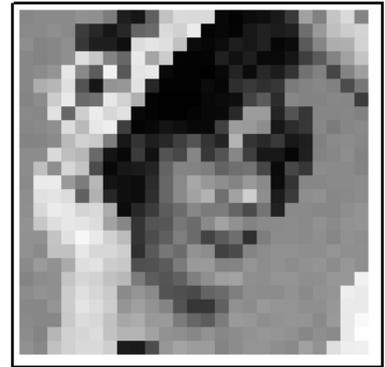
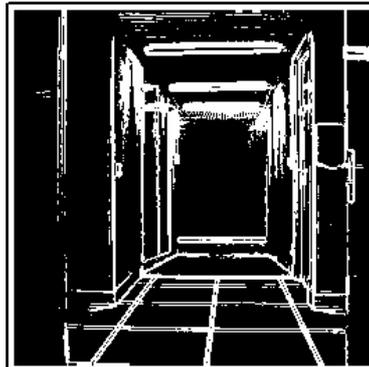
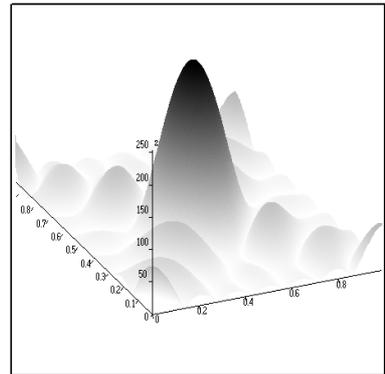
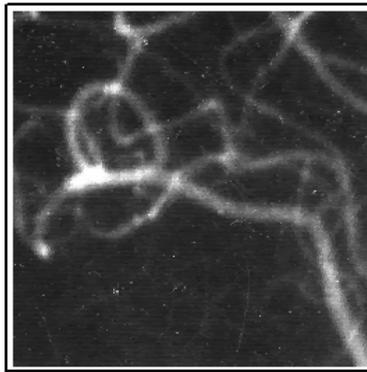
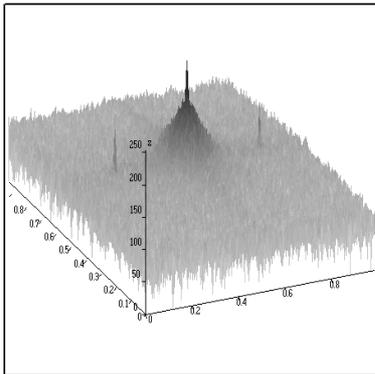

Digital Image Processing



Par Antoine Fouillé

D'après le cours de **Jocelyn Chanussot**

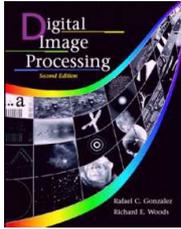
Professor, signal & Image processing

jocelyn.chanussot@gipsa-lab.inpg.fr

GIPSA-Lab (<http://www.gipsa-lab.inpg.fr/>)

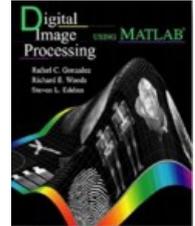
Grenoble Institute of Technology (<http://www.grenoble-inp.fr/>)

1. Bibliography



Digital Image Processing, 2nd Edition
by Gonzalez and Woods
Prentice Hall, 2002

Digital Image Processing Using MATLAB
by Gonzalez, Woods, and Eddins,
Prentice Hall, 2004



Le traitement des images
(Traité IC2, série Traitement du signal et de l'image)
By Maître
Hermès, 2003

Traitement et analyse des images numériques
By Bres, Jolion and Lebourgeois
Hermès, 2003



Analyse d'images : Filtrage et segmentation
By Cocquerez and Philipp
Masson, 1995

Introduction au traitement d'images:
Simulation sous Matlab
By Burel
Hermès, 2001



2. Digital images

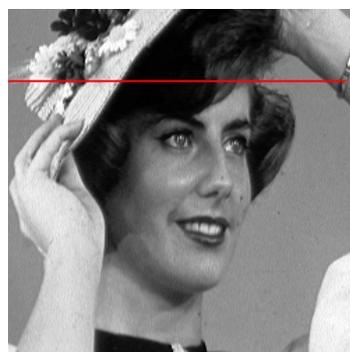
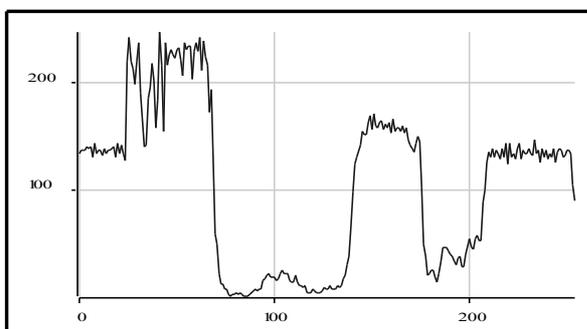
Une image numérique est un ensemble fini de pixels codés par des bits issus d'un échantillonnage spatial. Généralement chaque pixel est codé sur 8 bits (0 à 255) pour les images N&B. Pour les images couleurs, chaque pixel possède trois composantes (RVB) sur 8 bits, ce vecteur caractérise l'intensité de chaque couleur dans l'ensemble RVB (rouge, vert, bleu). L'image est alors une matrice de nombre.

2.1. Niveau de gris et table de correspondance des couleurs (Grey levels and look up table, LUT) :



A chaque valeur d'un pixel, il sera associé une couleur selon le tableau de correspondance des couleurs pour former l'image numérique.

2.2. Coupe d'image (Grey level profile (cut)):



Ceci est une coupe de l'image sur une ligne horizontale se trouvant (~1/6 en partant du haut). Elle indique le niveau de gris en fonction de l'ordonnée de l'image.

Exemple d'utilisation : Il est possible d'appliquer la TF à cette coupe pour chercher un motif se répétant sur une tranche de l'image. Ainsi il apparaît une fréquence dans la TF : la fréquence du motif.

2.3. La quantification (Quantization) :

La quantification est une opération de \mathbf{R} dans \mathbf{n} un ensemble fini de \mathbf{N} . La quantification transforme un point x de \mathbf{R} en son plus proche voisin dans \mathbf{n} .

Visage :



8 bits

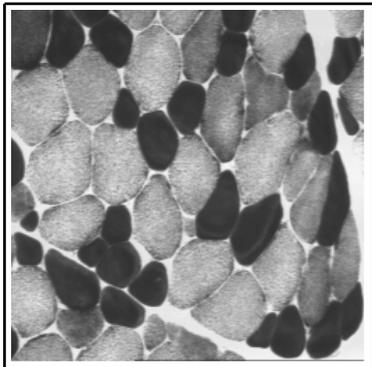


4 bits

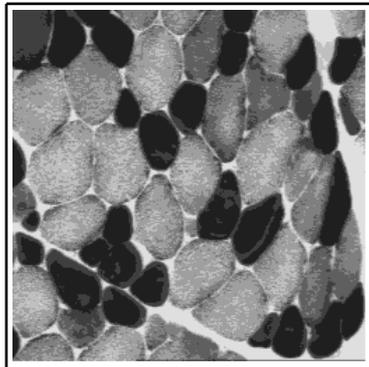


2 bits

Cellules musculaires :



8 bits



3 bits

Comme on l'observe dans l'échantillonnage du visage, un mauvais échantillonnage peut entraîner la formation de palier de couleurs. Les contours restent assez nets et ne sont pas déformés, cependant le dégradé de couleurs est grossier.

Il ne faut pas perdre de vue que la qualité de l'image dépend de ce que l'on veut en faire. En effet, un échantillonnage faible, avec une forte dégradation de l'image peut suffire parfois, comme l'illustre l'exemple des cellules musculaires. Si il faut dénombrer les cellules noires, un échantillonnage sur 3 bits est suffisant.

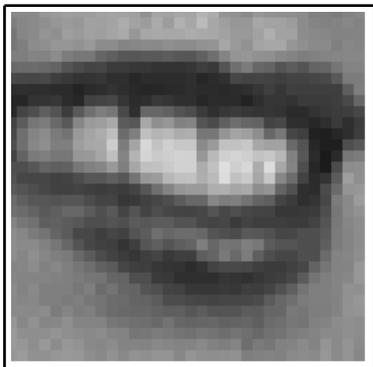
La qualité de l'image est donc un critère tout à fait subjectif car il dépend de l'utilisation de l'image.

2.4. La pixellisation (Sampling) :

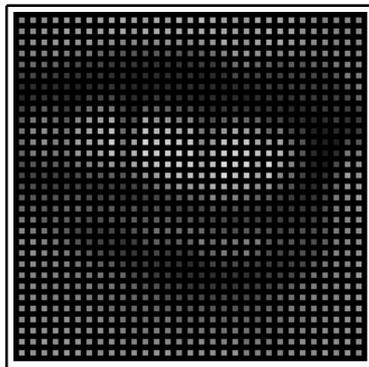


Contrairement à la quantification qui peut produire des palier de couleur ; un mauvais échantillonnage peut engendrer une pixellisation de l'image avec des contours in formes.

2.5. Zoom et interpolation (Zoom & interpolation) :



Duplication



Insertion de Zéros



Filtrage Passe Bas

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow \begin{bmatrix} a & a & b & b \\ a & a & b & b \\ c & c & d & d \\ c & c & d & d \end{bmatrix}$$

$$\begin{bmatrix} a & 0 & b & 0 \\ 0 & 0 & 0 & 0 \\ c & 0 & d & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} a & \frac{1}{2}a + \frac{1}{2}b & b & \frac{1}{2}b \\ \frac{1}{2}a + \frac{1}{2}c & \frac{1}{2}a + \frac{1}{2}b + \frac{1}{2}c + \frac{1}{2}d & \frac{1}{2}c + \frac{1}{2}b & 0 \\ c & \frac{1}{4}a + \frac{1}{4}b + \frac{1}{4}c + \frac{1}{4}d & d & \frac{1}{2}d \\ \frac{1}{2}c & 0 & \frac{1}{2}d & 0 \end{bmatrix}$$

Le but d'un zoom est d'agrandir l'image. Cependant il est possible que l'image n'est pas une résolution suffisante, alors une pixellisation de l'image devient visible, car un zoom basique (sur Windows MS par exemple) duplique les informations comme pour l'image 1.

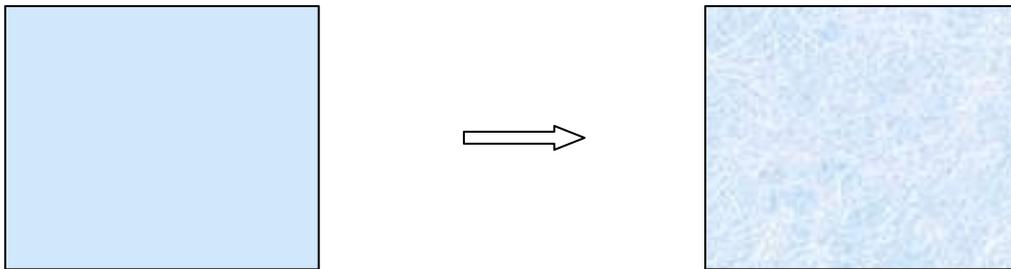
Ici, l'interpolation consiste à insérer des zéros (image 2), puis à les remplacer par une moyenne des valeurs des bits voisins. Cette opération peut s'effectuer grâce à la convolution de la matrice avec les zéros par la matrice $1,4 \begin{pmatrix} 0,5 & 1 & 0,5 & 0 \end{pmatrix}$, puis par la Matrice $4,1 \begin{pmatrix} 0,5 & 1 & 0,5 & 0 \end{pmatrix}$. Cela a pour effet de lisser l'image.

Le filtrage est équivalent à un filtre passe bas analogique ($H(\omega) = H^0 / (1 + j \omega / \omega^0)$)

2.6. Repliement de spectre et sous échantillonnage :

Lorsque l'échantillonnage n'est pas suffisant ($F_e < 2 F_{max}$, d'après Shannon), il y a repliement de spectre. Alors de nouvelles fréquences peuvent apparaître. C'est un effet responsable de la détérioration d'un fond d'image, par exemple. Si le fond d'une image n'est pas parfait et que l'on sous échantillonne, alors des imperfections peuvent apparaître à une nouvelle fréquence.

C'est typiquement ce qui arrivait lorsque l'on enlevait des lignes et des colonnes pour réduire la résolution(nombre de pixel) de l'image.



Pour éviter cela, il faut filtrer passe bas à la nouvelle fréquence d'échantillonnage, puis ensuite, seulement ré échantillonner.

2.7. Mesure de détérioration entre deux images (Measuring the difference between two images f and \tilde{f} ...):

La mesure de la dégradation de l'image se fait par comparaison (distance de Hamming) entre l'image d'origine et la nouvelle image.

Mean Absolute Error :

$$MAE = \frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N |f(m,n) - \tilde{f}(m,n)|$$

Erreur absolue moyenne

Mean Square Error :

$$MSE = \frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N (f(m,n) - \tilde{f}(m,n))^2$$

Erreur quadratique moyenne

Signal to Noise Ratio (dB) :

$$SNR = 10 \log \left(\frac{\frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N f^2(m,n)}{\frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N (f(m,n) - \tilde{f}(m,n))^2} \right)$$

Rapport signal à bruit

Peak Signal to Noise Ratio (dB) :

$$PSNR = 10 \log \left(\frac{255^2}{\frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N (f(m,n) - \tilde{f}(m,n))^2} \right)$$

Rapport crête de signal à bruit

Les normes grand public (GP) considèrent qu'une image est assez bonne si la SNR > 30 dB. En deçà, l'image est trop dégradée.

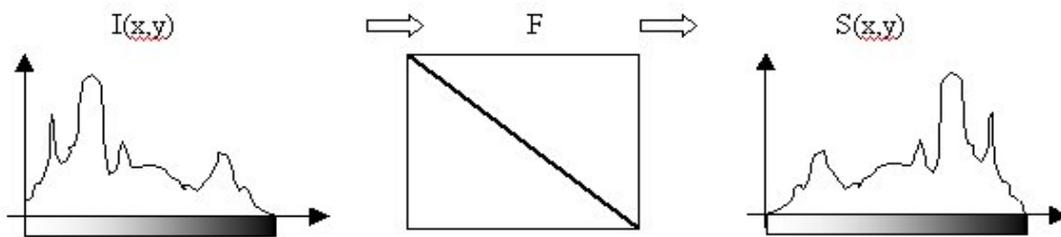
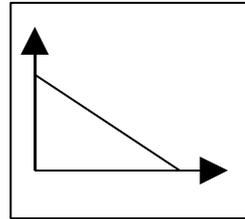
Rem 1 : Si le Bruit augmente, la SNR diminue.

Rem 2 : Si on perd un bit, la SNR diminue de 6 dB.

3. Histogram modification

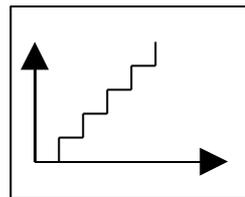
3.1. Transformation du niveau de gris (Grey level transform) :

La transformation est l'image en négative soit $v(x,y) = 255 - v(x,y)$. La fonction correspondante est : Niveaux de gris en sortie en fonction du niveaux de gris en entrée.



Une autre transformation peut être réalisée : la sous quantification.

Ici le nombre de palier indique le nombre de bits codant un pixel de l'image après transformation.



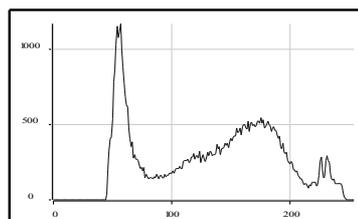
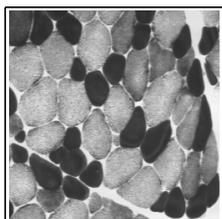
3.2. Transformation de seuil (Threshold) :

Le seuil est une sous quantification particulière.

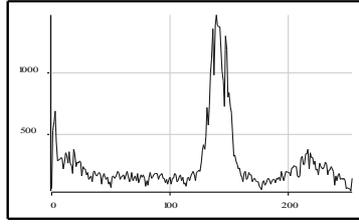


3.3. L'histogramme (Histogram) :

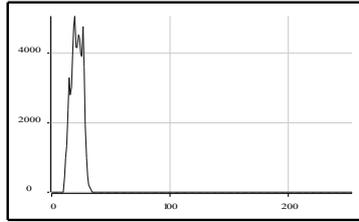
L'histogramme donne le nombre de pixels en fonction du niveau de gris.



Ici on voit clairement qu'il y a trois zones de répartition : une sombre, une intermédiaire et une claire.



Ici on observe une répartition presque uniforme en dehors du pic central. Le pic central indique que la plupart des pixels sont dans les niveaux de gris intermédiaires (fond et peau).



On pourrait penser que l'image est noire. Cependant l'histogramme indique qu'il y a plusieurs nuances de gris.

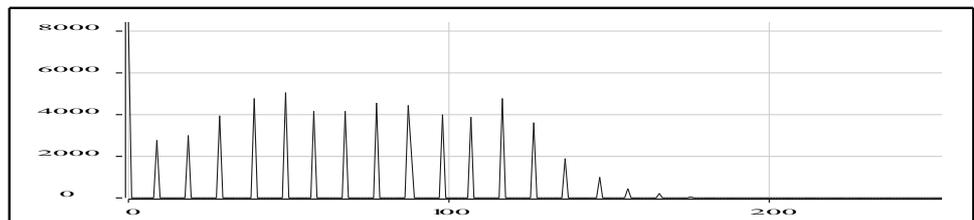
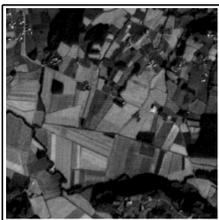
Rem 1 : L'oeil est plus sensible au faible contraste dans le clair que dans le sombre. Ne pas hésiter à passer au négatif.

Rem 2 : Pour la troisième image, il serait intéressant de dilater l'échelle des niveaux de gris : c'est le recadrage.

Rem 3 : Une donnée informatique contient un maximum d'informations lorsque que l'on a un histogramme plat, c'est-à-dire lorsque la distribution est uniforme, si on considère l'histogramme comme une densité de probabilités (Shannon).

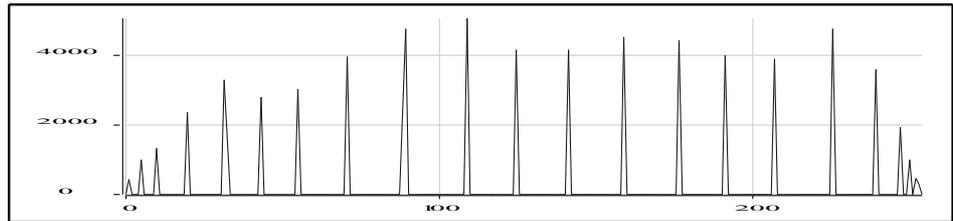
3.4. Recadrage linéaire de la dynamique (Linear rescaling of the range) :

Le recadrage consiste à dilater l'échelle des gris.



On redistribue les pics. La linéarité réside dans le fait que l'écart entre les pics est constant.

3.5. Egalisation (Histogram equalization) :



L'écart entre les pics est proportionnel à la hauteur des pics (~nombre de pixel à un niveau de gris). Ce n'est, donc, plus linéaire et le résultat semble mieux contrasté.

3.6. Convolution (convolution) :

1 D, continue (continuous space) : $f * h (t) = \int_{-\infty}^{\infty} (f(x)) \cdot (h(t - x)) dx$

2 D, continue : $f * h (u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (f(x, y)) \cdot (h(u - x, v - y)) dx dy$

2 D, discret (discrete space) : $f * h (u,v) = \sum_{y=-\infty}^{\infty} \left(\sum_{x=-\infty}^{\infty} ((f(x, y)) \cdot (h(u - x, v - y))) \right)$

Propriétés :

Commutativité (Convolution is *commutative*) :

$$c = a \otimes b = b \otimes a$$

Associativité (Convolution is *associative*) :

$$c = a \otimes (b \otimes c) = (a \otimes b) \otimes c = a \otimes b \otimes c$$

Distributivité (Convolution is *distributive*) :

$$c = a \otimes (b + d) = (a \otimes b) + (a \otimes d)$$

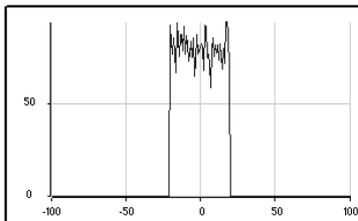
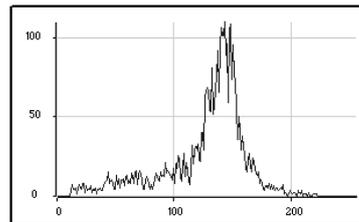
4. Noise reduction

La méthode expérimentale de réduction de bruit consiste, ici, à brouter une image et trouver le meilleur filtre pour ce type de bruit. Ensuite on expérimente différents types de filtres et on observe celui donnant les meilleurs résultats. Enfin on associe à un certain type de bruit, le filtre convenant le mieux.

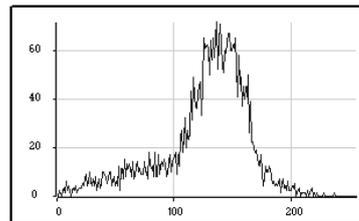
4.1. Models of noisy images



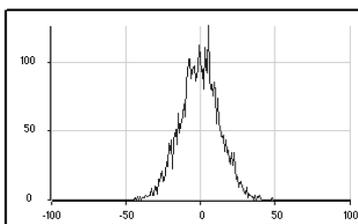
Original image



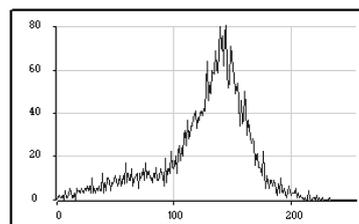
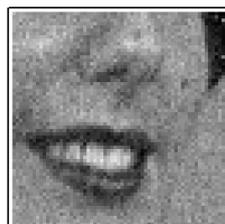
Uniform distribution



Le bruit à distribution gaussienne correspond, souvent, au bruit électronique. De manière plus générale, il est l'effet macroscopique d'une somme d'effets microscopiques.



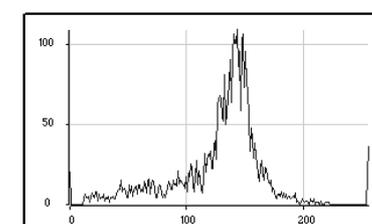
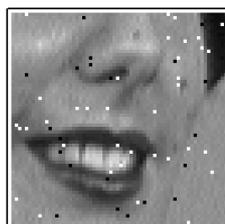
Gaussian distribution



Ce bruit correspond typiquement au acquisition CCD, ou certains capteurs seraient Défectueux (noir) ou saturé (blanc).

Ici, le bruit n'est plus additif comme dans les bruitages précédents.

Salt & Pepper noise

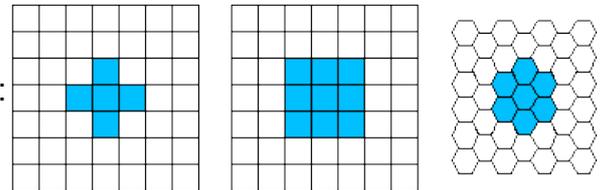


Rem 1 : Attention, l'histogramme de l'image bruitée n'est pas la somme des histogrammes de l'image et de bruit, mais leur convolution.

Rem2 : Il existe d'autres types de bruit couramment rencontrés : le bruit photonien, le bruit d'amplification, le bruit de quantification , ...

4.2. Fenêtre de filtrage (sliding filtering window) :

Il y a différents types de fenêtres de filtrages : Ici on prendra pour les exemples, celles du milieu.



4.3. Lissage avec une fenêtre de filtrage (Smoothing with a sliding filtering window) :

- Filtre linéaire (Linear filtering) : convolution

$$f * h(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$$

Le filtre moyeneur (Mean filter), est un filtre qui a pour action de remplacer chaque pixel par une moyenne arithmétique de ses voisins et lui-même. Le masque (fenêtre de filtrage) est alors de la forme :
Il équivaut à un intégrateur analogique.

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

Ce filtre a de très bon résultats sur une bruit gaussien ; cependant il introduit un flou dans l'image (les contours nets sont transformés en dégradé).

- Filtre non linéaire : filtre d'ordre (Non linear filtering : order filters)

|
|
|

10<45<51<51<61<88<124<134<152

Le filtre médian (Median filter) est un filtre non linéaire équivalent à un passe bande analogique. Il prend la valeur qui laisse autant de valeurs à gauche qu'à droite.

Sur l'exemple c'est 61 : 10<45<51<51<**61**<88<124<134<152

Ce filtre convient particulièrement au bruit de type poivre et sel. Il a tendance a arrondir les coins.

Le filtre milieu (Middle filter) est un filtre non linéaire équivalent à un dérivateur analogique. Il prend le maximum et le minimum de la fenêtre de filtrage et en fit la moyenne.

Sur l'exemple c'est 81 : **10<45<51<51<61<88<124<134<152**
 Ce filtre convient particulièrement au bruit uniforme. Il a tendance à créer un cadre de la largeur de la fenêtre de filtrage au niveau des contours.

	gaussien	uniforme	poivre /sel	flou	contours	linaire
médian	*		***	non	arrondi coins	non
milieu		*		oui	cadre	non
moyen	***	***		oui	dégradés	oui

4.4. Ré haussement de contraste (Enhancement) :

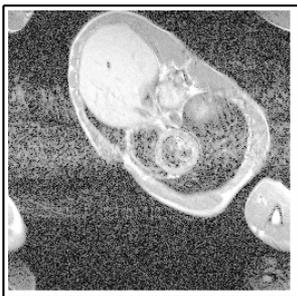
Le ré haussement de contraste obéit à l'algorithme suivant :

*Si moyenne > milieu
 Alors max (moyenne, milieu)
 Sinon min (moyenne, milieu)*

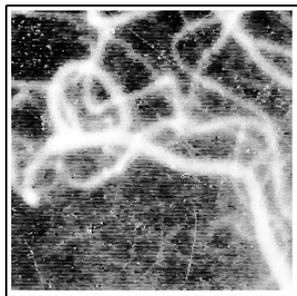


Le ré haussement de contraste accentue les contours mais aussi le bruit.

4.5. Real images



MRI



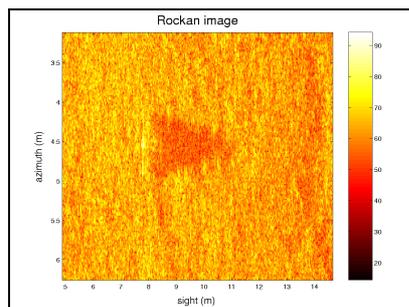
angiography (X-Ray)



echography (Ultra-Sound)



CCD



sonar

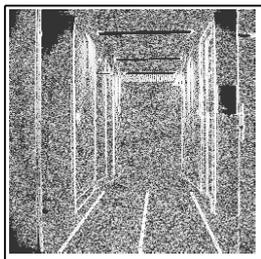
Dans le cas d'images réelles, ou il n'y a pas d'image de référence non bruitée, il faut tenter de deviner de quel type de bruit il s'agit. On sélectionne une partie que l'on suppose être uniforme sans le bruit. Puis on analyse cette partie de l'image avec un histogramme. A partir de l'histogramme, un test d'hypothèse est réalisé (χ^2). Puis le meilleur filtre, pour la loi utilisée pour le test du χ^2 , est appliqué.

5. Edge detection

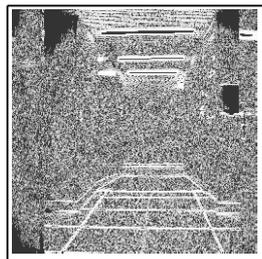
La détection de contours (edge detection) sert isoler un objet par sa forme. Un contour est un changement rapide d'intensité.

Gradient

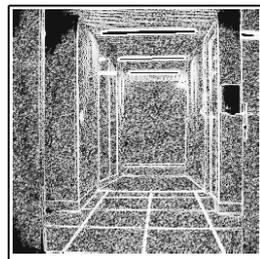
Le gradient est équivalent a un dérivateur analogique. Il très sensible au bruit.



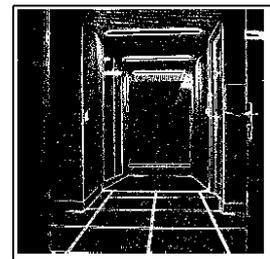
|grad1|
Gradient vertical



|grad2|
gradient horizontal

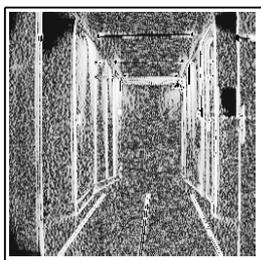


max(|grad1|, |grad2|)
norme1 des deux grad

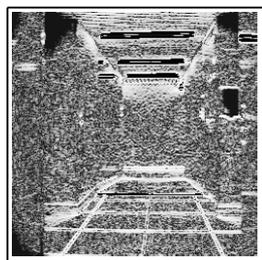


threshold
seuil

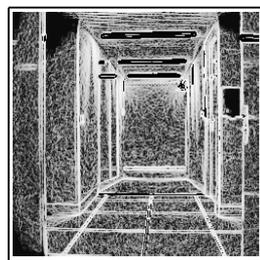
Comme le gradient est trop sensible au bruit, une détection de contours est communément utilisée : la détection de Sobel (il existe une variante de Prewitt).



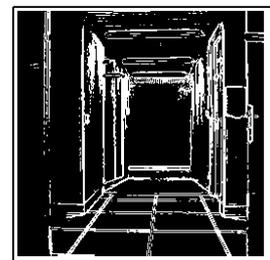
|sobel1|



|sobel2|



max



threshold

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

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

Ici l'astuce réside dans le fait que l'on filtre selon une direction et que l'on différencie selon une autre direction.

Il reste cependant un problème, les objets peuvent apparaître avec des contours qui ne sont pas fermés. Il faut donc créer un algorithme qui ferme les contours.

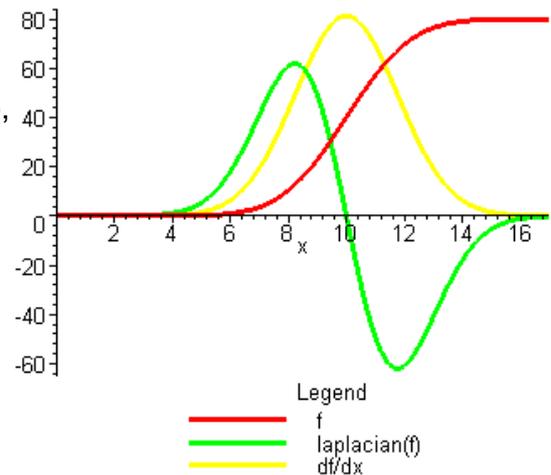
Le laplacien (Laplacian) :

L'opérateur laplacien est une dérivation au second ordre :

$$\Delta f(x,y) = \left(\frac{\partial^2}{\partial x^2} f(x,y) \right) + \left(\frac{\partial^2}{\partial y^2} f(x,y) \right)$$

Il permet de détecter le maximum ou minimum local de la dérivé première de la fonction (jaune), lorsque le laplacien (vert) passe par zéro :

rouge : f
jaune : df/dx
vert : laplacien(f)



Ce passage est très intéressant car c'est un point précis.

Le masque qui simule le laplacien est isotrope.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

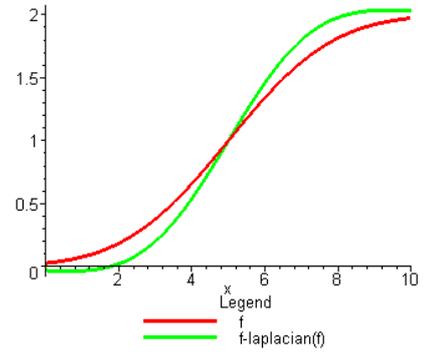
Il n'y a donc qu'une seule convolution à faire.

Le laplacien a deux défaut, il est très sensible au bruit et la détection du passage par zéro n'est pas aisée.

Réhaussement (Enhancement) :

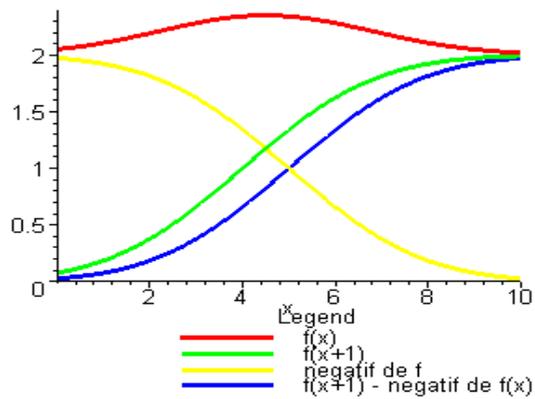
Un second réhaussement peut être utilisé : $f(x,y) - \Delta f(x,y)$. Il a pour effet d'accentuer les contours.

De même que le premier réhaussement que l'on a déjà vu, il est très sensible au bruit.

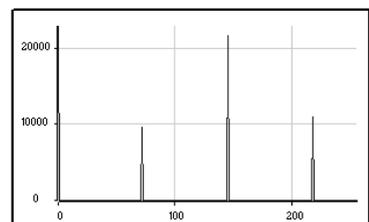


Mise en relief (« emboss ») :

L'effet de mise en relief (rouge) est réalisée en sommant le négatif de l'image originale (jaune) avec l'image originale décalée (vert).



5.5. Orientation analysis



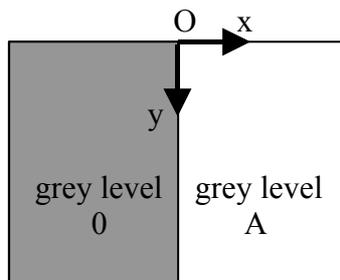
5.6. Linear filtering and noise reduction

We consider an image $f(x,y)$ composed of two homogenous regions separated by a step wise transition (see figure below).

This image is degraded by an additive white noise $b(x,y)$, independent from f , centred and stationary. As a consequence, we observe the following image:

$$g(x,y) = f(x,y) + b(x,y)$$

We want to evaluate the effect of the mean filter both in terms of noise reduction and in terms of edge degradation (the transition is enlarged).



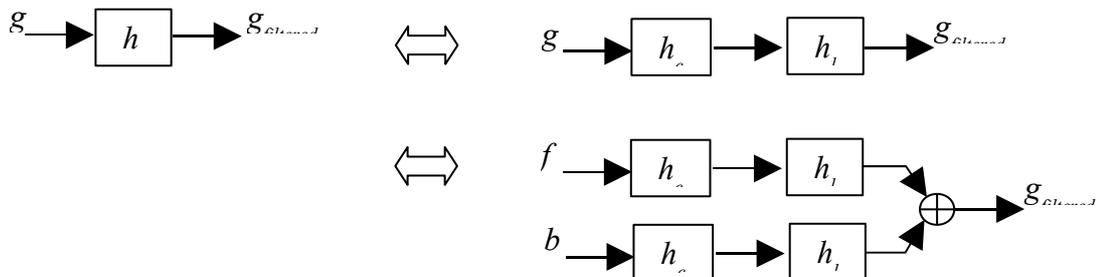
Give a mathematical model of $f(x,y)$ in function of the unit step $u(x)$ (assuming the image has infinite dimensions).

The impulse response of the filter is given by :

$$h(x,y) = \frac{1}{T} \Pi_T(x) \cdot \frac{1}{T} \Pi_T(y)$$

This filter being separable, give the impulse responses h_c and h_l respectively corresponding to the processing of the columns and of the lines.

Successively using the separability and the linearity of the mean filter, we get the following equivalent structures :

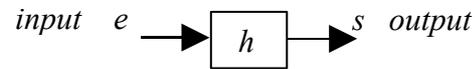


Compute the deterministic part of the obtained image (i.e. the part corresponding to the filtering of $f(x,y)$). Characterize the fuzziness introduced on the edge by giving the length L (centred on the origin) required to have 80% of the transition.

Compute the variance of the random component (i.e. $b(x,y)$) at the output of each operation (after filtering of the columns, are the pixels of a given line still independent?). What is the final gain of the mean filter in terms of noise reduction?

Recalls on linear filtering...

General framework :



We have : $\gamma_s(\nu) = |H(\nu)|^2 \cdot \gamma_e(\nu)$

and : $\Gamma_{ss}(t) = TF^{-1}\{\gamma_s(\nu)\} = \int_{-\infty}^{+\infty} \gamma_s(\nu) e^{2\pi j\nu t} d\nu$

then : $\Gamma_{ss}(0) = \int_{-\infty}^{+\infty} \gamma_s(\nu) d\nu$

but, per definition : $\Gamma_{ss}(t) = \int_{-\infty}^{+\infty} s(\tau) \cdot s(\tau - t) d\tau$

then : $\Gamma_{ss}(0) = \int_{-\infty}^{+\infty} s^2(\tau) d\tau = \text{var}(\text{sortie})$

As a consequence, the output variance can be expressed in function of the modulus of the filter gain and the input power spectral density (which is a constant in case of a white signal...).

Parseval equality: $\int_{-\infty}^{+\infty} |H(\nu)|^2 d\nu = \int_{-\infty}^{+\infty} h(t)^2 dt$

6. 2D Fourier Transform :synthetic image processing

6.1. Définition

La définition de la Transformée de Fourier (TF) pour une fonction f continue dans \mathbf{R} :

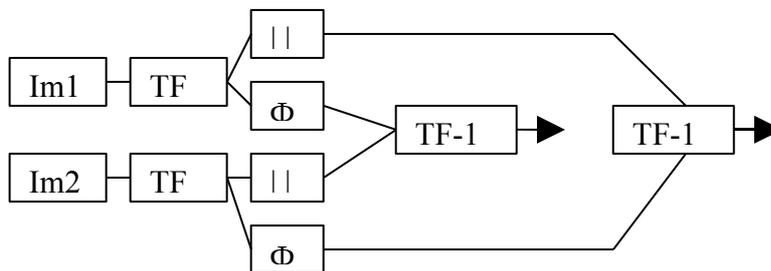
$$\text{TF} : \quad F(u,v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) \cdot \exp[-2i\pi(ux+vy)] \, dx \, dy$$

$$\text{TF-1} : \quad f(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u,v) \cdot \exp[+2i\pi(ux+vy)] \, du \, dv$$

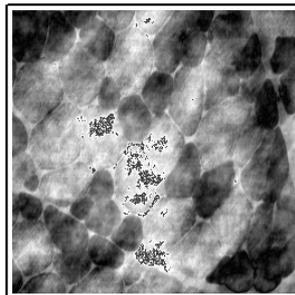
Les variables u et v sont des fréquences spatiales. De manière générale le module de la TF sera utilisé plus souvent que sa phase. En effet il est, encore aujourd'hui assez difficile d'interpréter la phase de la TF.

6.2. Module ou phase (magnitude or phase)?

Pour vérifier les informations portées par la phase et le module des TF des images, on « échange » les phases et les modules entre deux images :



$\Phi(\text{woman}) + |\text{muscle}|$



$\Phi(\text{muscle}) + |\text{woman}|$

Il est clair que la phase porte plus d'informations que le module car dans chacune des images, on reconnaît l'information protégée par la phase.

Pour sentir de manière plus physique les informations contenues dans la phase et le module de la TF, examinons le cas du sinus qui pour |TF| un Dirac. Le module nous renseigne sur la fréquence du sinus, alors que la phase nous donne sa localisation (déphasage).

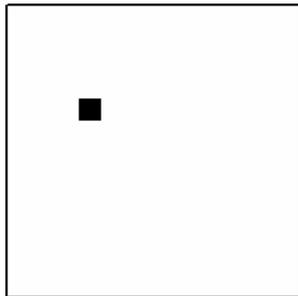
6.3. Carré

-produit de portes

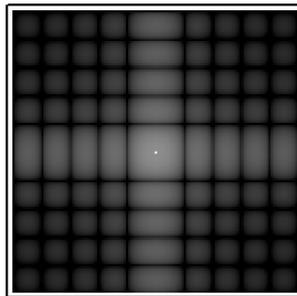
Ici, les images de gauche sont des les images originales. Elles sont séparables, c'est-à-dire, que la fonction de l'image peut se décomposée en un produit de fonctions selon l'axe x et y : $F(x,y) = f1(x) * f2(y)$.

Ainsi, l'image originale est un produit de portes. Or la TF d'une porte est un sinc.

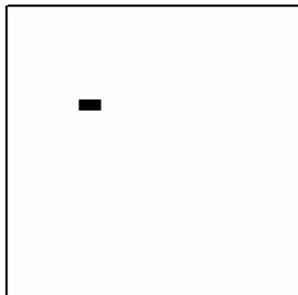
L'image de droite est obtenue par TF, recentrage, échelonnage logarithmique. Il y a toujours un pic central, sur la TF, qui représente la valeur moyenne.



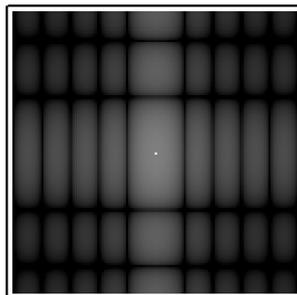
Original



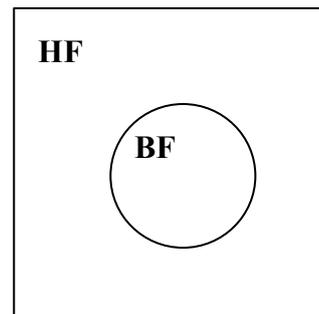
TF e dB



Original

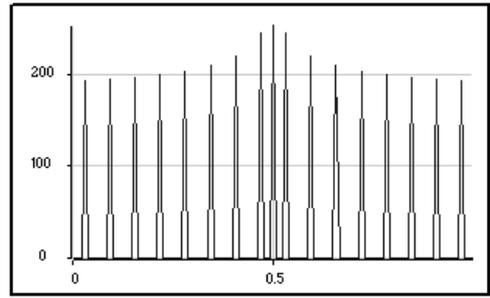
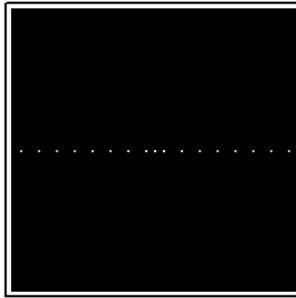
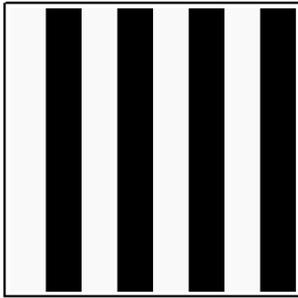


TF en dB



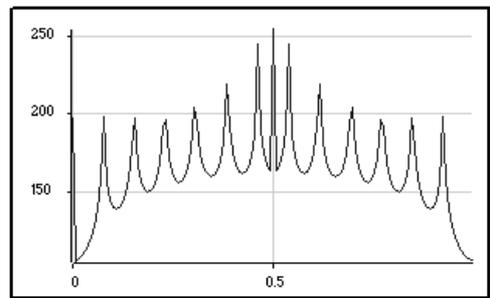
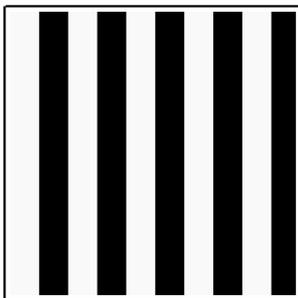
- visualisation dynamique

Ici, une TF une image périodique est réalisée, selon le même procédé que ci-dessus. Sur la TF apparaît bien la périodicité de l'image originale. L'histogramme de la TF montre la valeur moyenne entourée des deux fondamentaux et des harmoniques. Comme l'image est périodique, on vérifie que l'histogramme de la TF est bien discret.



- translation

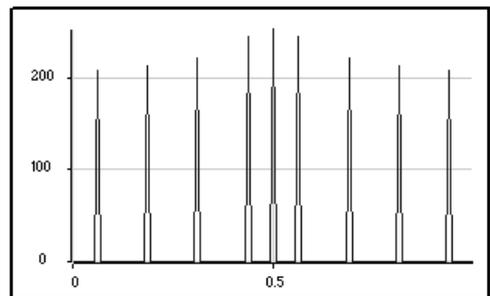
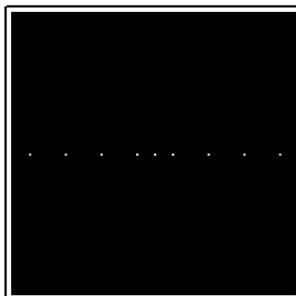
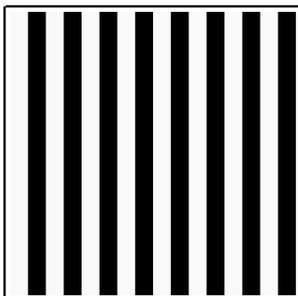
La même image que ci dessus est reprise à ceci près qu'elle n'est plus périodique car la dernière bande n'est plus entière. Le spectre devient alors continu.



- dualité

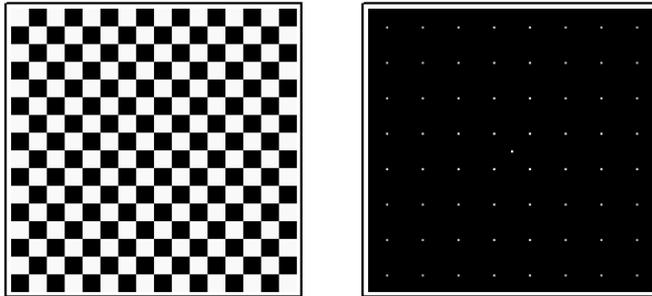
Il y a un effet de dualité entre un signal et sa TF. En effet, lorsque la période du signal augmente, l'écart entre les fréquences diminue et inversement (cette image est à comparée à la série visualisation dynamique).

Les variables spatiales et fréquentielles sont donc duales car une contraction dans un domaine entraîne une dilatation dans l'espace dual.



-2D

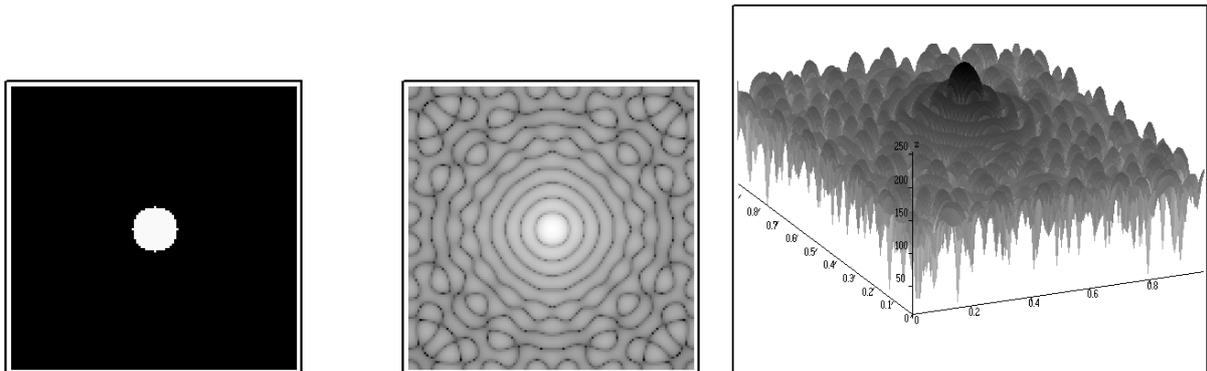
Sans surprise, la périodisation en 2D de l'image engendre une périodicité en 2D de la TF.



-disque & repliement (aliasing) :

Comme le disque n'est pas parfait du fait de l'échantillonnage, l'image n'est pas continue. Elle est donc discrète, ce qui entraîne une périodisation du spectre : c'est le phénomène de repliement spectral (aliasing).

Les superpositions du spectre provoquent des interférences, c'est pour cela que le spectre n'est pas un ensemble de cercles concentriques.



Rem 1 : La période entre deux motifs du spectre est inversement proportionnel à la moitié du pas d'échantillonnage de l'image originale.

- Equivalence entre la convolution discrète et le filtrage fréquentiel

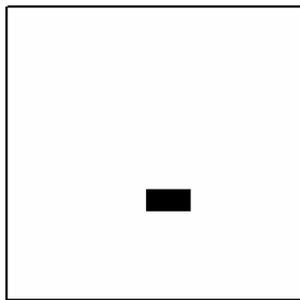
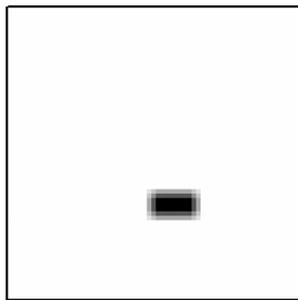
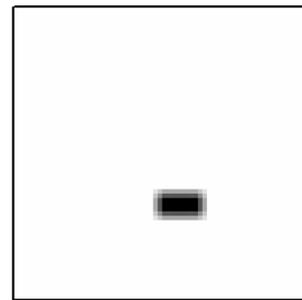


Image originale



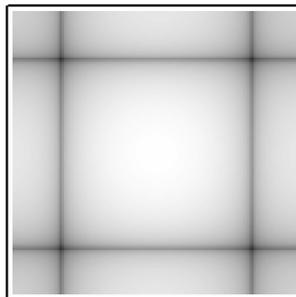
carré conv masque



TF-1(TF(masque).TF(carré)



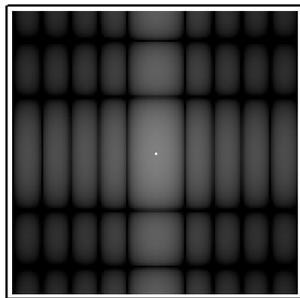
Masque



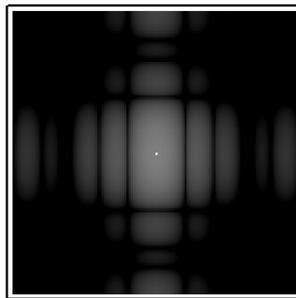
TF du masque

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

matrice du masque



TF du carré



TF(carré).TF(masque)

Il y a bien équivalence entre la convolution et la TF par la formule suivante :

$$g = f \text{ conv } h \Rightarrow \text{TF}(g) = \text{TF}(f) \cdot \text{TF}(h)$$

Annexe I : Exercice - Théorème coupe-projection

Cet exercice introduit le théorème coupe-projection, qui est à la base de l'imagerie tomographique (imagerie médicale, contrôle non-destructif...). Il établit une relation entre la transformation de Fourier 2D d'une image continue, et sa projection 1D suivant un angle θ .

1) On considère l'image continue :

$$f(x,y) = r(x+y) \cdot r(x-y) \quad \text{où} \quad r(x) = \begin{cases} |x| & \text{si } |x| \leq 1 \\ 0 & \text{sinon} \end{cases}$$

1a) Représenter l'image $f(x,y)$

1b) Calculer et commenter sa transformée de Fourier continue $F(u,v)$.

2) On définit le signal $p(x) = \int f(x,y) dy$

2a) Calculer $p(x)$ pour l'image donnée en 1)

2b) Représenter le signal obtenu.

2c) Que représente physiquement $p(x)$ par rapport à $f(x,y)$?

3) Soit $P(u)$, la transformée de Fourier continue de $p(x)$.

3a) Donner l'expression de $P(u)$.

3b) Peut-on observer une relation entre P et F ?

4) De façon générale, on définit : $p_\theta(x) = \int f(x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta) dy$

Soit $R(u)$, la transformée de Fourier continue de $p_\theta(x)$.

Montrer qu'il existe une relation entre $R(u)$ et $F(u \cos \theta, u \sin \theta)$.

5) Utiliser la question précédente pour montrer que $\{p_\theta(x) / x \in \mathbb{R} \text{ et } \theta \in [0, \pi]\}$ permet de déterminer l'image $f(x,y)$.

Annexe II : Rappel de quelques propriétés

P1- Changement d'échelle :

$$\text{si } g(x,y) = f(ax, by), \text{ avec } a \neq 0 \text{ et } b \neq 0, \text{ alors : } G(u,v) = \frac{1}{|a \cdot b|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

P2- Décalage spatial : si $g(x,y) = f(x-a, y-b)$, alors : $G(u,v) = F(u,v) \cdot e^{-2i\pi(u \cdot a + v \cdot b)}$

P3- Décalage fréquentiel : si $G(u,v) = F(u-u_0, v-v_0)$, alors : $g(x,y) = f(x,y) \cdot e^{2i\pi(u_0 x + v_0 y)}$

P4- Conjugaison : $TF^{-1}(f) = (TF(f^*))^*$ et $TF(f^*)(u,v) = (TF(f)(-u, -v))^*$

P5- Retourneement : si $g(x,y) = f(-x, -y)$, alors : $G(u,v) = F(-u, -v)$

P6- Rotation : si $f_\theta(x,y) = f(M_\theta(x,y))$, alors : $F_\theta(u,v) = F(M_\theta(u,v))$

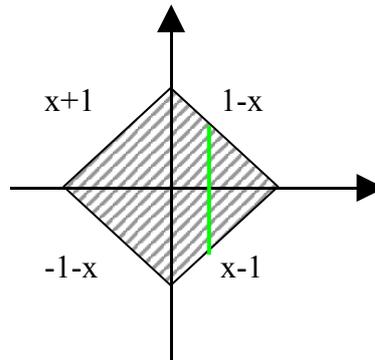
P7- Séparabilité : si $f(x,y) = g(x) \cdot h(y)$, alors : $F(u,v) = G(u) \cdot H(v)$

P8- Différentiation :

$$TF\left(\frac{\partial f}{\partial x}\right)(u,v) = 2i\pi u F(u,v) \quad TF\left(\frac{\partial f}{\partial y}\right)(u,v) = 2i\pi v F(u,v) \quad TF\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)(u,v) = -4\pi^2(u^2 + v^2)F(u,v)$$

Annexe III : Théorème coupe-projection (éléments de correction)

1a) losange



1b) $f(x,y) = r(x+y) r(x-y) = \text{rect}_2(x+y) \text{rect}_2(x-y)$ car $r = \text{rect}_2$

$$F(u,v) = \iint r(x+y) r(x-y) \exp(-2i\pi (ux+vy)) \, dx \, dy$$

on pose $x' = x+y$ alors $dx'dy' = 2 \, dx \, dy$
 $y' = x-y$

$$F(u,v) = (1/2) \iint r(x') r(y') \exp(-i\pi (u(x'+y')+v(x'-y'))) \, dx' \, dy'$$

$$F(u,v) = (1/2) \iint \text{rect}_2(x') \text{rect}_2(y') \exp(-2i\pi (\frac{u+v}{2} x' + (\frac{u-v}{2} y')) \, dx' \, dy'$$

$$F(u,v) = (1/2) [2 \text{sinc}(2 (\frac{u+v}{2}))] [2 \text{sinc}(2(\frac{u-v}{2}))]$$

$$F(u,v) = 2 \text{sinc}(u+v) \text{sinc}(u-v)$$

produit de sinc avec orientations principales sur les diagonales

2) a) $p(x) = \int f(x,y) \, dy = \int r(x+y) r(x-y) \, dy$

$p(x) = p(-x)$ donc on le calcule pour $x > 0$

Si $x > 1$ alors $r(x+y) r(x-y) = 0$

Si $0 < x < 1$ $r(x+y) r(x-y) = 1 \Leftrightarrow x-1 < y < 1-x$ (se voit graphiquement)

$$p(x) = \int_{x-1}^{1-x} dz = 1-x - (x-1) = 2 - 2x$$

Si $-1 < x < 0$ $p(x) = p(-x) = 2 + 2x$

Si $x < -1$ $p(x) = 0$

d'où $p(x) = 2(1 - \text{abs}(x))$ $r(x) = \text{tri}(x)$

b) Fct triangle

c) $p(x)$ représente la projection de l'image dans la direction 0 (somme sur des lignes verticales)

3)
$$P(u) = \int_0^1 2(1-x) \exp(-2i\pi ux) \, dx + \int_{-1}^0 2(1+x) \exp(-2i\pi ux) \, dx$$

$$P(u) = \int_0^1 2(1-x) \exp(-2i\alpha ux) dx + \int_0^1 2(1-x) \exp(+2i\alpha ux) dx$$

$$P(u) = \int_0^1 4 \cos(2\alpha ux) dx - 4 \int_0^1 x \cos(2\alpha ux) dx$$

$$P(u) = 4[\sin(2\alpha ux)]_0^1 / (2\alpha u) - 4([x\sin(2\alpha ux)]_0^1 / (2\alpha u) + 4 \int_0^1 \sin(2\alpha ux) / (2\alpha$$

u)dx

$$P(u) = 4 [-\cos(2\alpha ux)]_0^1 / (2\alpha u)^2 = 4 (1 - \cos(2\alpha u)) / (2\alpha u)^2$$

$$P(u) = 2 \sin^2(\alpha u) / (\alpha u)^2 = 2 [\text{sinc}(u)]^2 = F(u, 0)$$

4) $p_\alpha(x) = \int f(x \cos\alpha - y \sin\alpha, x \sin\alpha + y \cos\alpha) dy$

$$P_\alpha(u) = \int p_\alpha(x) \exp(-2i\alpha ux) dx$$

$$P_\alpha(u) == \int \int f(x \cos\alpha - y \sin\alpha, x \sin\alpha + y \cos\alpha) \exp(-2i\alpha ux) dx dy$$

$$P_\alpha(u) == \int \int f(x', y') \exp(-2i\alpha u(x' \cos\alpha + y' \sin\alpha)) dx dy$$

$$P_\alpha(u) == \int \int f(x', y') \exp(-2i\alpha (x'(u \cos\alpha + y' (u \sin\alpha))) dx dy$$

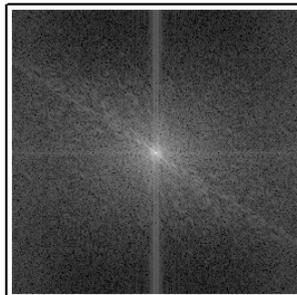
$$P_\alpha(u) = F(u \cos\alpha, u \sin\alpha).$$

5) Si on connaît $\{ p_\alpha(x) / x \in \mathbb{R} \text{ et } \alpha \in \tilde{\alpha}\pi \}$ en prenant la TF de chaque projection, on connaît $\{ P_\alpha(u) / u \in \mathbb{R} \text{ et } \alpha \in \tilde{\alpha}\pi \}$, c'est à dire la TF de l'image en coordonnées radiales. Donc par TF inverse, on peut retrouver l'image.

7. 2D Fourier Transform : real image processing

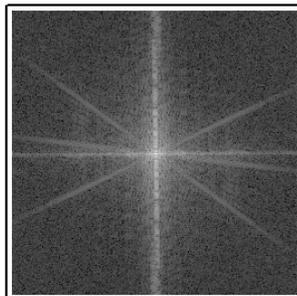
7.1. Quelques exemples

Le but de cette partie est de montrer comment il est possible d'interpréter la TF d'une image réelle. Cela est loin d'être intuitif. Pour le calcul de la TF, il est considéré que l'image est repliée sur elle-même : la suite du bord gauche est le bord droit, ...

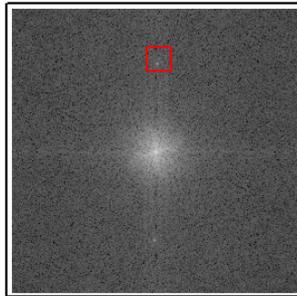
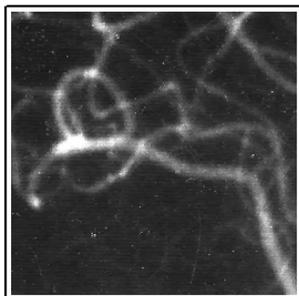


Sur la TF, apparaissent un pic central (la moyenne de l'image), le fait que la plupart des informations sont concentrées dans les BF et trois droites. Les droites représentent des discontinuités ou des contours. Elles sont perpendiculaires aux discontinuités qu'elles représentent.

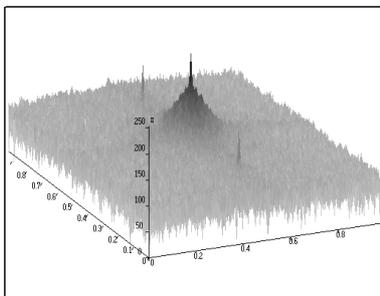
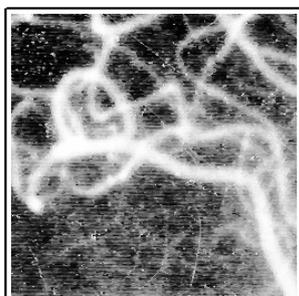
Sur l'image de la femme, les droites verticales et horizontales sont l'effet du repliement des bords. La droite oblique est l'effet engendré par le chapeau.

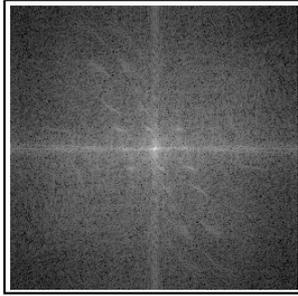


Ici on voit bien les espacements entre les dalles ou les néons.



Ici l'image est concentrée dans les BF. Deux points blancs apparaissent : ce sont les fréquences des stries de l'image.

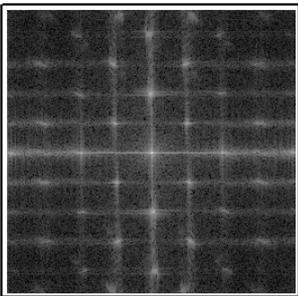
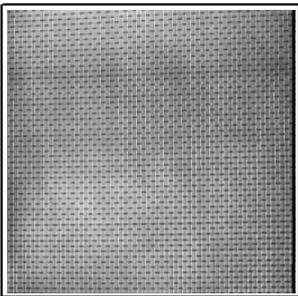




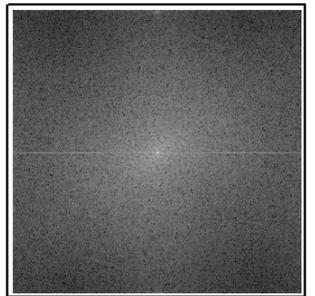
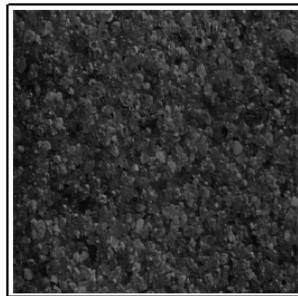
Les petits traits obliques correspondent aux fenêtres et aux drapeaux.

7.2. Textures

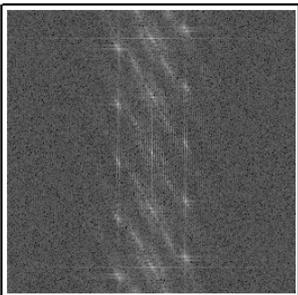
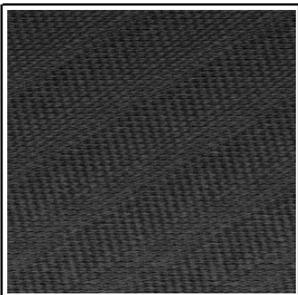
Les motifs se répétant sont caractéristiques des textures. Cela rend leur TF intéressante, car on peut observer la dualité et les informations contenues dans la TF.



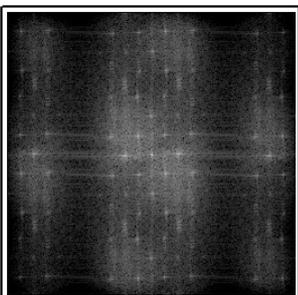
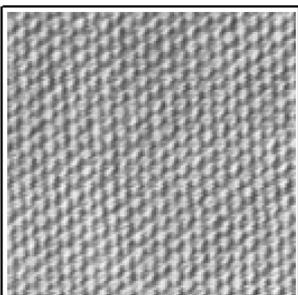
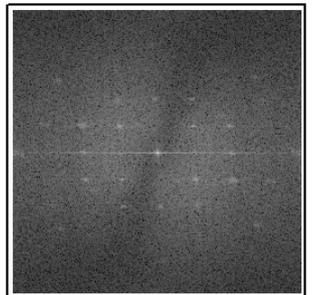
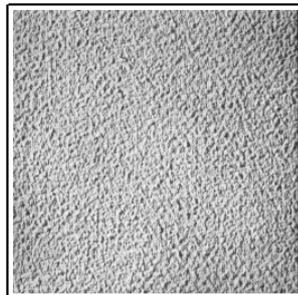
La TF permet de reconnaître une texture.



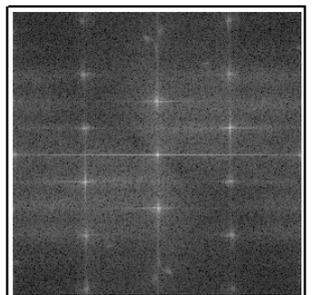
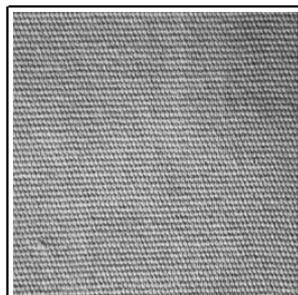
Ici, la TF est un bruit blanc.



Il y a deux types de stries : visible sur la TF



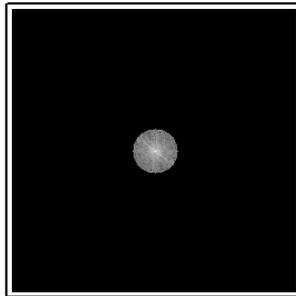
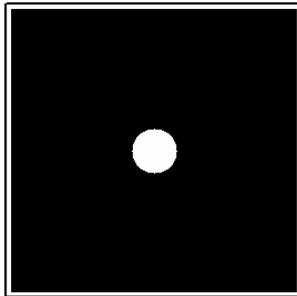
Gros motifs -> TF avec motifs rapprochés



Petits motifs -> TF avec motifs éloignés

7.3. Filtrage fréquentiel

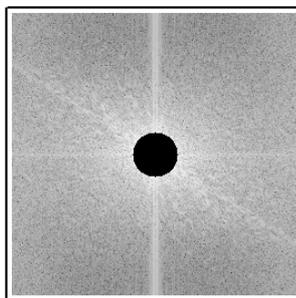
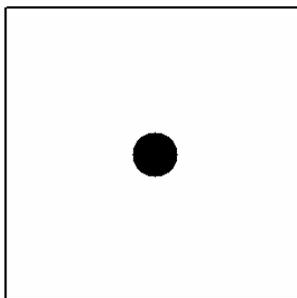
7.3.1. Passe-bas et passe-haut « idéaux »



Passe Bas idéal

Un flou apparaît

Il y a de la
résonance de
contours.

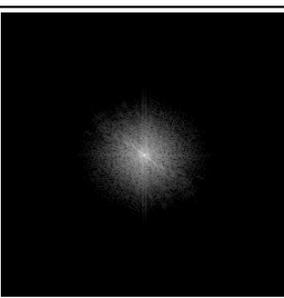
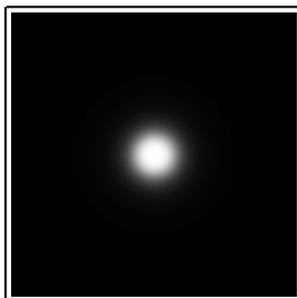


Passe haut idéal

Seuls les
contours
apparaissent.

7.3.2. Filtre de Butterworth

Comme les filtres PB et PH idéaux semblent un peu violent, du fait de leur troncature nette des fréquences, la transition peut être adoucie : c'est le filtre de Butterworth. Il a 2 paramètres : le rayon et la chute :

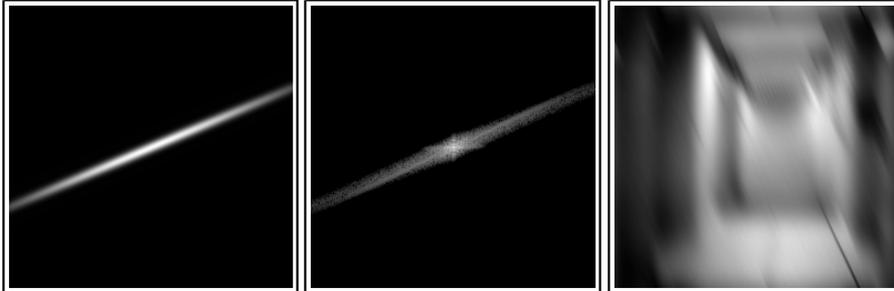


On remarque qu'avec ce filtre, l'essentiel de l'image est conservée. La majeure partie de l'information se trouve donc dans les BF. (compression)

7.3.3. Filtre de Gabor

Le filtre de Gabor est une ellipse avec une décroissance gaussienne. Il a donc trois paramètres : . Il permet de sélectionner la direction selon laquelle on ne dégradera pas l'image (ou les HF ne seront pas coupées).

Dans cet exemple, la séparation en ciment du carrelage perpendiculaire à la direction du filtre n'est pas dégradé :



Comme l'image a des stries engendrée par l'acquisition, il faut les supprimer. Il faut donc supprimer les pics sur le graphe de la TF.

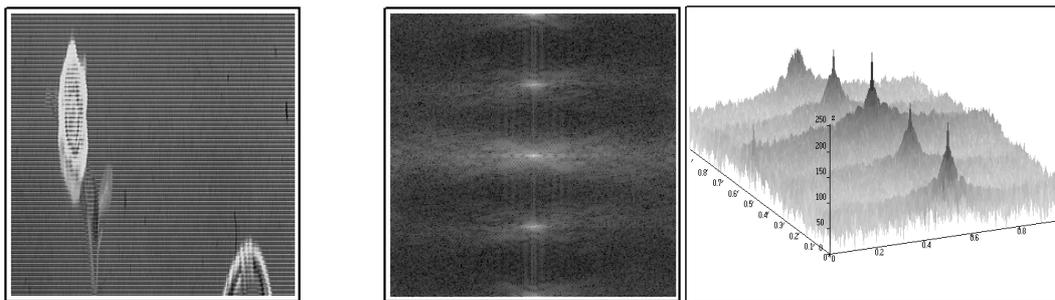
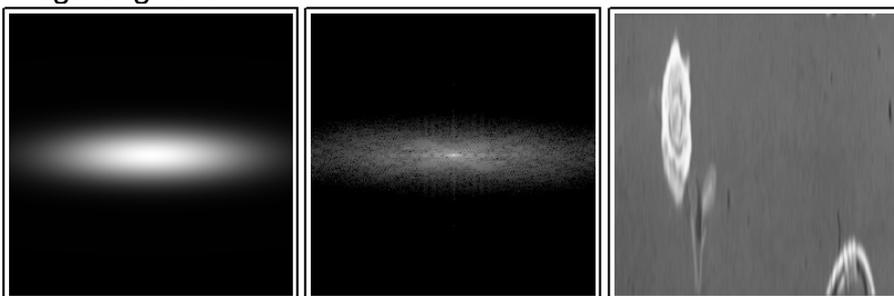


Image originale

TF 2D

TF 3D



Filtre

TF(Image finale)

Image finale

Rem : Il est possible de mettre des filtre de butterworth sur les pics pour améliorer l'image.

8. Bases of mathematical morphology

8.1. Eléments de morphologie mathématique

Les éléments de morphologie sont des opérateurs non linéaires qui comparent localement l'image avec une forme de référence (appelé élément structurant) et prennent une décision par rapport à l'environnement direct du pixel. L'idée de la morphologie mathématique est de combiner deux opérateurs duaux.

Les deux opérateurs de bases sont :

L'érosion : l'érosion est le lieu des points de l'image tel que l'élément structurant placé en ce point est inclus dans la forme.

L'effet de l'érosion est l'amincissement des formes, le gommage des petites formes, la déconnexion des éléments.

La dilatation : la dilatation est le lieu des points de l'image tel que l'élément structurant placé en ce point a une intersection non vide avec la forme.

L'effet de la dilatation est la fusion des formes. Elle est équivalente à une érosion sur le fond.



Originale



érosion avec un carré



dilatation avec un carré



Ouverture



fermeture

Les opérateurs sont indépendants du niveau de gris global (absolu). Ils ne dépendent que de l'environnement local. C'est cela qui les rend intéressants.

Alors il peut être envisager de faire de la détection de contours en manipulant des ouvertures, des fermetures et l'image originale.



Ouverture top-hat



seuillage

La transformation Top-Hat 1 correspond à l'image originale moins le résultat de l'ouverture.



Fermeture top-hat



seuillage

La transformation Top-Hat 2 correspond au résultat de la fermeture l'image original moins l'image originale.



Original – image érodée

Cette transformation engendre un problème de localisation du contour. En effet l'épaisseur symbolisant le contour se trouve à l'intérieur de l'objet et non à cheval entre les deux objets.



Image dilatée – original

Cette transformation engendre un problème de localisation du contour. En effet l'épaisseur symbolisant le contour se trouve à l'extérieur de l'objet et non à cheval entre les deux objets.



Cette transformation engendre un problème d'épaisseur du contour. En effet l'épaisseur symbolisant le contour se trouve bien à cheval mais le trait est deux fois plus épais que pour les détections de contours vues précédemment. Il est possible de palier à cet effet en diminuant la taille de l'élément structurant. La taille minimum étant bien sûr celle du pixel, la résolution de l'image devient alors le facteur limitant ; le contour a alors pour épaisseur deux pixels.

Image dilatée – image érodée

Rem 1 : Tout filtre morphologique peut se décomposer en une suite d'érosion et de dilatation.

Rem 2 : Pour l'érosion, le passage de la couleur au N&B se fait en prenant le minimum du voisinage.

Rem 3 : Pour la dilatation, le passage de la couleur au N&B se fait en prenant le maximum du voisinage.

9. Détection by corrélation

9.1. Détection par corrélation (Detection by correlation) :

voir Annexe ci-dessous

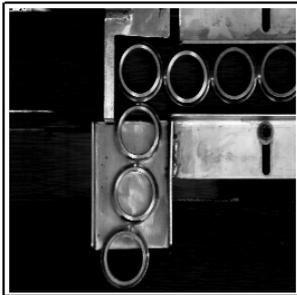
9.2. Détection par corrélation de phase (Detection by correlation of phase) :

La phase de la transformée de Fourier d'une image, bien que plus difficilement interprétable que son module, est porteuse de beaucoup d'information. Pour utiliser cette information, on peut calculer :

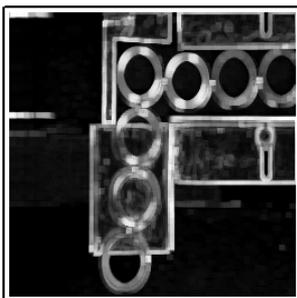
$$Corr_p(I_1, I_2) = TF^{-1} \left[\frac{TF(I_1) \times Conj(TF(I_2))}{|TF(I_1) \times Conj(TF(I_2))|} \right] = TF^{-1} \left[\Phi(TF(I_1)) \times Conj(\Phi(TF(I_2))) \right].$$

Avec qui est liée à la phase de l'image

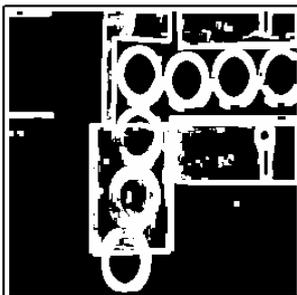
9.3. Application :



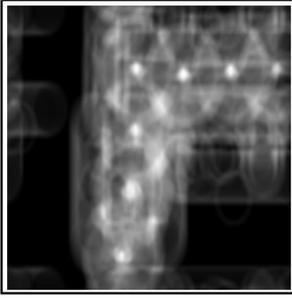
Le but de cette application est de compter les roulements à billes de cette image. Comme l'image semble déformée du fait de la forme des capteurs de la caméra CCD, il faut le prendre en compte. En effet, il semble que les capteurs soient des rectangles de rapport de côté 3/2, et non des carrés. Cela donne une forme elliptique aux roulements. On effectue alors un gradient morphologique.



Après l'application du gradient morphologique à l'image, on observe qu'une partie des zones claires de l'image a disparue. Seuls les contours restent encore en clair.

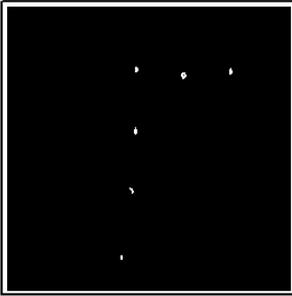


On effectue alors un seuillage, qui correspond à une prise de décision sur les éléments sur lesquels on aura à faire un traitement.



On effectue un corrélation avec image d'ellipse blanche sur fond noir, de même taille que les roulements à billes de l'image.

Il apparaît des pics de blanc qui symbolisent là où l'image est la plus ressemblante à l'ellipse blanche. Comme les roulements ne sont entièrement blancs et certains contours peuvent se confondre avec les roulements ; des parasites apparaissent alors.

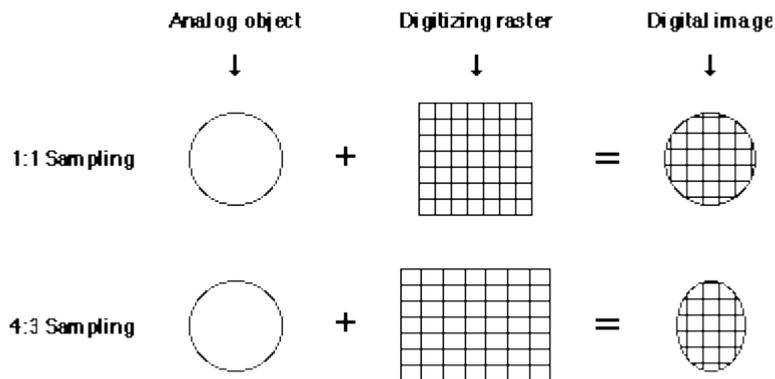


Le seuillage fait alors office de filtre. Mais on remarque que le résultat n'est pas parfait : il manque un roulement (en haut à droite).

Pour corriger ce problème on peut réajuster le seuillage, ou prendre une image réelle de roulement pour faire la corrélation.

Rem 1 : Le mouvement, la perspective et la forme des capteurs peuvent source de déformations.

Exemple : forme des capteurs non carré



Rem 2 : Pour palier au fait que l'on a pas tous les roulements, on peut créer un cache car la localisation des roulements sur l'image est connue.

Rem 3 : Pour palier au fait que l'on a pas tous les roulements, on peut utiliser un filtre de Gabor .

Rem 4 : Pour implanter la corrélation de manière informatique, il est préférable de passer par la TF. En effet : $TF(f \otimes h) = TF(f) \cdot TF(h)^*$.

Annexe : Corrélation d'images continues

Le produit de corrélation associé à deux images, $f(x,y)$ et $g(x,y)$, une nouvelle image, notée $f \otimes g(x,y)$, définie par :

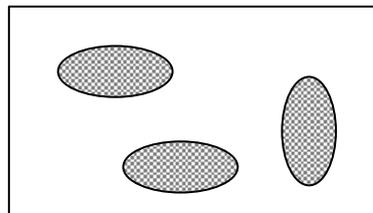
$$f \otimes g(x,y) = \iint f(\alpha, \beta) \cdot g(\alpha - x, \beta - y) d\alpha d\beta \quad (1)$$

- 1) Exprimer la relation (1) en termes d'un produit de convolution.
- 2) Exprimer la transformée de Fourier du produit de corrélation à partir des transformées de Fourier de f et g . On notera F l'opérateur de transformation de Fourier.
- 3) On pose $g(x,y) = f(x - x_0, y - y_0)$. Donner l'expression de $g \otimes f(x,y)$ en fonction de la fonction d'autocorrélation de f , $A_f(x,y) = f \otimes f(x,y)$.
- 4) On pose $f(x,y) = \text{rect}_X(x) \cdot \text{rect}_Y(y)$.
 - a) Calculer la fonction d'autocorrélation de f , et la représenter.
 - b) soit $g(x,y) = \text{rect}_X(x - x_1) \text{rect}_Y(y - y_1) + \text{rect}_X(x - x_2) \text{rect}_Y(y - y_2)$,
Calculer $f \otimes g(x,y)$.
- 5) On peut montrer que la fonction d'autocorrélation d'une image f , $A_f(x,y)$ est maximale lorsque $x = y = 0$. En déduire une méthode pour identifier une forme $h(x,y)$ dans une image $g(x,y)$. Présenter la méthode sous la forme d'un schéma algorithmique. Combien d'éléments cette méthode permettra-t-elle de détecter dans l'image ci-dessous ? Comment adapter la méthode pour que tous les éléments soient détectés (Donner le schéma algorithmique correspondant) ?

$h(x,y)$:



$g(x,y)$:



Eléments de correction

- 1) $f \otimes g(x,y) = f * g^-(x,y)$ avec $g^-(x,y) = g(-x,-y)$
- 2) $F(f \otimes g)(u,v) = Ff(u,v) Fg(-u,-v)$
- 3) $f \otimes g(x,y) = f \otimes f(x-x_0, y-y_0) = A_f(x-x_0, y-y_0)$
- 4) a) $A_f(x,y) = A_{\text{rect}_X}(x) A_{\text{rect}_Y}(y) = \text{tri}_X(x) \text{tri}_Y(y)$
 b) $f \otimes g(x,y) = A_f(x-x_1, y-y_1) + A_f(x-x_2, y-y_2)$
 $= \text{tri}_X(x-x_1) \text{tri}_Y(y-y_1) + \text{tri}_X(x-x_2) \text{tri}_Y(y-y_2)$
- 5) a) Méthode pour identifier une forme $h(x,y)$, dans une image $g(x,y)$:
 Pour chaque translation x_0, y_0
 calcul de $f \otimes g(x,y)$ avec $g(x,y) = f(x-x_0, y-y_0)$
 Prendre (x_0, y_0) qui réalise le max de $A_f(x-x_0, y-y_0)$

// ou garder les points qui donnent une valeur au dessus d'un certain seuil (si on a plusieurs objets à détecter)

b) La méthode permettra de détecter 2 éléments dans l'image ci-dessous

c) Pour détecter tous les éléments, il faut introduire une rotation :

Pour chaque translation x_0, y_0

Pour chaque rotation α

calcul de $f \otimes g(x,y)$ avec $g(x,y) = f(M_\alpha(x-x_0, y-y_0))$

Prendre (x_0, y_0, α) qui réalise le max de $A_f(x-x_0, y-y_0)$

//idem

10. Restoration and inverse filtering

La démarche est de modéliser la dégradation pour pouvoir la corriger, plutôt que d'essayer un tas de filtres au hasard.

10.1.1 Unhook the moon !

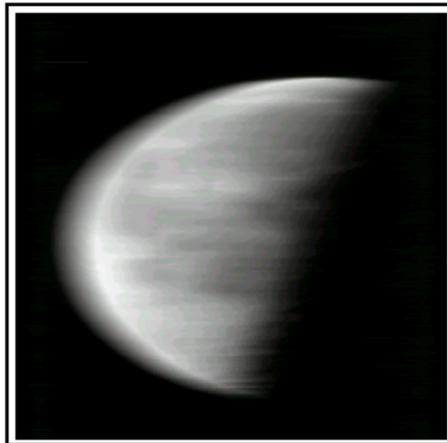
Chaque pixel de l'image bruitée (g) est une CL des ses voisins dans l'image originale (f). En effet, l'image originale semble déformée par un mouvement lors de la prise de vue, et cela se traduit par un moyennage de l'intensité selon la direction du mouvement.

Image originale = TF⁻¹ (image dégradée / masque) + TF⁻¹ (bruit / masque)

Comme on ne connaît pas le masque, il faut l'identifier. Pour cela on recherche un endroit sur l'image où l'on croit connaître la forme de départ. Souvent c'est la réponse à une impulsion que l'on cherche, ainsi on dispose directement de la fonction de transfert de l'effet du masque -qui est une transformation. Une impulsion est un point d'un niveau de gris sur un fond d'un niveau de gris différent.

Ici on cherche l'effet de la dégradation sur une étoile (point blanc sur un fond noir).

One would like to analyze the surface of the moon, but, hell !, the acquired image is of poor quality. As a matter of fact, a movement of the camera during the long-exposure acquisition led to a fuzzy image...



10.1.2 Modelling the problem

A fairly simple mathematical model can be given for the degradation undergone by this picture. The resulting grey level of each pixel is the average grey level "seen" by the pixel during the movement. Assuming the movement is rectilinear, this is

modelled by a convolution of the original image $f(x,y)$ (with no noise) by a linear mask whose size corresponds to the movement amplitude.

With: $f(x,y)$ the original image (unknown)

$h(x,y)$ the convolution mask or impulse response (length unknown)

$g(x,y)$ the observed fuzzy image

we have:

$$g=f * h.$$

There can also be some unknown noise $b(x,y)$. Assuming it is additive, we have:

$$g=f * h+b.$$

Computing the Fourier Transform of the previous equation, we get:

$$G=F.H+B$$

and then :

$$F=G/H-B/H$$

Assuming the noise can be neglected, this turns to :

$$F=G/H$$

The original image $f(x,y)$ can then be recovered by computing the inverse Fourier Transform.

This restoration technique is called by (pseudo) inverse filtering ($f(x,y)$ is restored thanks to a filter whose transfer function is $1/H$)

However, directly applied, this method may fail. Actually, for the frequencies where H gets null (or take values extremely close to zero), B/H cannot be neglected any longer: the restored image is then heavily corrupted by the noise. This problem can be tackled by using a filter whose transfer function is $1/H$ when $|H|$ is greater than a given threshold \square , and zero otherwise (pseudo-inverse filtering).

10.1.3 Identifying the model's parameters

Intuitivement, la déformation semble être celle d'un défaut de focale : une traînée selon la direction horizontale. On fait alors plusieurs coupes verticales pour voir si il y a une discontinuité sur le fond (éventuellement après un seuillage). Oh miracle, il y en a une en haut à gauche ligne 32. On refait une coupe sur la ligne 32 pour voir la taille du trait : 27.

The first operation consist in identifying the parameters of the designed model.

With astronomic images, one can be lucky... if a far star is in the black background of $f(x,y)$, it can be consider as a Dirac. In $g(x,y)$, it then directly leads to the impulse response of the filter who degraded the image (*i.e.* h)...

Look for such a star in the dark regions of g (use profiles, thresholds...).

If one can't directly estimate h from g , another solution consists in the trial-error strategy.

10.1.4 Using the identified model

The image can now be restored.

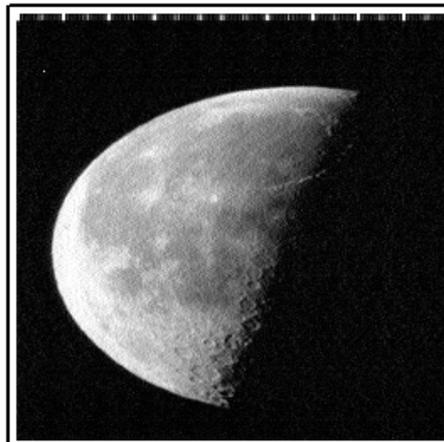
Try various values of σ , try to slightly change the size of the mean filter...
Please comment.

10.1.5 Further studies...

Once a reasonably good quality obtained, don't stop !

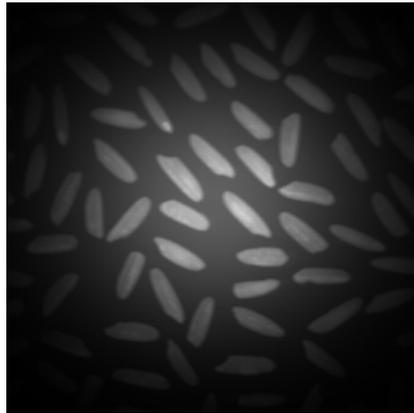
You can most possibly further improve the result (contrast, edges...). You can compare this method with other filters providing a contrast enhancement (leading to less fuzzy transitions).

You can also start to analyse the restored image. For instance, try to extract some craters, derive quantitative results (area, diameter...)



10.2.1 A portion of rice...

One would like to analyze a picture featuring some rice seeds. The aim is to obtain a segmented binary image, but, hell !, this is not that easy since the acquired image is of poor quality. As a matter of fact, the background is not uniform, and neither are the seeds...

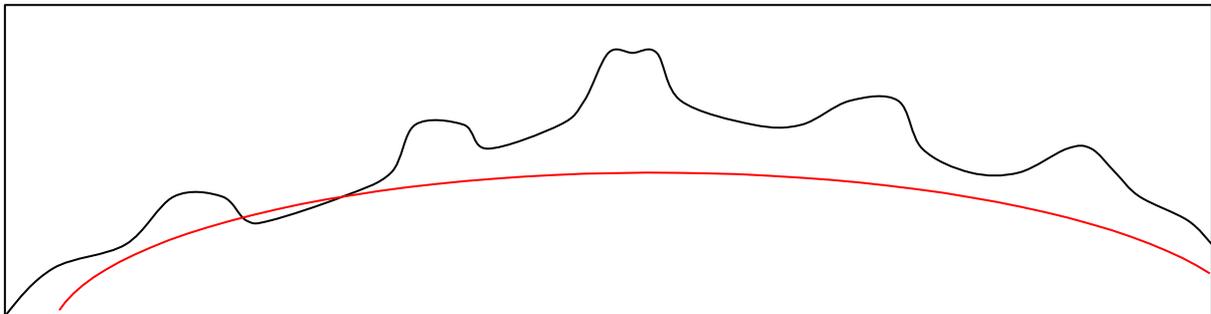


What shall we do ?

Dans ce cas le problème est radicalement différent. En effet la dégradation de l'image est clairement causée par les défauts de l'appareil numérique : l'appareil est moins sensible au contraste sur les bords.

Pour dénombrer les grains de riz, un seuillage pourrait faire l'affaire si le fond au centre de l'image était plus sombre que les grains de riz du bord de l'image. Ce n'est pas le cas. Il faut donc essayer autre chose.

Une coupe peut nous permettre de mieux comprendre cette dégradation :



Les pics sont les grains de riz. Il apparaît que le fond possède une déformation comme une coupole. Il serait intéressant de soustraire cette déformation à l'image. Puis ensuite de faire une seuillage et une fermeture. Alors les grains de riz pourraient être dénombrés.

On a le choix de modéliser le dôme ou de prendre une image à vide (ou image du noir) si l'on possède l'appareil. La technique de photo du noir est souvent utilisée, notamment en astronomie pour corriger les défauts des appareils d'observation.

Conclusion : Chaque image exige un traitement différent et adapté. Le travail de l'ingénieur est justement d'utiliser les outils adéquates parmi les techniques étudiées.

11. Diverses ressources

Logiciels mathématiques :

Matlab :

Scilab (GNU) : <ftp.inria.fr/INRIA/Projects/Scilab> , <http://www.scilab.org>

maple. (Waterloo) : <http://www.maplesoft.com>

Mathematica (Wolfram Research) : <http://www.wolfram.com>

:

Logiciels de traitement des images :

Photoshop (Adobe) :

the Gimp (GNU) :

Mustig (GNU) : <http://www.lis.inpg.fr>

Digital Image Processing (Wolfram Research) : <http://www.wolfram.com>

Cours en ligne :

- **Université de elft (en anglais) :**
<http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip.html>
- **Livres en ligne (en anglais) :**
<http://homepages.inf.ed.ac.uk/rbf/CVonline/books.htm>
- **Plus d'infos sur les images couleurs :**
<http://fr.selfhtml.org/graphisme>

Agences :

- **Agence numérique** : <http://www.agencenumerique.com>
- Agence numérique Rhône Alpes : <http://www.numera.org>

Droit : http://www.websitekit.org/site_legamedia/mainf.html

Remerciements : Jocelyn Chanussot,