

# Prétraitement *Preprocessing*

DANIEL RACOCEANU

PROFESSEUR, SORBONNE UNIVERSITÉ  
[daniel.racoceanu@sorbonne-universite.fr](mailto:daniel.racoceanu@sorbonne-universite.fr)



1

## Plan

- a) Restauration d'images
- b) Amélioration d'images / *Image Enhancement*
- c) Compression d'images

2



2

## a) 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

3



3

## a) Restauration d'images

### Bruit

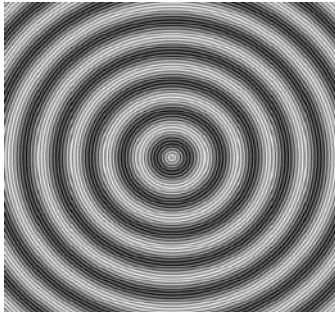
- Bruit lié au contexte de l'acquisition
  - ♦ Bougé, dérive lumineuse, flou, poussière, ...
- Bruit lié au capteur
  - ♦ Distorsion de la gamme des niveaux de gris, distorsion géométrique, mauvaise mise au point, ...
- Bruit lié à la numérisation
  - ♦ Codage, quantification, échantillonnage (moiré,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

5

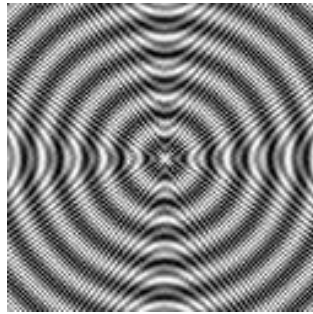


5

## Effet de Moiré / Moiré effect



original image

subsampled image  
(effet de moiré)

The Moiré effect is an embarrassing phenomenon. Structured patterns, unpleasant to the eye, appear in the image.

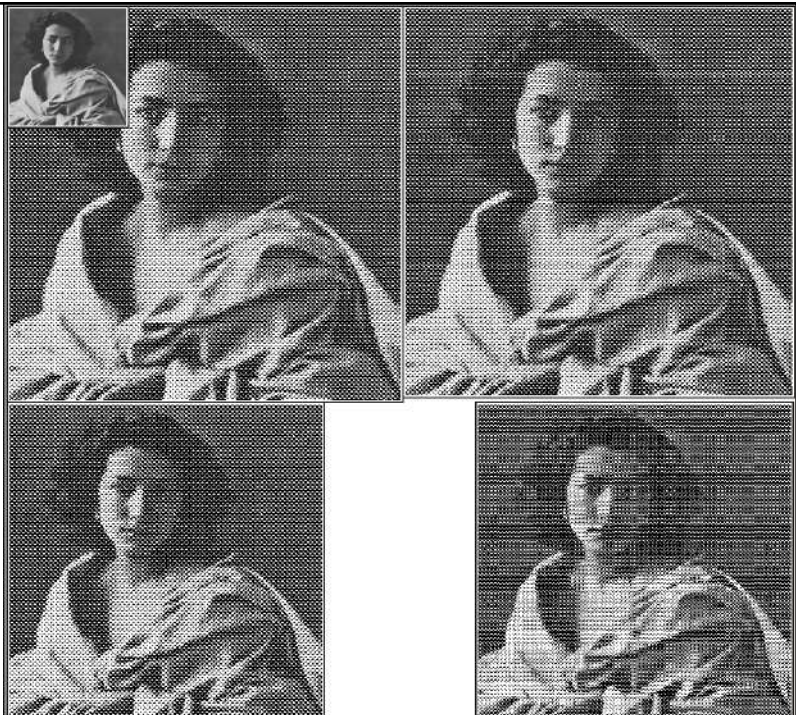
Le moiré est un phénomène gênant.  
Des motifs structurés, désagréables à l'œil, apparaissent dans l'image.

6

6

## Effet de Moiré / Moiré effect

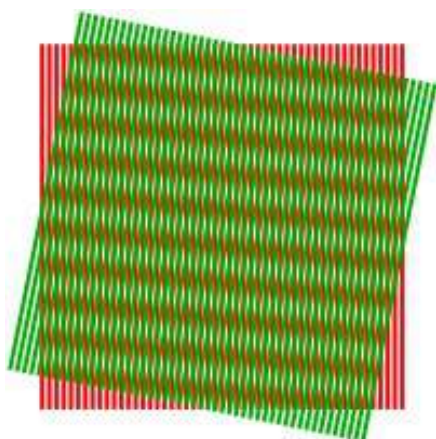
Différents effets de moiré selon le rapport entre les pixels et les points de l'image initiale (l'exemple ici est un portrait de [Sarah Bernhardt](#) par [Félix Nadar](#) en 1864).



7

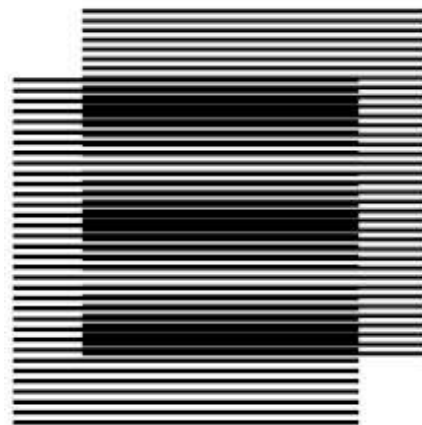
7

## Effet de Moiré / Moiré effect



Incidence de l'angle

8



Déplacement des lignes du moiré avec le déplacement d'un réseau par rapport à l'autre.

SORBONNE  
UNIVERSITÉ

8

## Bruit poivre et sel / Salt-and-pepper noise

Le **bruit poivre et sel** également appelé **bruit impulsif**, est une dégradation de l'image sous la forme de pixels noirs et blancs (d'où le nom poivre et sel) répartis d'une manière aléatoire dans l'image.

Ce bruit est dû soit à des erreurs de transmission de données, soit aux dysfonctionnement ou à la présence de particules fines sur les éléments du capteur de la camera ou à des emplacements mémoire défectueux dans le matériel

9



Fat-tail distributed or "impulsive" noise is sometimes called **salt-and-pepper noise** or spike noise. An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission, etc. It can be mostly eliminated by using dark frame subtraction, median filtering, combined median and mean filtering and interpolating around dark/bright pixels. Dead pixels in an LCD monitor produce a similar, but non-random, display.

SORBONNE  
UNIVERSITÉ

9

## Bruits : exemples



Image source



Flou de mise au point

10

10

## Bruits : exemples



Image source



Flou de bougé

11

11



## Bruits : exemples



Bruit uniforme (gaussien)



Bruit aléatoire (impulsionnel)

12

SORBONNE  
UNIVERSITÉ

12

Illustration de  
l'île du  
Pharaon, dans  
le nord du golfe  
d'Aqaba, au  
large de la  
péninsule du  
Sinaï de  
l'Égypte  
orientale. 1839.



13

13

La même image, restaurée : la luminosité et le contraste ont été augmentés, et le profil colorimétrique recalibré.

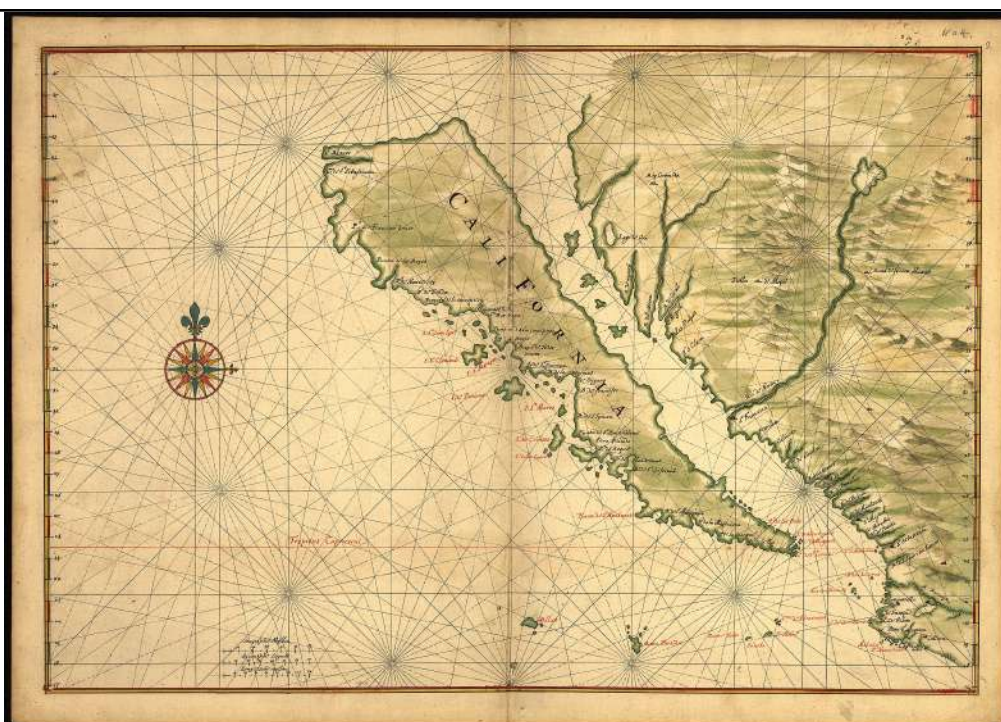
14



14

Une carte de Californie du xvii<sup>e</sup> siècle : le papier est jauni, les couleurs défraîchies. Sur cette carte, la Californie est représentée comme une île.

15



15



La même, après restauration : le jaunissement, les taches et les pliures du papier ont disparu. Le texte est plus lisible, les couleurs plus vives.

16



16

## a) Restauration d'images

### 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 restauration d'images, on utilise souvent l'opérateur de corrélation à condition d'avoir fait pivoter le noyau (ou masque) du filtre de 180°.

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

17

17



## a) Restauration d'images

Rotation d'un filtre de taille  $3 \times 3$

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

Filtre  $h$  de taille  $3 \times 3$

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

Filtre  $h_{\pi}$  de taille  $3 \times 3$

18

18

## a) Restauration d'images

Parcours d'un noyau de convolution  $K$  sur une image  $I$

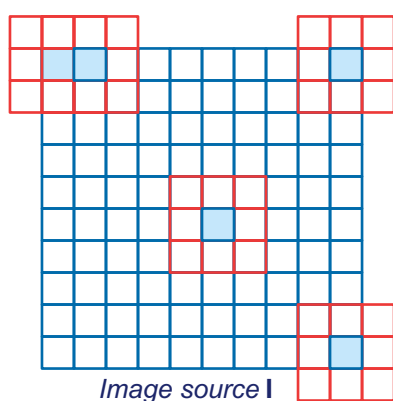


Image source  $I$



Noyau  $K$

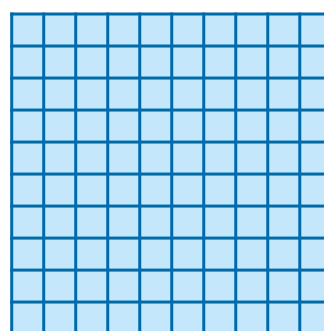


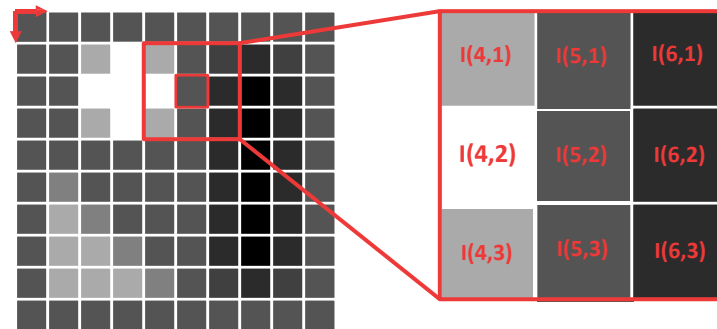
Image destination  $J$

19

19

## a) Restauration d'images

Exemple

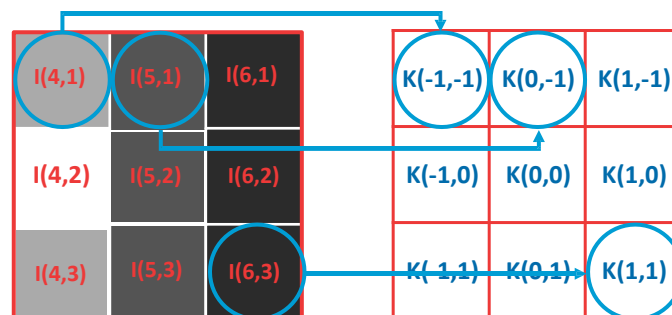


20

20

## a) Restauration d'images

Exemple



$$J_{\text{centr}} = J(5,2) = I(4,1)K(-1,-1) + I(5,1)K(0,-1) + I(6,1)K(1,-1) \\ + I(4,2)K(-1,0) + I(5,2)K(0,0) + I(6,2)K(1,0) \\ + I(4,3)K(-1,1) + I(5,3)K(0,1) + I(6,3)K(1,1)$$

21

21

## a) Restauration d'images

### ■ Notion de filtre passe-bas

- ◆ Il est utilisé pour atténuer les valeurs aberrantes de l'intensité des pixels. Il filtre les hautes fréquences spatiales (variations rapides des niveaux de gris dans le voisinage) 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
  - ...

22



22

## a) 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, dans le voisinage) comme les régions homogènes.
- ◆ Ils correspondent à l'approximation de dérivées premières ou secondes.
- ◆ Il existe différents types de noyaux :
  - Les gradients horizontaux ou verticaux qui sont des estimations de la dérivée première (le gradient dans une 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.

23

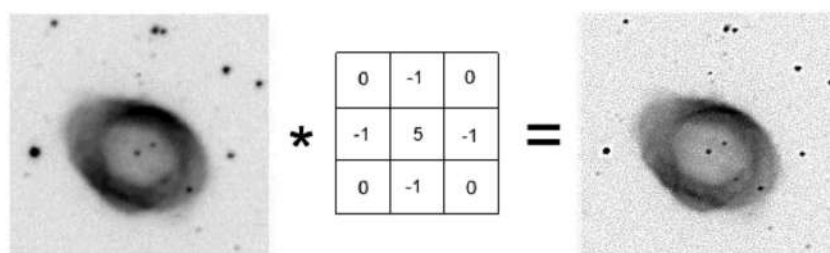


23



## Le filtre passe-haut

L'application principale des produits de convolution est la création des filtres "passe haut" et "passe bas". Un filtre "passe haut" favorise les hautes fréquences spatiales, comme les détails, et de ce fait, il améliore le contraste. Un filtre "passe haut" est caractérisé par un noyau comportant des valeurs négatives autour du pixel central, comme dans l'exemple ci-dessous:



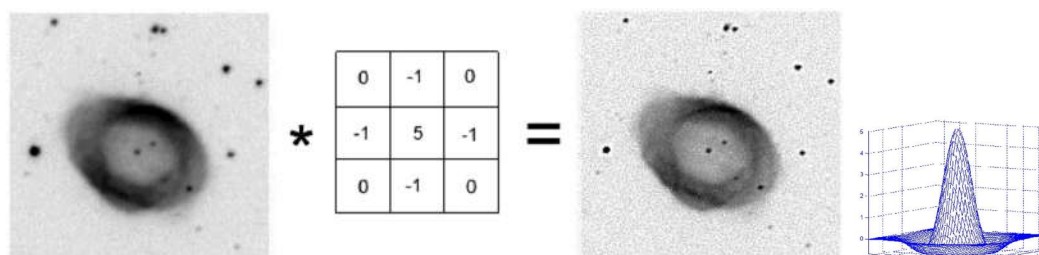
24

24

## Le filtre passe-haut

Effets secondaires :

- **Augmentation du bruit.** Particulièrement dans les images avec un rapport Signal Bruit faible, le filtre augmente le bruit granuleux dans l'image.
- **Effet couronne.** L'effet du filtre sur les objets stellaires est bien sur positif, et augmente le contraste et la différence par rapport au fond de ciel mais aussi des effets secondaires comme l'apparition de couronnes noires autour des étoiles. La cause de cet anneau noir est localisée dans le noyau du "passe-haut" : les valeurs négatives créent une sorte de dépression autour de l'étoile qui peut dégénérer en un anneau noir plus sombre que le fond de ciel.
- **Effet de bord.** Il est possible que sur les bords de l'image apparaisse un cadre. Mais cet effet est souvent négligeable et peut s'éliminer en tronquant les bords de l'image



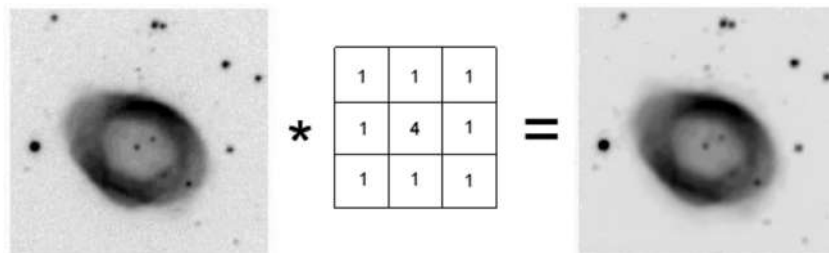
25

25

## Le filtre passe-bas

Les filtres “passe” bas agissent en sens inverse des filtres “passe haut” et le résultat est, un adoucissement des détails, ainsi qu'une réduction du bruit granuleux.

Par exemple le filtre passe-bas peut être appliqué dans la zone sombre du fond de ciel, pour réduire le bruit granuleux sans perdre les détails.



26

26

## a) Restauration d'images

Quelques filtres usuels

- ♦ Filtre moyenneur (lissage) : c'est un filtre passe-bas défini par  $h$  :

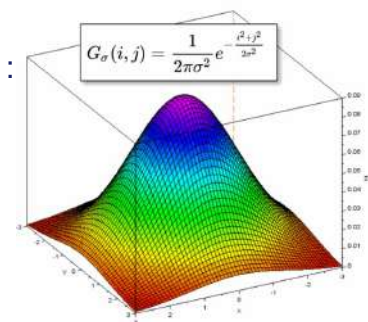
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

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  $h$  :

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

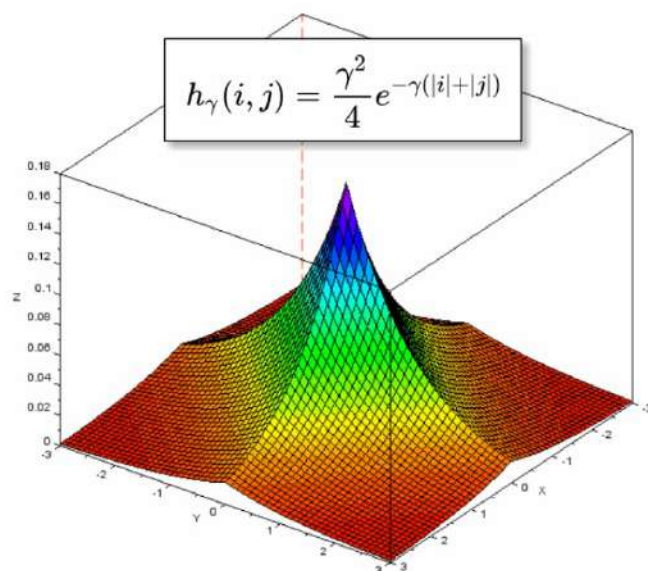
0,011	0,084	0,011
0,084	0,619	0,084
0,011	0,084	0,011



27

27

## Filtre passe-bas exponentiel

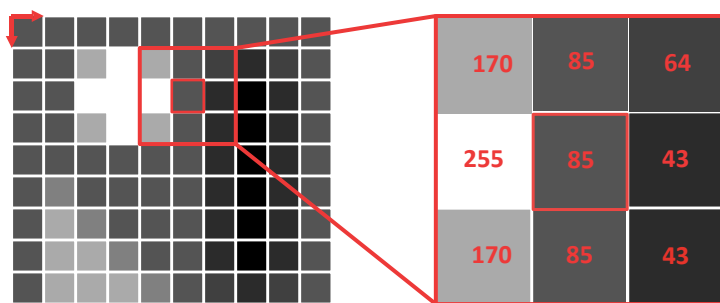


28

28

## a) Restauration d'images

Exemple avec un filtre moyenneur



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

30

30



## a) Restauration d'images

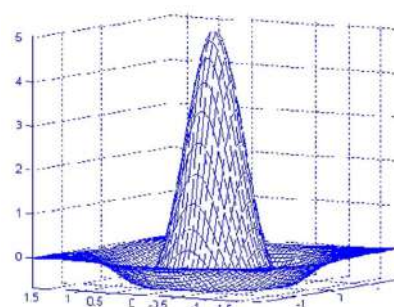
- ◆ Filtre Laplacien : c'est un filtre passe-haut défini par  $h_\pi$  :

-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



31

31

## a) Restauration d'images

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

-1				-1	-1	-1		
0	*	1	1	1	=	0	0	0
1						1	1	1

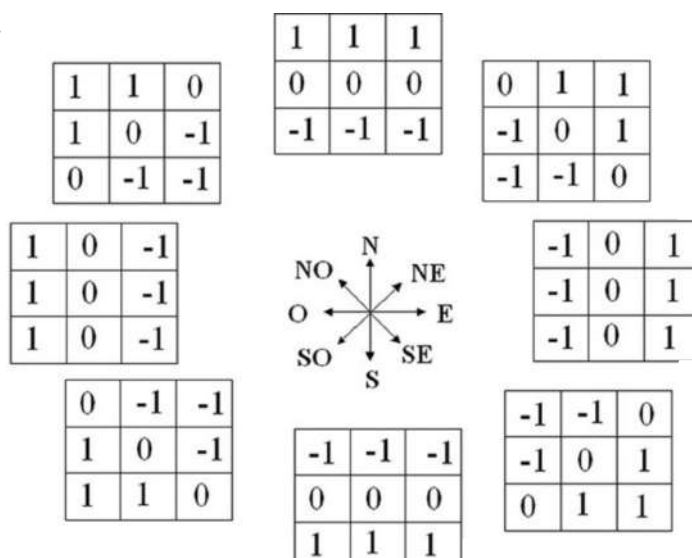
1				-1	0	1		
0	*	-1	0	1	=	-1	0	1
1						-1	0	1

32

32

## Masque de Robinson

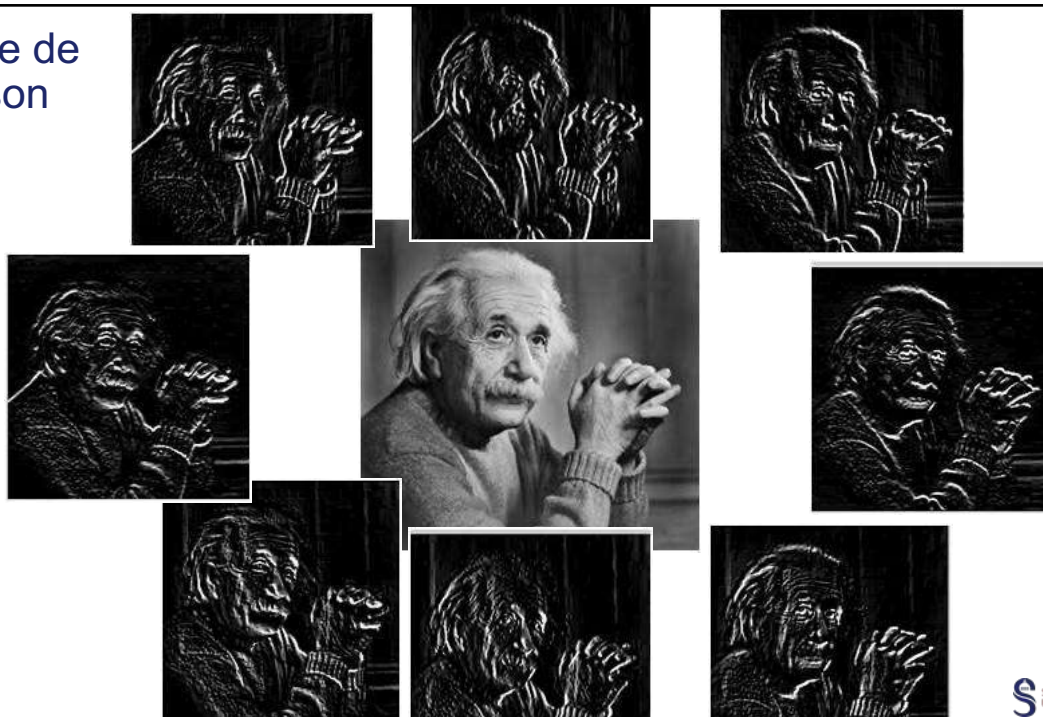
Exemple d'opérateurs  
Gradient boussole avec  
le *masque de Robinson*  
de niveau 3



33

33

## Masque de Robinson



34

34

## a) Restauration d'images

Application



*Filtre Gaussien*



*Filtre moyennneur*

35

35

## a) Restauration d'images

Application



*Filtre laplacien*



*Filtre rehausseur*

36

36



## a) Restauration d'images

Application



*Filtre de Prewitt vertical*



*Filtre de Sobel horizontal*

37

37

## a) 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 à 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).

38

38

## a) Restauration d'images

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

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

- 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 :

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

39

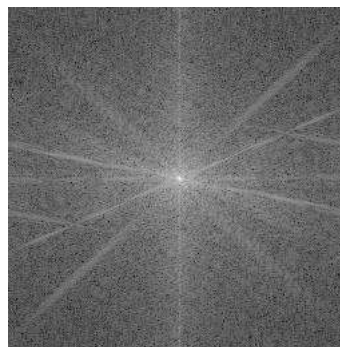
39

## Transformée de Fourier

FFT (Fast Fourier Transform) d'une image



Image I



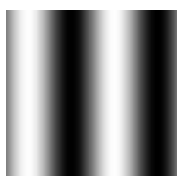
FFT de I

40

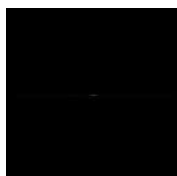
40

## Transformée de Fourier

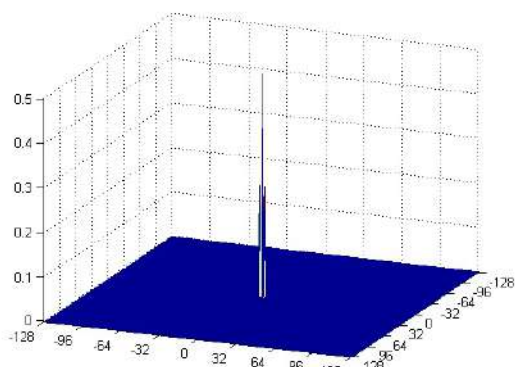
Fréquence spatiale (image 256 × 256)



*Image sinus (période 128)*



*Image de la FFT (fréquence 2)*

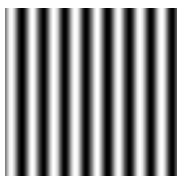


*Module de la transformée de Fourier*

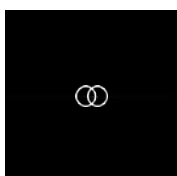
41

41

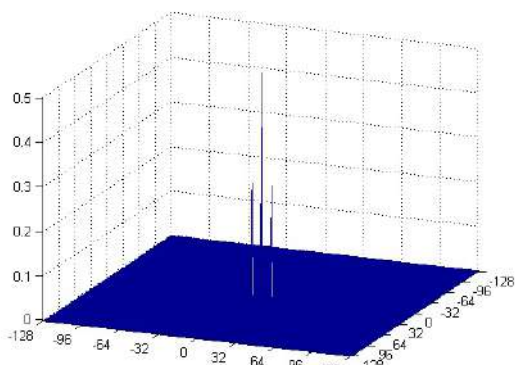
## Transformée de Fourier



*Image sinus (période 32)*



*Image de la FFT (fréquence 8)*



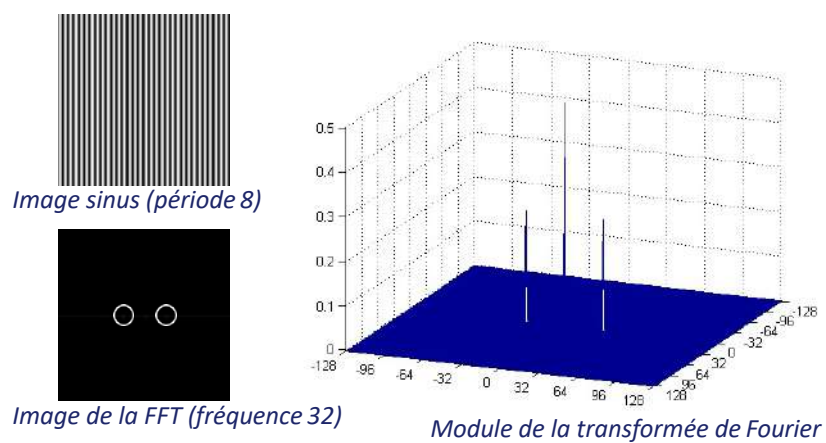
*Module de la transformée de Fourier*

42

42



## Transformée de Fourier

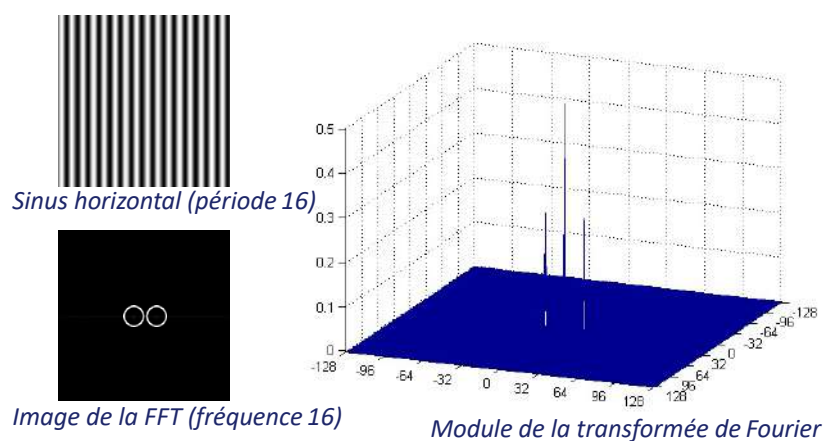


43

43

## Transformée de Fourier

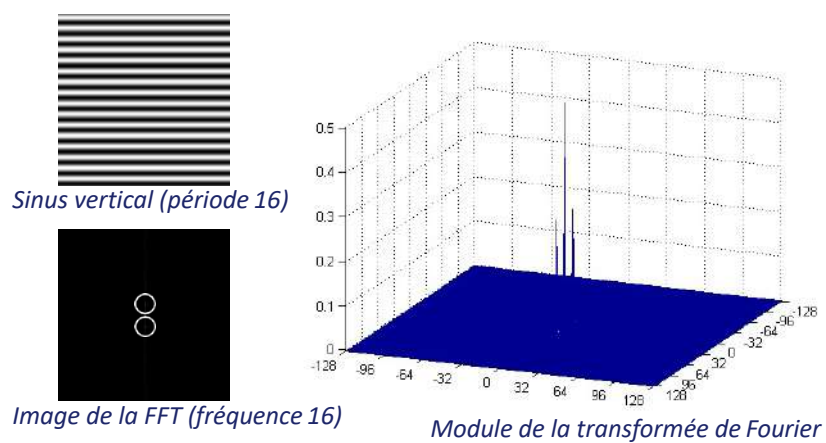
### ■ Orientation



44

44

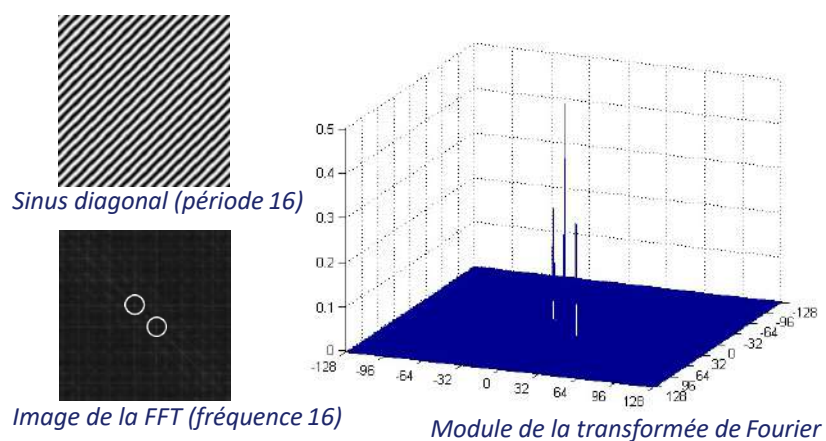
## Transformée de Fourier



45

45

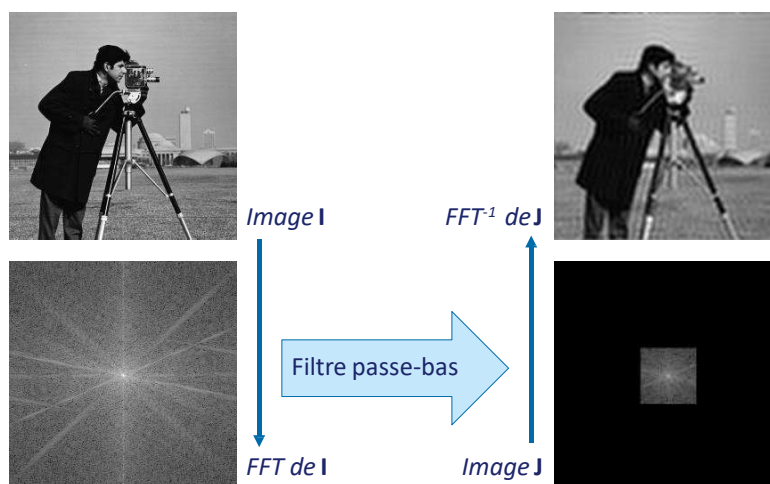
## Transformée de Fourier



46

46

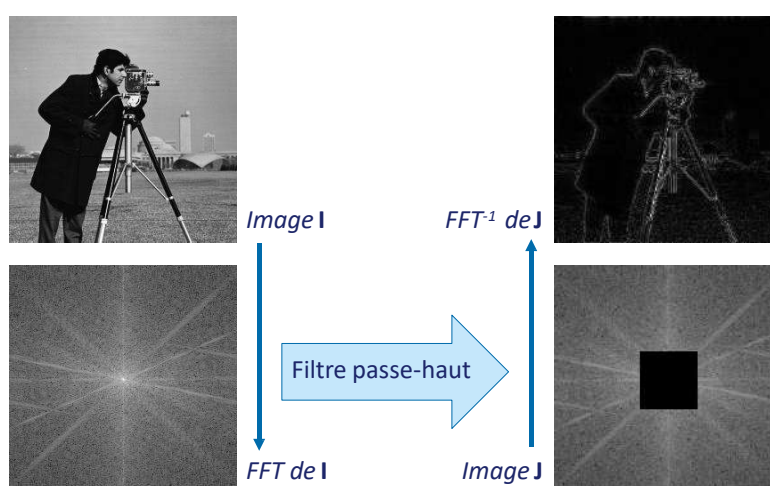
## Transformée de Fourier



48

48

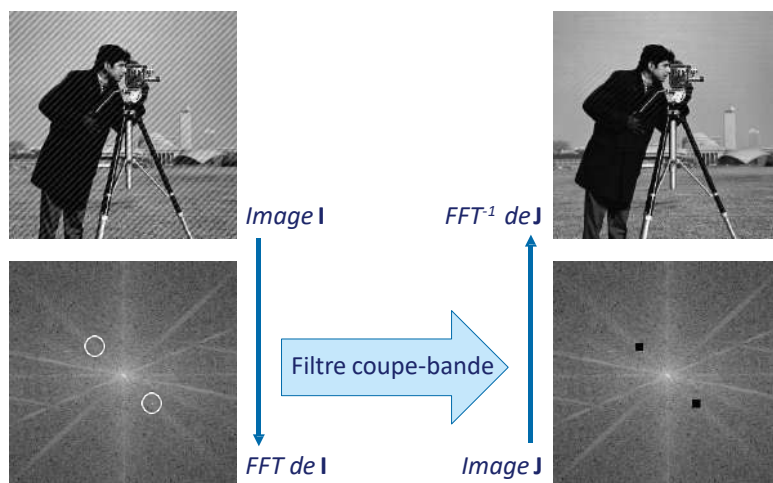
## Transformée de Fourier



49

49

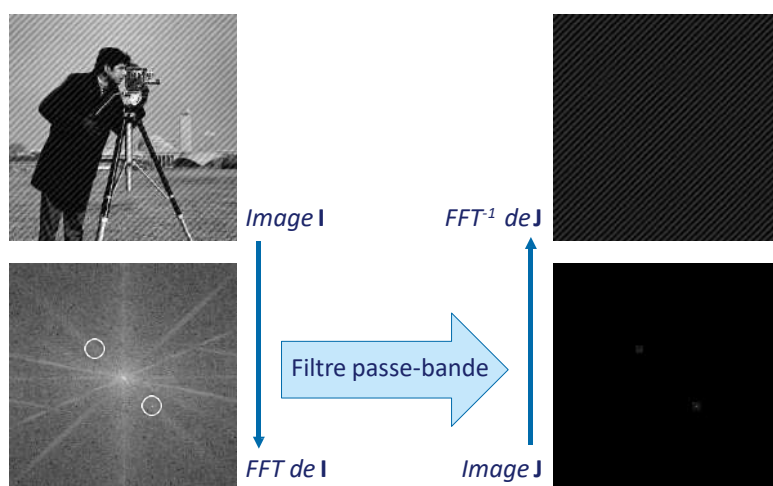
## Transformée de Fourier



50

50

## Transformée de Fourier

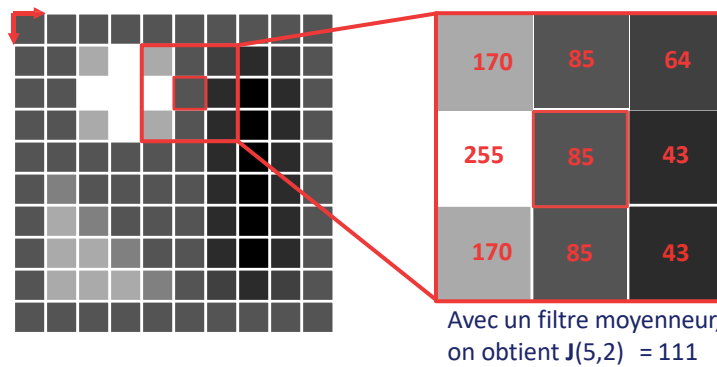


51

51

## a) Restauration d'images

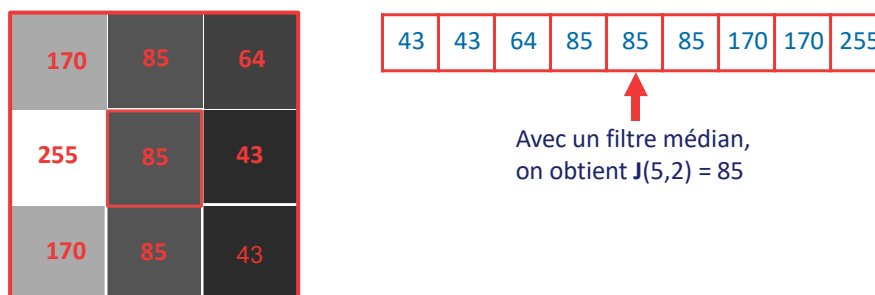
- Filtrage non linéaire
  - Filtre médian



52

52

## a) Restauration d'images



53

53



## a) Restauration d'images

### ■ Lissage robuste

170	85	64
255	85	43
170	85	43

Demi intervalle le plus compact

$$85 - 43 = 42$$

$$J_{\text{centr}} = J(5,2) = 85$$

255	170	170	85	85	85	64	43	43
-----	-----	-----	----	----	----	----	----	----

255	170	170	85	85
-----	-----	-----	----	----

→ 170 (= 255-85)

170	170	85	85	85
-----	-----	----	----	----

→ 85 (= 170-85)

170	85	85	85	64
-----	----	----	----	----

→ 106 (=170-64)

85	85	85	64	43
----	----	----	----	----

→ 42 (=85-43)

85	85	64	43	43
----	----	----	----	----

54

54

## a) Restauration d'images

### ■ Lissage min/max

170	85	64
255	85	43
170	85	43

Valeur minimale,  $m = 43$

Valeur maximale,  $M = 255$

Si  $J_{\text{centr}} < (M - m)/2$ , alors  $J_{\text{centr}} = m$

Si  $J_{\text{centr}} > (M - m)/2$ , alors  $J_{\text{centr}} = M$

Ici  $(M - m)/2 = 106$ , donc  $J_{\text{centr}} = 43$

55

55

## a) Restauration d'images

### ■ Lissage SNN (Symetric Nearest Neighbor)

170	85	64
255	85	43
170	85	43

Après avoir gardé 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  $J_{\text{centr}} = 64$

56

56

## a) Restauration d'images

### ■ 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

1/85	0	1/21
1/170	0	1/42
1/85	0	1/42

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

57

57

## Bruit impulsionnel / bruit uniforme



58

*Bruit impulsionnel*



*Bruit uniforme*



58

## Filtre moyennneur



59

*Bruitée*



*Moyennée*



59

## Filtre médian



*Bruitée*



*Filtre médian*

60

60

## Lissage robuste



*Bruitée*



*Lissée*

61

61

## Lissage min/max



*Bruitée*



*Lissée*

62

62

## Lissage SNN (Symmetric Nearest Neighbor)

•Description : atténuation du bruit avec conservation des contours

•Principe : calculer la valeur moyenne des pixels voisins qui ont une valeur proche du pixel central.

•Pour chacune des 4 paires de voisins, on conserve la valeur qui est la plus proche du pixel central. On fait ensuite la moyenne des 4 valeurs.



*Bruitée*



*Lissée*

63

63



## Lissage adaptatif



64

SORBONNE  
UNIVERSITÉ

64

## a) Restauration d'images

### Autres filtres

- ◆ Autres filtres d'ordre
- ◆ Filtres homomorphiques
- ◆ Filtres morphologiques
- ◆ Autres filtres adaptatifs

65

SORBONNE  
UNIVERSITÉ

65

## b) Amélioration d'images

### Définition

- L'amélioration d'images est le processus d'ajustement des images numériques afin que les résultats soient plus adaptés à l'affichage ou à une analyse d'image plus poussée. Par exemple, vous pouvez supprimer le bruit, accentuer ou éclaircir une image, ce qui facilite l'identification des fonctionnalités clés.
- L'amélioration a pour but de satisfaire l'oeil de l'observateur humain, 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

66

66

## b) Amélioration d'images / image enhancement

### 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

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

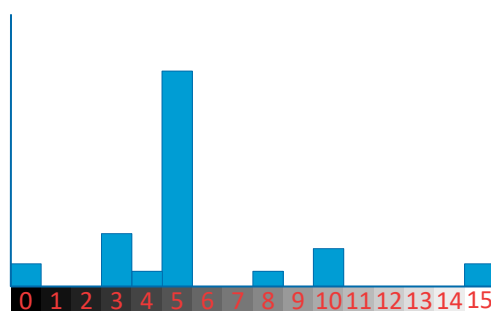
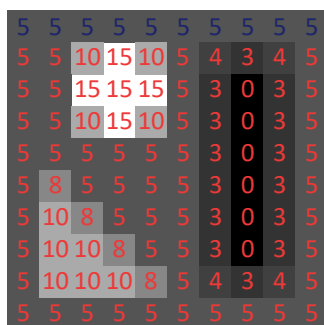
68

68

## b) Amélioration d'images

- La modification d'histogramme

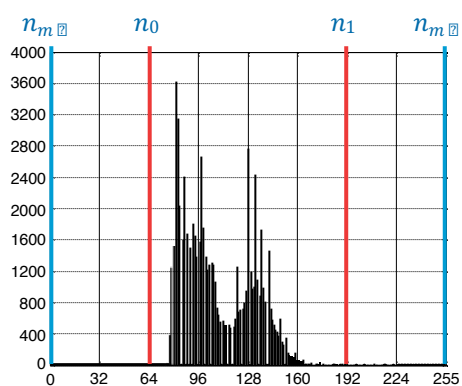
- L'histogramme



69

69

## Recadrage dynamique

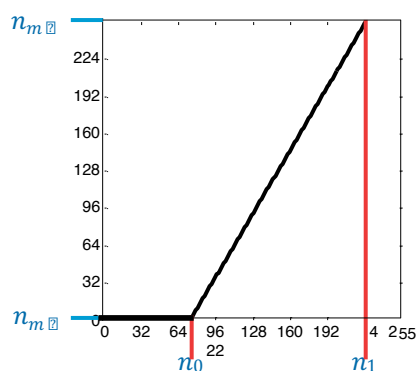


70

70

## Recadrage dynamique

### ■ Recadrage dynamique



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

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

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

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

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

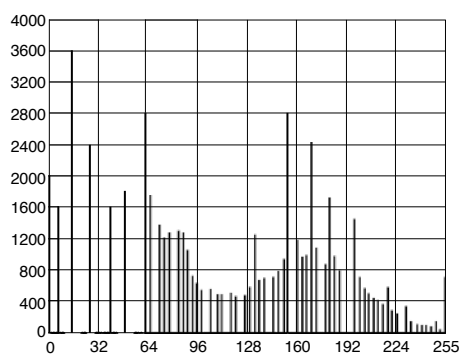
$$\Rightarrow n' = 255 \times \frac{n - n_0}{n_1 - n_0}$$

→ Valeur entière de  $n'$

71

71

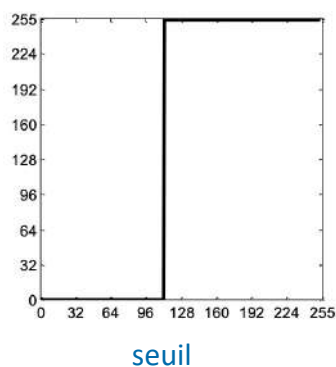
## Recadrage dynamique



72

72

## Binarisation

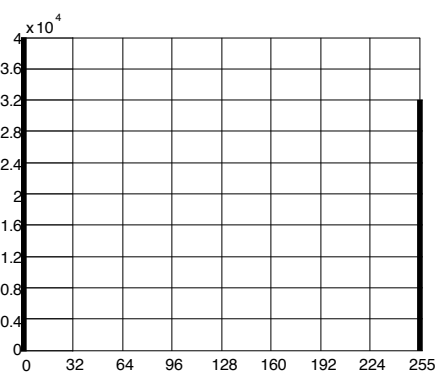


Si  $n < \text{seuil}$  alors  $n' = 0$   
sinon  $n' = 1$

75

75

## Binarisation

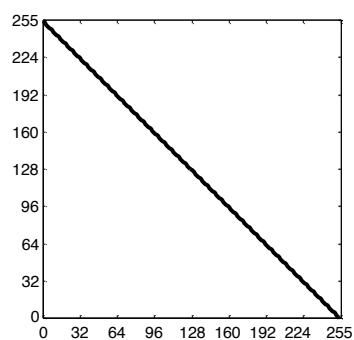


76

76



## Inversion



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

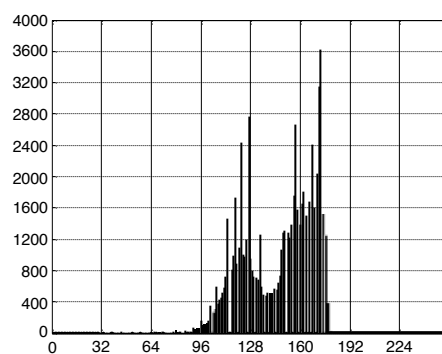
$$n' = n \oplus 1$$

$$n' = \sim n$$

77

77

## Inversion

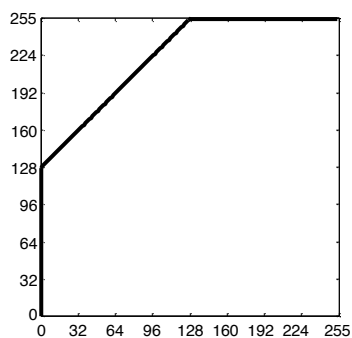


78

78

## b) Amélioration d'images

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

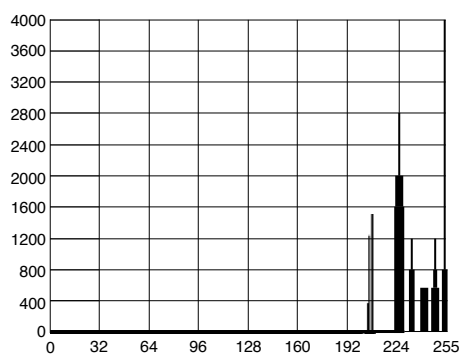


$$n' = n + offset$$

79

79

## b) Amélioration d'images

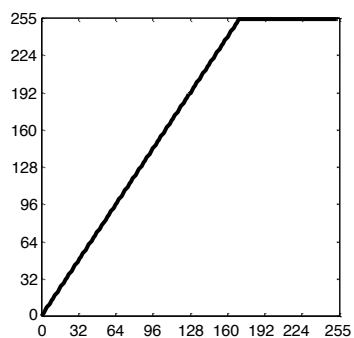


80

80

## b) Amélioration d'images

- Gain (Multiplication : produit par une constante)



$$n' = n \times \text{gain}$$

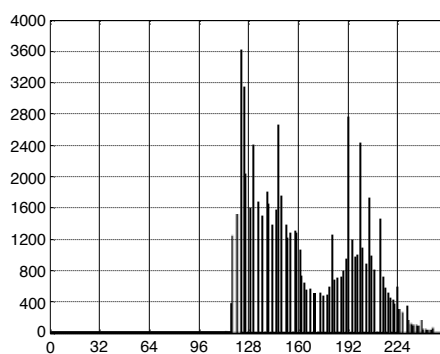
→ Valeur entière de  $n'$

81



81

## b) Amélioration d'images



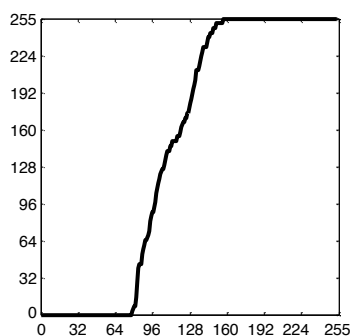
82



82

## b) 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  $Q = 256$ :

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

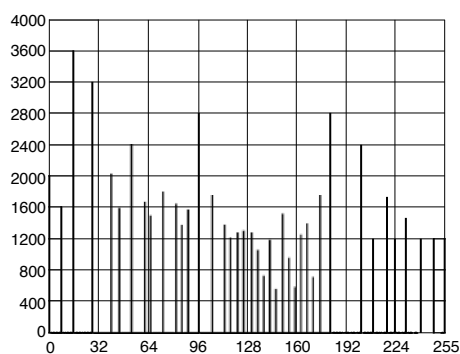
→ Valeur entière de  $n'$

83



83

## b) Amélioration d'images



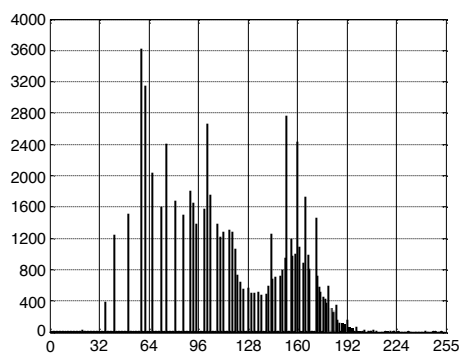
84



84

## b) Amélioration d'images

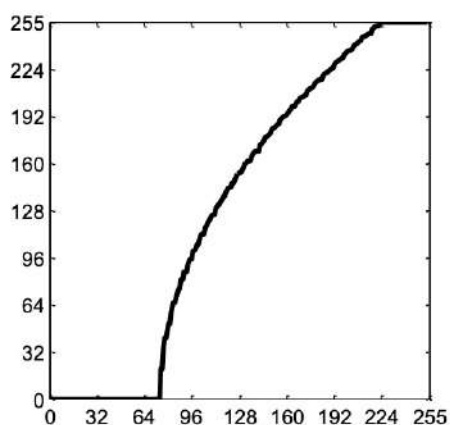
### ■ Spécification d'histogramme



85

85

## b) 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.

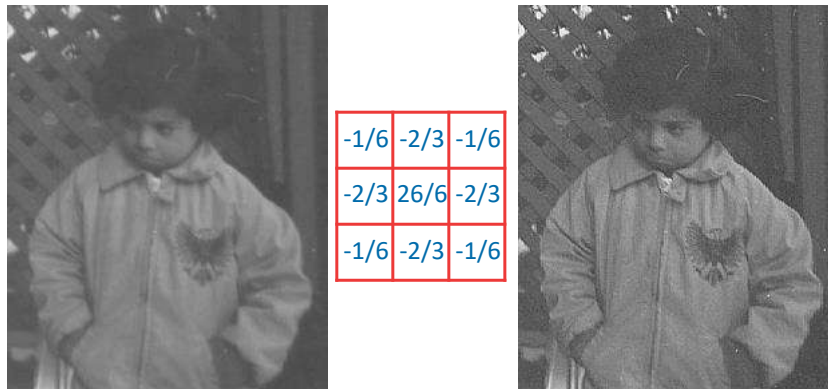
86

86



## b) Amélioration d'images

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



87

87

## b) 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.

88

88

## b) 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 inférieure ou égale à la valeur calculée alors on garde le résultat de l'érosion sinon on garde le résultat de la dilatation.*



89

S SORBONNE  
UNIVERSITÉ

89

## b) Amélioration d'images

### Autres méthodes

- ◆ Autres méthodes linéaires
- ◆ Filtrage homomorphiques
- ◆ Filtrage d'ordre adaptatifs
- ◆ Autres méthodes morphologiques
- ◆ Multiresolution

90

S SORBONNE  
UNIVERSITÉ

90

## c) Compression d'images

La **compression d'image** est une application de la compression de données sur des images numériques. Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement.

La compression d'image peut être effectuée **avec perte** de données ou **sans perte**.

La compression sans perte est souvent préférée là où la netteté des traits est primordiale : schémas, dessins techniques, icônes, bandes dessinées. La compression avec perte, plus radicale, est utile pour les transmissions à bas débit, mais dégrade la qualité de l'image restituée. Les méthodes de compression sans perte sont également préférées là où la précision est vitale : balayages médicaux ou numérisations d'images pour archivage.

Les méthodes avec perte restent acceptables pour des photos dans les applications où une perte mineure de fidélité (parfois imperceptible) est tolérée pour réduire les coûts de stockage ou d'envoi.

Pourquoi peut-on compresser ? Parce qu'une image où chaque point serait parfaitement indépendant des autres n'aurait pour nous pas d'intérêt : une image ne nous est utile que si elle contient des corrélations, qui dès lors qu'elles existent peuvent permettre les compressions en question.

91



91

## c) 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,  $\sigma$ :

▪ Taux de compression,  $T$ :

▪ Gain de compression,  $\gamma$ :

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

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

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

93



93

## 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 Length Encoding) ou RLC (Run Length Coding)
  - Shannon, Fano (Variable Code Length)
  - Huffman (Variable Code Length)
  - Lempel, Ziv, Welch (LZW)

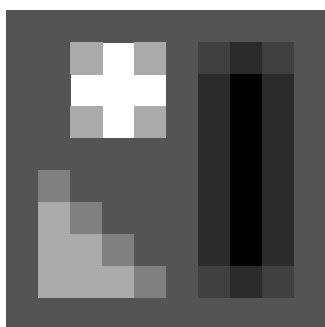
94



94

## Compression sans perte

### La compression RLE



Taille image =  $10 \times 10 \times 8 = 800$  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 parcourue dans des directions variables (balayage particulier). Ici, on a 75 octets.

95

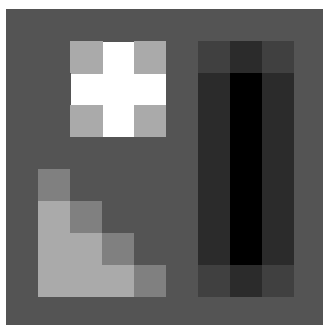
110



95

## Compression sans perte

### • 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

96



96

## 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 =  $1 \times 57 + 3 \times 14 + 3 \times 10 + 4 \times 6 + 4 \times 5 + 4 \times 4 + 4 \times 4 = 205$  bits

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

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

97



97

## 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

98



98

## 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			

99



99



## 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

100



100

## 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.

101



101

## Compression sans 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

102



102

## La compression AVEC perte

### JPEG

- L'image est décomposée en blocs (en général  $8 \times 8$ )
- 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.

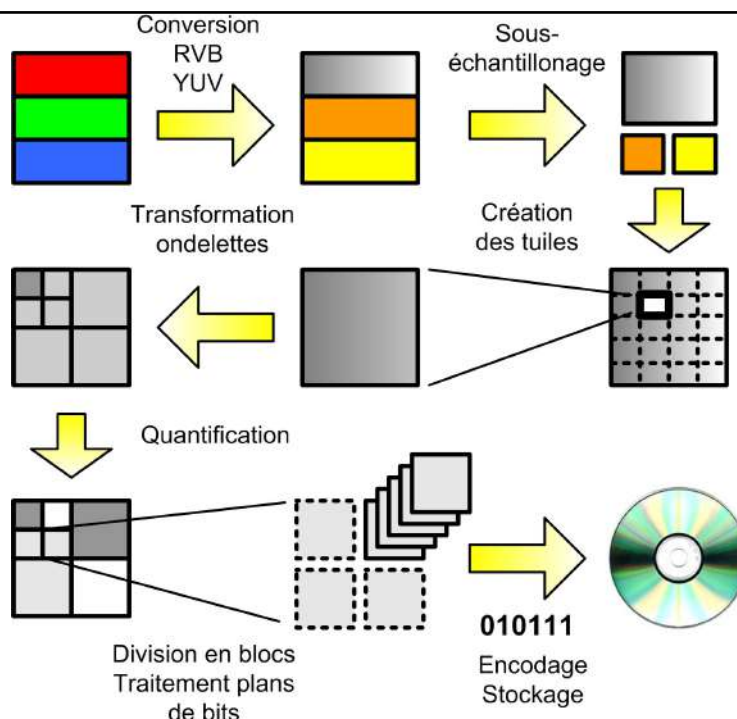
103



103

## La compression AVEC perte

### JPEG



104

104

## La compression AVEC perte

### Les ondelettes

- Apparues au début des années 80
- 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

105

105

## Ondelettes (ex 1 : limites de la théorie de Fourier)

Les séries de Fourier sont habituellement utilisées pour estimer le spectre des fréquences d'un signal donné en fonction du temps. En médecine, l'électrocardiogramme d'un patient malade diffère de celui d'un patient sain. Cette différence, parfois très difficile à repérer lorsque l'électrocardiogramme est donné en fonction du temps, devient évidente lorsque celui-ci est donné en fonction des fréquences, c'est-à-dire lorsqu'on regarde ses coefficients de Fourier. Mais les séries de Fourier donnent la quantité de chaque fréquence présente dans le signal pour l'ensemble de la période d'observation.

La théorie de Fourier devient donc inefficace pour un signal dont le spectre des fréquences varie considérablement dans le temps.

106



106

## Exemple 2 : limites de la théorie de Fourier

Les séries de Fourier sont aussi utilisées pour approximer des fonctions. Certaines fonctions régulières ont une série de Fourier économique, autrement dit elles sont bien approximées avec peu de coefficients de Fourier. Mais comme les fonctions sinus et cosinus ont un support infini, la série de Fourier ne fonctionne pas bien quand elle doit décrire localement une fonction qui montre des discontinuités.

Contrairement à la transformée de Fourier, l'analyse par ondelettes offre une large gamme de fonctions de base parmi lesquelles on peut choisir la plus appropriée pour une application donnée.

107

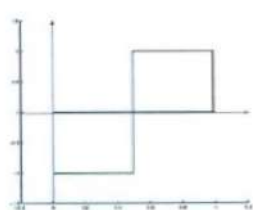


107

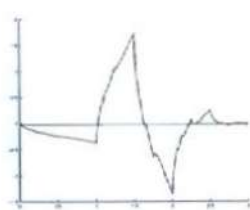
## Ondelettes

Une ondelette est une fonction oscillante (ce qui explique le mot "onde") de moyenne nulle, appelée  $\psi$ , possédant un certain degré de régularité et dont le support est fini (ce qui explique le mot "ondelette", qui veut dire petite onde).

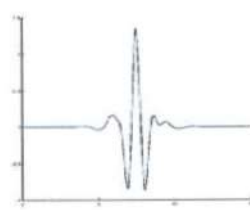
Exemples d'ondelette  $\psi(t)$  :



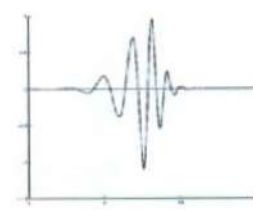
Haar



2-Daubechies



8-Symmet



8-Daubechies

108

## Ondelettes

L'ondelette mère  $\psi$  génère une famille d'ondelettes :

$$\{\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right)\}_{(u,s) \in \mathbb{R} \times \mathbb{R}_0^+}$$

où  $u$  est le paramètre du temps (délai) et  $s$  le paramètre d'échelle.

L'ondelette  $\psi_{u,s}$  est simplement l'ondelette mère  $\psi$  translatée de  $u$  et dilatée (contractée si  $s < 1$ ) par  $s$ .

Par conséquent, quand l'échelle  $s$  augmente, la résolution augmente.

Ceci veut dire que le support de la partie non-nulle de l'ondelette mère augmente.

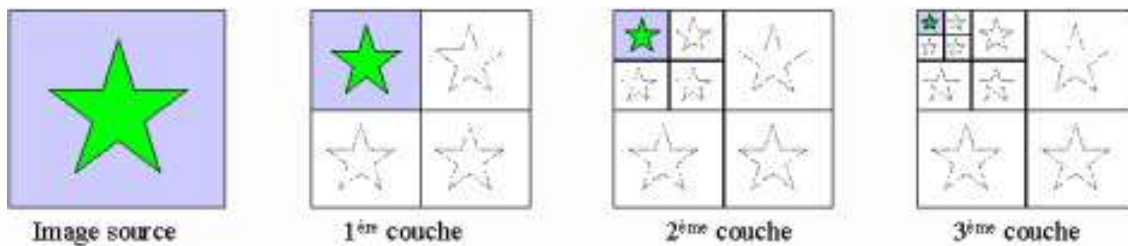
(voir doc intro\_ondelettes @ Moodle)

109

109

## Utilisation des ondelettes en TI

Transformée par ondelettes



- Analyse multi résolution de l'image
- Sur plusieurs couches
  - ✓ A chaque couche : les dimensions géométriques de l'image sont réduites dans un rapport 2 par un ensemble de filtres orthogonaux

110

110

## Utilisation des ondelettes en TI

Transformée par ondelettes

Résultat = 4 petites images

- Image « smooth » : représente la réduction de l'image source
- Images « détails » : contiennent les infos de hautes fréquences spatiales perdues lors de la réduction.
- Le passage d'une couche à une autre se fait en prenant l'image réduite (smooth), il y a alors 4 fois moins de points à traiter.



111

111



## Utilisation des ondelettes en TI

Décomposition d'une image en ondelettes



112

112

## Utilisation des ondelettes en TI

Compression d'images sur bases d'ondelettes : format JPEG 2000



- Figure de gauche : Image  $256 \times 256 = 65536$  valeurs, 256 niveaux de gris.
- Figure du milieu : 4000 plus grands coefficients d'ondelettes
- Figure de droite : image recomposée à partir de ses 4000 plus grands coefficients sur une base d'ondelettes à 4 moments nuls.
- Compression =  $(65536 - 4000) \times 100 / 65536 = 93,9 \%$

113

113

## Utilisation des ondelettes en TI

Compression d'images sur bases d'ondelettes : format JPEG 2000

Restitution par  
Ondelettes



... à taux de compression identique !

114

114

## Hidden Dimension of the Nature

Une **fractale** naturelle:  
la feuille de fougère est  
constituée d'éléments  
ressemblant eux-  
mêmes à des feuilles  
de fougère.

A natural **fractal**: the  
fern leaf is made up of  
elements that  
themselves resemble  
fern leaves.



115

115

## HIDDEN DIMENSION NATURE

Chou Romanesco  
Romanesco cabbage



116

116

## HIDDEN DIMENSION OF NATURE

Cabbage



117

117





Aselliformis  
Pelecyphora



Ludwigia Sedioides



Lobelia



Camelia

118



Chou Romanesco



Aloe Polyphylla



Dalia



Crassula

119





Spiral aloe vera



Cactus

Hoya Aldrichii



120



Nénufar géant  
Giant Amazon  
water Lily

Drosophyllum  
Lusitanicum



Choux rouge  
Red cabbage



Tournesol  
Sunflower

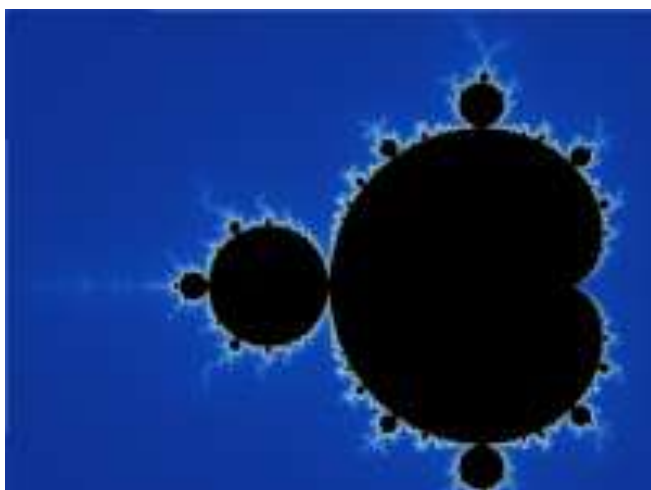


121

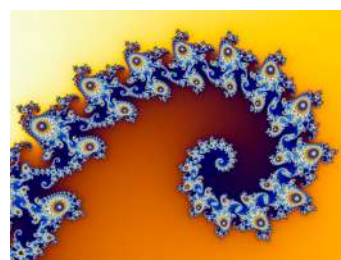
[https://en.wikipedia.org/wiki/File:Mandelbrot\\_sequence\\_new.gif](https://en.wikipedia.org/wiki/File:Mandelbrot_sequence_new.gif)

## FRACTALES

Une figure **fractale** est un objet mathématique, telle une courbe ou une surface, dont la structure est invariante par changement d'échelle<sup>1</sup>.



Exemple de figure fractale  
(détail de l'ensemble de Mandelbrot)



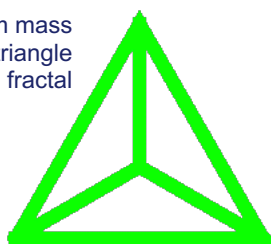
122

SORBONNE  
UNIVERSITÉ

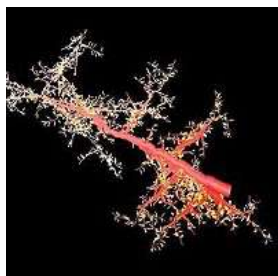
122

## FRACTALS

Uniform mass  
center triangle  
fractal



Self-similar  
branching pattern  
modeled in  
silico using  
L-systems principles



A strange  
attractor that  
exhibits multifractal  
scaling



123

[https://upload.wikimedia.org/wikipedia/commons/5/5e/Karperien\\_Strange\\_Attractor\\_200.gif](https://upload.wikimedia.org/wikipedia/commons/5/5e/Karperien_Strange_Attractor_200.gif)

SORBONNE  
UNIVERSITÉ

123



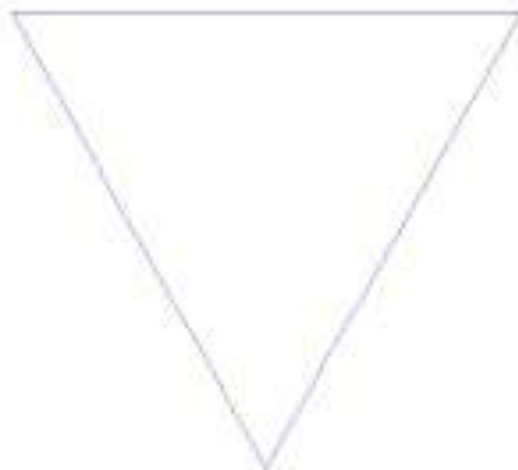
## Flocon de Koch

Le flocon de Koch est l'une des premières courbes fractales à avoir été décrites, bien avant l'invention du terme « fractal(e) » par Benoît Mandelbrot.

Elle a été inventée en 1904 par le mathématicien suédois Helge von Koch.



124



SORBONNE  
UNIVERSITÉ

124

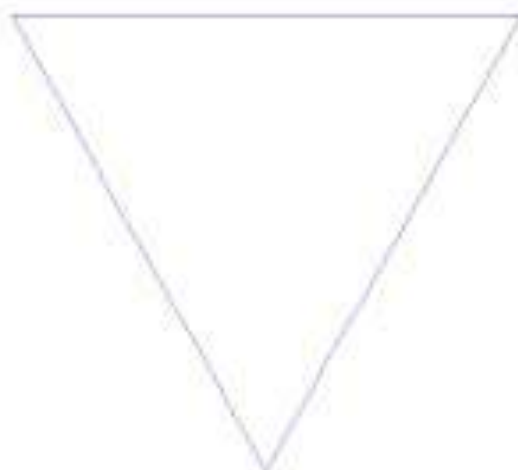
## Koch snowflake

**Niels Fabian Helge von Koch**  
(25 January 1870 – 11 March 1924)

was a [Swedish mathematician](#) who gave his name to the famous [fractal](#) known as the [Koch snowflake](#), one of the earliest fractal curves to be described



125



SORBONNE  
UNIVERSITÉ

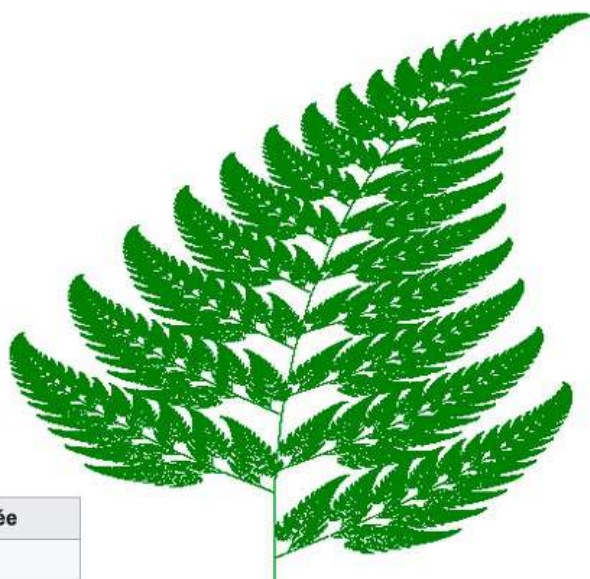
125

## Fougère de Barnsley

La fougère de Barnsley est une fractale nommée d'après le mathématicien Michael Barnsley qui l'a décrite pour la première fois dans son livre *Fractals Everywhere*

$$f(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

w	a	b	c	d	e	f	p	Partie générée
$f_1$	0	0	0	0.16	0	0	0.01	Tige
$f_2$	0.85	0.04	-0.04	0.85	0	1.60	0.85	Petites folioles
$f_3$	0.20	-0.26	0.23	0.22	0	1.60	0.07	Grandes folioles de gauche
$f_4$	-0.15	0.28	0.26	0.24	0	0.44	0.07	Grandes folioles de droite



126

## Barnsley fern

The Barnsley fern is a fractal named after the British mathematician Michael Barnsley who first described it in his book *Fractals Everywhere*.

$$f(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

w	a	b	c	d	e	f	p	Portion generated
$f_1$	0	0	0	0.16	0	0	0.01	Stem
$f_2$	0.85	0.04	-0.04	0.85	0	1.60	0.85	Successively smaller leaflets
$f_3$	0.20	-0.26	0.23	0.22	0	1.60	0.07	Largest left-hand leaflet
$f_4$	-0.15	0.28	0.26	0.24	0	0.44	0.07	Largest right-hand leaflet



127

## La compression fractale (avec perte)

- 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 repose sur la détection de la récurrence des motifs, et **tend à éliminer la redondance d'informations dans l'image.**
- C'est une méthode destructive puisque l'ensemble des données de départ ne se retrouve pas dans l'image finale. Il existe plusieurs méthodes (subdivision de triangles, Delaunay etc.) mais la **compression par la méthode Jacquin** est la plus connue.

128



128

## La compression fractale (avec perte)

- 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. Par contre la décompression est rapide.
- La méthode est la plus efficace pour les photos des scènes de la nature (arbres, montagnes, nuages).

129



129

## Methode Jacquin (compression fractale)

- La compression fractale consiste à réaliser deux segmentations (appelées aussi pavages, ou partitionnements) sur une image : une segmentation de figures Sources et une segmentation de figures Destinations.
- Il s'agit alors de trouver pour chaque figure Source, quel est le meilleur couple (figure source, figure destination) minimisant une erreur. Cette erreur est généralement calculée en soustrayant les deux figures. Pour réaliser l'opération de soustraction, il est nécessaire d'opérer une transformation de la figure source aux dimensions (et à la géométrie) de la figure destination. De plus, des règles comme la rotation et les retournements sont possibles.
- Une fois que tous les couples ont été trouvés, le fichier de sortie contient les différents couples, ainsi que les différentes transformations effectuées (rotation, réduction de la moyenne etc.).
- Lors de la décompression, l'image est recrée à partir de ces transformations. La convergence est alors garantie par le fait que d'une part il y a une minimisation d'erreur (différence) et une modification des pixels, et d'autre part, que les figures sources sont plus grandes que les figures destinations. La compression fractale utilise la même propriété pour reconstruire l'image.

130



130

## Partitionnement

- Le partitionnement est l'opération qui consiste à segmenter une image en régions. Dans la compression par la méthode Jacquin, nous avons besoin de 2 partitionnements : Source et Destination. La méthode Jacquin utilise par exemple des figures carrées, mais d'autres formes sont possibles (nids d'abeilles, triangles etc).
- Un point essentiel dans les partitionnements Source et Destination est que le pavage destination doit être plus petit que le pavage source. En effet, dans le cas contraire, nous serions amenés à faire un agrandissement (et non une réduction) lors de la transposition des figures sources vers les figures destinations. Une fractale possède un motif se répétant à l'infini, en se rétrécissant. Aussi, nous perdons cette propriété si le partitionnement destination est plus grand que le partitionnement source, l'image ne pourra alors pas converger.
- Le partitionnement par la méthode Jacquin est un partitionnement statique. L'utilisation d'un partitionnement adaptatif (qui dépend de l'image à traiter) améliore considérablement le facteur de compression

131



131

## Décompression

- La décompression consiste en la lecture du fichier contenant les correspondances figure source-figure destination. Il suffit ensuite d'appliquer les transformations plusieurs fois. Ce procédé de reconstruction itéré, aussi connu sous le nom de système de fonctions itérées, garantit une convergence, relative, vers l'image de départ.
- La qualité du résultat dépend fortement de la taille des figures de segmentation, plus les figures seront nombreuses, et plus l'image résultante sera de qualité.

132



132

## Assises mathématiques

- Une image en noir et blanc peut être représentée par un vecteur dans un espace vectoriel de dimension  $n$  : la  $i$ -ième composante du vecteur représente le niveau de gris du  $i$ -ième pixel. Le niveau de gris d'un pixel peut aller de 0 à 255. C'est sur ce modèle mathématique d'une image que pourront s'appliquer la compression et la décompression. On peut considérer une image en couleur comme étant la superposition de 3 calques primaires rouge vert bleu.
- Pour la compression, la garantie de pouvoir trouver les couples (figure source, figure destination) ainsi que les transformations adaptées repose sur le **Théorème du collage**.
  - *Indique que si la différence d'erreur entre l'image cible et la transformation de cette image est inférieure à une certaine valeur, les transformées sont une représentation équivalente de l'image.*
- Pour la décompression, la garantie de convergence vers l'image compressée est assurée par le **Théorème du point fixe**. C'est ce théorème qui impose que le pavage destination doit être plus petit que le pavage source

133



133

## Contexte mathématique

Commencé avec Michael Barnsley (1988-), et affiné par A. Jacquin (1991-)

Essayer de trouver un ensemble de transformations qui font correspondre une image sur elle-même.

La clé est le **théorème du collage**

- Il stipule que si la différence d'erreur entre l'image cible et la transformation de cette image est inférieure à une certaine valeur, les transformations sont une représentation équivalente de l'image.

134



134

## Comment cela fonctionne-t-il ? - Encodage

- Prendre une image de départ et la diviser en petits blocs carrés qui ne se chevauchent pas, généralement appelés "blocs parents".
- Divisez chaque bloc parent en 4 autres blocs, ou "blocs enfants".
- Comparez chaque bloc enfant à un sous-ensemble de tous les blocs chevauchants possibles de la taille du bloc parent.
  - **Il faut réduire la taille du bloc parent pour que la comparaison fonctionne.**
- Déterminer quel bloc plus grand présente la plus petite différence, selon une certaine mesure, entre lui et le bloc enfant.
- Calculer une transformation en niveaux de gris pour faire correspondre précisément les niveaux d'intensité entre le grand bloc et le bloc enfant. En général, on utilise une transformation affine ( $w \cdot x = a \cdot x + b$ ) pour faire correspondre les niveaux de gris.

136



136

## Comment fonctionne l'encodage ?

Bloc enfant dans le coin supérieur gauche, très similaire au bloc parent supérieur droit.

Calculer la transformation affine.

Enregistrez l'emplacement du bloc parent (ou bloc de transformation), les composants de la transformation affine et le bloc enfant correspondant dans un fichier.

Répétez l'opération pour chaque bloc enfant.

De nombreuses comparaisons peuvent être effectuées.

- Image originale 256 x 256
- Blocs parents de taille 16 x 16
- $241 \times 241 = 58\,081$  comparaisons de blocs

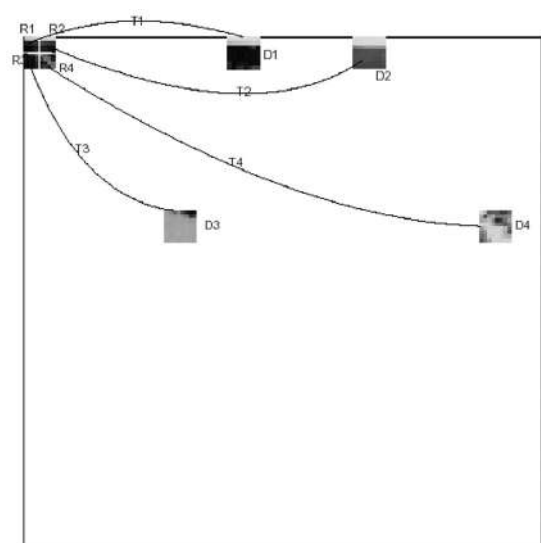


138

138

## Encodage / Compression

Cartographie des blocs de portée (angl. range) (R) et des blocs de domaine (D) et leurs transformations (T)



140

140



## Comment ça marche ? - Décodage

- Lire les informations de position, de transformation et de taille du bloc enfant et du bloc de transformation.
- Utiliser une image de départ vierge de la même taille que l'image originale.
- Pour chaque bloc enfant, appliquer les transformations stockées au bloc de transformation spécifié.
- Remplacer les valeurs des pixels du bloc enfant par les valeurs des pixels du bloc de transformation.
- Répétez l'opération jusqu'à ce que vous obteniez une qualité d'image acceptable.

142



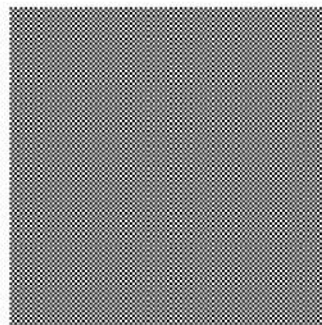
142

## Exemples

Original Image



Starting Image for Decoding



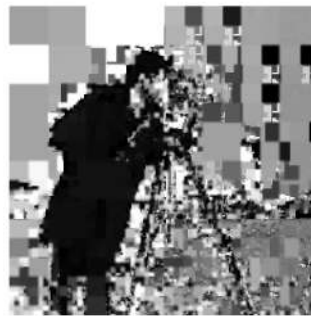
144



144

## How does it Work? – Decoding

First Iteration



Second Iteration



145

SORBONNE  
UNIVERSITÉ

145

## How does it work? - Decoding

Fifth Iteration



Tenth Iteration



146

SORBONNE  
UNIVERSITÉ

146

## Décodage / décompression



Initial image

Les transformations et les mises en correspondance sont appliquées sur l'image initiale de manière itérative jusqu'à ce que l'image soit restaurée.



1st iteration



2nd



3rd



4th



5th



6th

147

147

## Compression JPEG vs. Fractale

- Le JPEG est meilleur à des taux de compression faibles, et la compression fractale est meilleure à des taux élevés.
- Point d'intersection à 40:1
- Les images compressées de façon fractale ont un bruit beaucoup plus naturel que le JPEG.
- Même temps de décompression que le JPEG, parfois plus rapide.
- La compression fractale induit un temps de compression beaucoup plus lent que le JPEG.
- On peut zoomer sur l'image fractale et l'image aura toujours un aspect naturel -> taux de compression effectif plus élevé

149

149

## Compression fractale - Améliorations possibles

- La plus grande faiblesse est le temps d'encodage
  - **Accélération possible**
    - Ordonner les blocs de transformation en domaines basés sur l'intensité moyenne et la variance.
    - Recherche uniquement dans les blocs ayant des structures similaires
    - Ne pas rechercher tous les blocs possibles
    - Réduire le nombre de blocs enfants
- Amélioration de la qualité et de la compression grâce à
  - **Quadtrees ou HV Trees**
  - **Rotations des blocs de transformation pendant la comparaison**
  - **Transformations en niveaux de gris améliorées**

151



151

## Décompression indépendante de la résolution (Zoom)

L'image fractale zoomée semble contenir plus de détails que le JPEG zoomé.

Mais les détails ont été générés, si bien qu'en zoomant sur le visage d'une personne, on ne verra pas les pores.



Full sized original

Fractal image

Zoomed JPEG

153



153



155

155

## La compression avec perte - evaluation

### Evaluation de la qualité de la compression

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

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

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

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

156

156