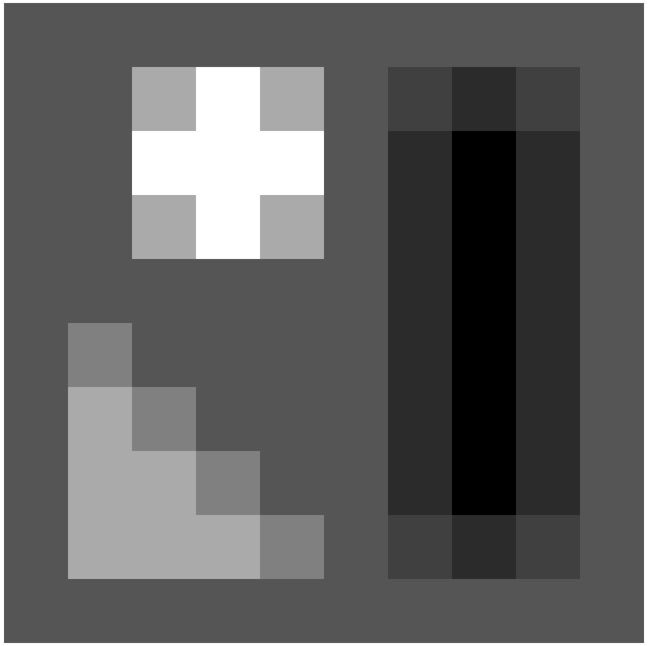
$$\gamma = 1 - \frac{Taille\ image\ compressée}{Taille\ image}$$

La compression sans perte

- Principe
  - La taille de l'image est réduite sans que l'information soit perdue.
  - L'image est indexée et chaque index est codé différemment.
  - Différentes approches :
    - RLE (Run Lenght Encoding) ou RLC
    - Shannon, Fano (Variable Code Lenght)
    - Huffman (Variable Code Lenght)
    - Lempel, Ziv, Welch (LZW)

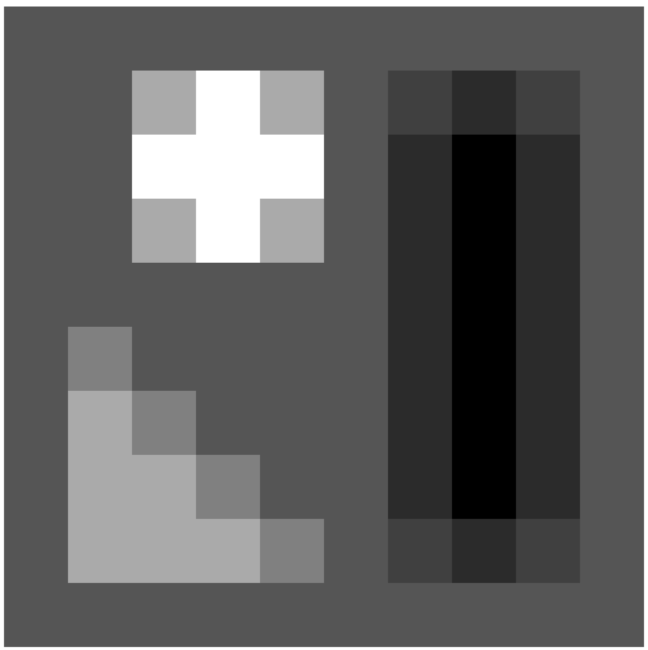
La compression sans perte

- La compression RLE



Taille image = 10×10×8 = 800 bits = 100 octets.
Si 3 éléments ou plus se répètent consécutivement, on utilise un octet pour indiquer le nombre de ces éléments qui se suivent et un octet pour indiquer la valeur.
Des codes séparateurs sont insérés.
L'image peut être parcouru dans des directions variables (balayage particulier).
Ici, on a 75 octets.

• Shannon - Fano



7 niveaux de gris sur 255 présents dans l’image triés en fonction de leur fréquence d'apparition :

64	128	255	0	170	43	85
4	4	5	6	10	14	57

Division en deux groupes d’égale population :

64	128	255	0	170	43		85
4	4	5	6	10	14		57
			0				**1**

La compression sans perte

64	128	255	0	170	43	85
4	4	5	6	10	14	57
00				01		1

64	128	255	0	170	43	85
4	4	5	6	10	14	57
000		001		010	011	1

64	128	255	0	170	43	85
4	4	5	6	10	14	57
0000	0001	0010	0011	010	011	1

Taille image = 1×57 + 3×14 + 3×10 + 4×6 + 4×5 + 4×4 + 4×4 = 205 bits

Plus entête de fichier (information, taille, table de codage, séparateur).

Code préfixé : chaque code ne peut être le début d'un autre.

La compression sans perte

- Huffman

64	128	255	0	170	43	85
4	4	5	6	10	14	57

Dans cette méthode, chaque code ne peut être le début d'un autre (code préfixé).

64	128	255	0	170	43	85
4	4	5	6	10	14	57
0	**1**					

Séparation des deux niveaux de plus faibles population

64	128	255	0	64-128	170	43	85
4	4	5	6	8	10	14	57
0	**1**						

Ajout d'un nœud correspondant aux niveaux séparés

La compression sans perte

64	128	255	0	64-128	170	255-0	43	85
4	4	5	6	8	10	11	14	57
0	1	0	1					

64	128	255	0	64-128	170	255-0	43	64-128-170	85
4	4	5	6	8	10	11	14	18	57
0	1	0	1	0	1				

64	128	255	0	64-128	170	255-0	43	64-128-170	255-0-43	85
4	4	5	6	8	10	11	14	18	25	57
0	1	0	1	0	1	0	1			

La compression sans perte

64	128	255	0	64-128	170	255-0	43	64-128-170	255-0-43		85
4	4	5	6	8	10	11	14	18	25	43	57
0	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**

64	128	255	0	170	43	85
0000	**0001**	**0100**	**0101**	**001**	**011**	**1**

La compression sans perte

- LZW
  - Construction d’un dictionnaire.
  - Le dictionnaire est alimenté par les séquences d’éléments différentes qui se répètent le plus souvent, ce qui permet d’indexer plusieurs pixels de même valeur dans des zones homogènes.
  - Chaque séquence est indexée.
  - Possibilité de l’associer aux méthodes statiques (ARJ, PKZIP).
  - Possibilité de ne pas transmettre le dictionnaire.

La compression avec perte

- Principe
  - La taille de l’image est nettement réduite mais au détriment d’une perte d’information.
  - Différentes approches :
    - Moyennage de blocs
    - Transformée linéaire optimale
    - Transformée en cosinus (JPEG)
    - Quantification vectorielle
    - Les ondelettes
    - Les fractales

La compression avec perte

- JPEG
  - L'image est décomposée en blocs (en général 8×8)
  - La transformée en cosinus discrète (DCT) est appliquée sur chaque bloc.
  - Les composantes fréquentielles de faible amplitude et de haute fréquence sont supprimées par une quantification des composantes qui est effectuée grâce à une matrice de quantification.
  - Un codage à longueur variable de type Huffman est enfin utilisé.

La compression avec perte

- Les ondelettes
  - Extension de l’analyse de Fourier
  - Décomposition en sous-bandes (de fréquences)
  - Transformation mathématique par projection sur des bases orthogonales
  - Traitement progressif (sous échantillonnage et passe-bas)
  - Fort taux de compression
  - Contrôle de la qualité et du taux de compression indépendant
  - Algorithme plus rapide que JPEG
  - Pas d’effet de mosaïque

La compression avec perte

- Les fractales
  - Le principe de la compression fractale est que toute image est la limite d'une séquence de transformations mathématiques (rotations, translations, changement d'échelle) appliquées à un ensemble de pixels.
  - La compression fractale permet d'atteindre des taux de compression très importants, et permet une reconstruction de l'image à toutes les tailles.
  - Son principal inconvénient est le temps de calcul nécessaire pour la compression.

La compression avec perte

- Evaluation de la qualité de la compression
  - Erreur quadratique moyenne, EQM :
    - $M \times N$, nombre de pixels de l'image I
    - I', image compressée

$$EQM = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(I(i,j) - I'(i,j)\right)^2$$

  - Rapport signal/bruit crête (*Peak Signal to Noise Ratio*), $PSNR$
    - Pour une image codée sur 8 bits (255 est la valeur crête) :

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{EQM}\right)$$