

UNIVERSITÉ PARIS IX DAUPHINE
IUP GÉNIE MATHÉMATIQUE ET INFORMATIQUE

Formation continue
Maîtrise de Génie Mathématique et Informatique
Option : Informatique

Traitement numérique du signal et de l'image

Table des matières

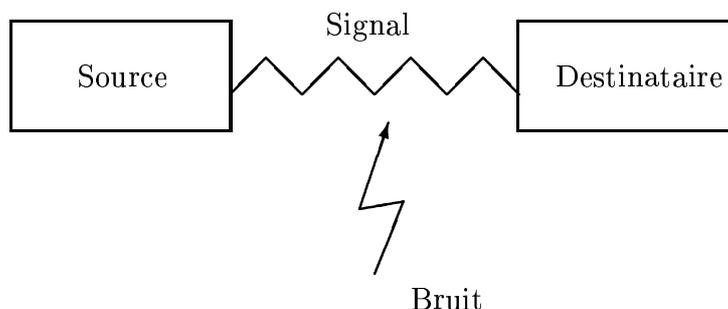
1	Introduction	1
2	Traitement du signal	5
2.1	Signaux numériques.	5
2.2	Systèmes discrets.	17
2.2.1	Systèmes linéaires invariants.	18
2.2.2	Filtres discrets bidimensionnels.	25
2.3	Échantillonnage et quantification	32
2.3.1	Échantillonnage de signaux continus.	32
2.3.2	Quantification de signaux analogiques	38
2.3.3	Convertiseur A/D.	39
2.4	Transformation de Fourier discrète	41
2.4.1	Signaux périodiques.	41
2.4.2	Signaux à durée limitée.	42
2.4.3	Propriétés de la transformation de Fourier discrète.	47
2.4.4	Calcul rapide de la TFD (TFR/FFT)	52
2.4.5	Transformation de Fourier discrète bidimensionnelle	58
3	Traitement de l'image	61
3.1	Analyse d'images	61
3.1.1	Détection de bords et de contours	61
3.1.2	Segmentation d'images	69
3.1.3	Opérations morphologiques	75
3.2	Amélioration de l'image.	84
3.2.1	Rehaussement d'images	84
3.2.2	Lissage et restauration d'images	88
3.3	Compression d'images	95
3.3.1	Le standard JPEG	95
3.3.2	Le format GIF	97
	Bibliographie	98

1 Introduction

Concepts et vocabulaire de base

En un premier temps on va définir et délimiter le sujet abordé dans ce cours.

On définit un signal de la manière suivante : Un *signal* est la représentation physique de l'information qu'il convoie de sa source à son destinataire.



Un signal s sera représenté par une fonction d'un espace E dans un espace F :

$$s : E \longrightarrow F.$$

On peut distinguer la *théorie du signal*, où l'on s'occupe de la description mathématique des signaux et de leurs transformations, du *traitement du signal*, qui s'occupe de l'élaboration, du transport et de l'interprétation des signaux, considérés comme porteurs d'informations.

Parmi les signaux on distingue les *signaux analogiques* qui représentent l'information par une grandeur physique, et les *signaux numériques* (digitaux) qui représentent l'information par des suites de nombres.

Pour les premiers, l'espace de définition est $E = \mathbb{R}^n$, ce sont les phénomènes physiques classiques continus que l'on rencontre dans la nature : courant électrique, un son, un signal radio,...

Pour les signaux numériques $E = \mathbb{Z}^n$, ce sont typiquement les signaux numériques idéaux traités par un ordinateur, on parle de *signaux discrets*.

Exemple : Comparer la façon dont un morceau de musique est stocké sur un disque vinyle et un disque compact.

Suivant la dimension, n , de l'espace de départ, E , c'est-à-dire le nombre de variables dont dépend le signal, on a pour :

$n = 1$, un signal monodimensionnel (son, cours de la bourse, ...), dans ce cas la variable unique est souvent considérée comme étant le *temps*.

$n = 2$, un signal bidimensionnel (image numérique, photo, ...).

$n = 3$, un signal 3D (vidéo(1,c,t), hologramme(x,y,z), ...).

L'espace d'arrivée F , où le signal prend ses valeurs, sera \mathbb{R}^p ou \mathbb{Z}^p .

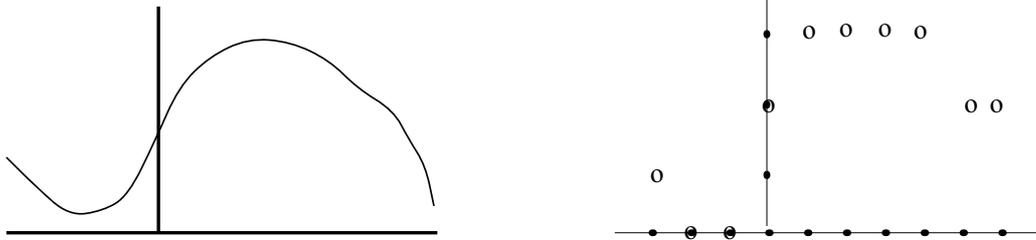
Dans la modélisation mathématique on distingue entre *signaux déterministes* et *signaux aléatoires*, dans ce cours on va se restreindre à une approche déterministe.

Le passage «analogique \Leftrightarrow digital», sans perte, est un problème important en traitement du signal. En effet, comme on a vu plus haut, les signaux que l'on rencontre dans le monde réel (physique) sont de nature analogique, continues en leurs variables et valeurs.

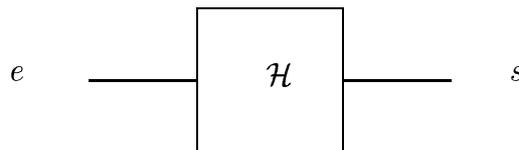
Par contre les entrées et les sorties d'un ordinateur sont *numériques*, c'est-à-dire discrètes en variables et valeurs.

Pour convertir un signal continue en un signal discret on fait un *échantillonnage* : la valeur du signal est mesurée à certains instants/positions. Pour convertir les valeurs analogiques d'un signal en valeurs discrètes on fait une *quantification* : les valeurs du signal sont prises dans un ensemble fini.

Sur la figure suivante on a représenté un signal continue et le même signal échantillonné et quantifié.



Sous l'appellation *système* on regroupe les phénomènes qui peuvent agir sur un signal. On utilise souvent la représentation sous forme de «boîte noire», à une entrée e on fait correspondre une sortie s .



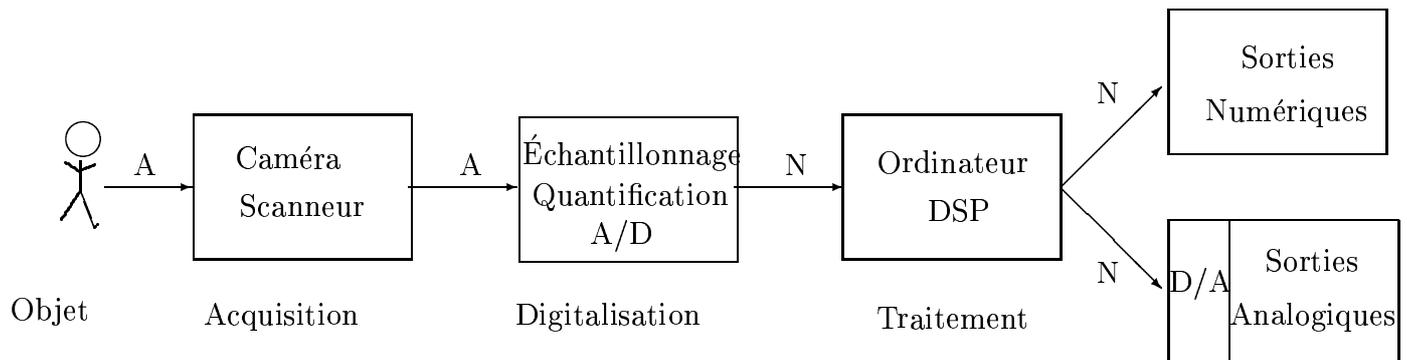
\mathcal{H} est un opérateur : à une fonction $e : E \rightarrow F$ on associe une fonction $s : E \rightarrow F$ et on écrit $s = \mathcal{H}[e]$. Suivant les propriétés de \mathcal{H} , détectées en utilisant des entrées tests, on pourra trouver des modélisations mathématiques précises.

Contenu du cours

Dans la première partie du cours on va étudier le traitement du signal et introduire les principaux outils utilisés pour le traitement des signaux monodimensionnels et bidimensionnels. Dans la deuxième partie on va s'intéresser au traitement de l'image, signal bidimensionnel spatial. On verra les méthodes spécifiques à ce domaine.

Pour terminer cette introduction énonçons quelques applications du *traitement de l'image* et les problèmes soulevés.

La figure suivante présente les principales composantes qui forment une chaîne de traitement d'images. Par 'A' on indique un signal analogique, et par 'N' un signal numérique, véhiculé à travers la chaîne. On peut noter que les appareils d'acquisition actuels effectuent souvent la digitalisation, et même la compression, des données. Le traitement est effectué par ordinateur ou *digital signal processor* (DSP).



Les problèmes fondamentaux qui se posent, et que l'on va aborder dans la partie traitement de l'image, sont les suivants :

- Amélioration de la qualité de l'image.

Les techniques de *débruitage* sont utiles comme prétraitement, mais peuvent détruire/affecter l'information initiale.

- Analyse d'images.

On essaye d'obtenir des informations quantitatives comme la surface, la quantité ou la forme, d'objets contenus dans l'image.

- Compression d'images.

On veut réduire la taille des données sans perte d'information pertinente, ceci pour pouvoir transmettre ou sauvegarder les images à moindre frais.

Les applications du traitement de l'image sont nombreuses :

Les images satellites servent à détecter des richesses naturelles, à la cartographie, à la prévision météorologique,...

L'imagerie médicale sert à la détection d'anomalies, à faire des mesures sans intervention directe,...

Les images radar, infrarouges permettent la détection de cibles, de reconnaître des objets,...

La transmission d'images vidéo est importante pour la surveillance de sites, transfert de dossiers médicaux, téléconférences,...

Un enjeu important est l'*indexation*, il s'agit de retrouver dans des grandes bases de données d'images ou films numériques, les images qui contiennent un certain objet (ou personne, ...).

Dans ce cours on ne parlera pas de la vision humaine. On notera cependant que certains résultats de psychovision sont utiles pour les applications en traitement de l'image :

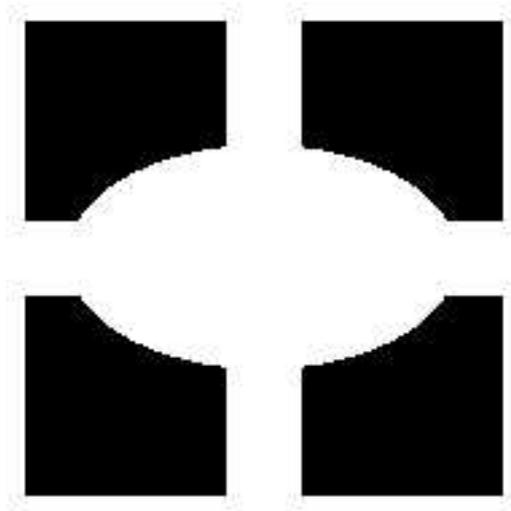
- Ce n'est pas l'intensité absolue qui compte, mais le changement d'intensité entre régions voisines.

- Les occultations et transparences sont complétées de façon naturelle par le système de vision humaine.

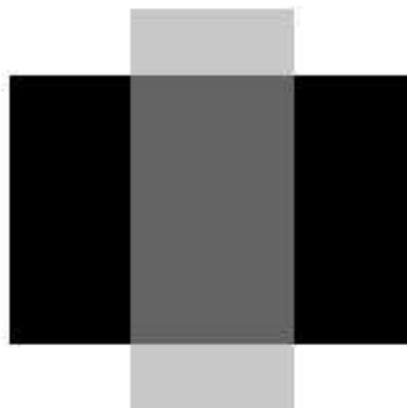
Les images suivantes illustrent ces propriétés de la vision humaine.



Quatre carrées et seulement trois niveaux de gris distincts (1, 160, 180).
(Importance du contraste)



Quatre carrées noirs et un ballon du rugby blanc?
(Traitement des occultations)



Deux rectangles mais trois niveaux de gris distincts (0, 100, 200).
(Transparence)

2 Traitement du signal

2.1 Signaux numériques.

Un signal numérique, ou discret, est une fonction définie sur \mathbb{Z}^n .

Définitions :

- Un *signal discret monodimensionnel* est une suite de réels $(x(n))_{n \in \mathbb{Z}}$.
- Un *signal discret bidimensionnel* est une suite de réels à double indice, $(x(i, j))_{(i, j) \in \mathbb{Z}^2}$.
- Le *support* d'un signal discret est le plus petit ensemble connexe où le signal prend des valeurs non nulles.

Pour un signal monodimensionnel c'est un intervalle, pour un signal bidimensionnel on choisira souvent un rectangle.

- Un signal discret est à *support compact* si le support contient un nombre fini de points, *i.e.* si et seulement si un nombre fini d'éléments prend des valeurs non nulles. Pour un signal monodimensionnel on parle de *durée*.
- Si n_i , resp. n_f , est le premier, resp. dernier point où x est non nul, alors la *durée* de x est $n_f - n_i + 1$.

Exemples :

- (i) $u(n) = 0$ si $n < 0$ et $u(n) = 1$ si $n \geq 0$, c'est l'*échelon unitaire*.
- (ii) $x(n) = |n|$.
- (iii) $x(n) = 1$ si $n = 0$ et $x(n) = 0$ sinon. C'est la *fonction de Kronecker*, notée $\delta(n)$.
- (iv) Soit $R = [0, 255] \times [0, 255]$, pour tout $(i, j) \in R$ on définit

$$x(i, j) = \begin{cases} 0 & , (i - 127)^2 + (j - 127)^2 < 64 \\ 255 & , \text{sinon} \end{cases} .$$

Note : On dit que $x(i, j)$ est un *pixel* (angl. picture element).

Remarque : Dans le monde réel un signal discret monodimensionnel est souvent supposé commencer à l'instant $n = 0$, c'est-à-dire qu'il n'existe pas dans le «passé» ($n < 0$).

On a $\text{support}(x) \subset \mathbb{Z}_+$ et on dit que x est *causal*.

On va présenter en un premier temps les définitions, opérations et propriétés des signaux discrets monodimensionnels, ces définitions se généralisent facilement à la dimension deux, comme on verra plus tard.

Signaux discrets monodimensionnels.

Définition : Un signal discret monodimensionnel x , est *périodique*, de période N ($N \in \mathbb{N}$), si et seulement si :

$$\forall n \in \mathbb{Z} : x(n) = x(n + N) .$$

Exemples : $x(n) = (-1)^n$, $x(n) = e^{i2\pi n/N}$.

Définition : On définit la *translation* du signal discret monodimensionnel x , de la quantité $\alpha \in \mathbb{Z}$, par

$$\forall n \in \mathbb{Z} : \tau_\alpha x(n) = x(n - \alpha) .$$

Le signal $\tau_\alpha x$ est le *retardé* de x , de retard α .

Exemple : $\delta_\alpha = \tau_\alpha \delta$.

Opérations de base sur les signaux.

Les opérations de base que l'on peut effectuer sur les signaux discrets monodimensionnels (d.m.) sont :

- Addition deux signaux x et y . Le résultat est un signal d.m. $z = x + y$, défini comme suit

$$\forall n \in \mathbb{Z} : z(n) = x(n) + y(n).$$

- Multiplication d'un signal x par un scalaire $\lambda \in \mathbb{C}$, on obtient un signal d.m. $z = \lambda x$:

$$\forall n \in \mathbb{Z} : z(n) = \lambda x(n).$$

- Multiplication d'un signal x par un signal y , on obtient un signal d.m. $z = x \cdot y$:

$$\forall n \in \mathbb{Z} : z(n) = x(n) \cdot y(n).$$

Exercice : Vérifier que ces opérations sont stables sur l'ensemble des signaux discrets monodimensionnels (à support compact).

Propriétés :

L'addition et la multiplication des signaux hérite des propriétés bien connues sur les nombres réels (resp. complexes).

L'addition et la multiplication par un scalaire induisent une structure d'espace vectoriel sur l'ensemble des signaux.

Corrélation :

Soient x et y deux signaux numériques à durée limitée. On définit un signal φ_{xy} par :

$$\forall n \in \mathbb{Z} : \varphi_{xy}(n) = \sum_{k \in \mathbb{Z}} x(k-n)y(k).$$

On appelle φ_{xy} la *fonction d'intercorrélation* des signaux x et y .

Propriétés de la corrélation :

- Si x , resp. y , est de durée M_1 , resp. M_2 , alors φ_{xy} est un signal de durée $M_1 + M_2 - 1$.
- On a $\varphi_{yx}(n) = \varphi_{xy}(-n)$, pour tout $n \in \mathbb{Z}$.

Si $x = y$ on dit que φ_{xx} est la *fonction d'autocorrélation* du signal.

Convolution :

La *convolution* de deux signaux x et y , notée $z = x \star y$, est définie par :

$$\forall n \in \mathbb{Z} : z(n) = \sum_{k \in \mathbb{Z}} x(n-k)y(k).$$

Pour que z existe il suffit que x , ou y , est un signal d.m. à support compact.

Propriétés de la convolution :

- Si x , resp. y , est de durée M_1 , resp. M_2 , alors $x \star y$ est un signal de durée $M_1 + M_2 - 1$.
- On a $x \star y = y \star x$.
- On a $\delta \star x = x$.

- Si $z = x \star y$, alors $\sum_{k \in \mathbb{Z}} z(k) = \left(\sum_{k \in \mathbb{Z}} x(k) \right) \left(\sum_{k \in \mathbb{Z}} y(k) \right)$.

Convolution circulaire :

On se propose de définir une opération de convolution pour les signaux périodiques.

Soient x et y deux signaux périodiques de période N , on considère la restriction des deux signaux sur $\{0, 1, \dots, N-1\}$ et on définit la convolution circulaire, notée $z = x \star_c y$, par

$$\forall n \in \mathbb{Z} : z(n) = \sum_{k=0}^{N-1} x(n-k)y(k) = \sum_{k=0}^{N-1} x((n-k)_N)y(k).$$

On utilise $x(-k) = x(N-k)$, resp. $-k = N-k \pmod{N}$, et l'on obtient ainsi un signal discret périodique de période N .

Énergie d'un signal : L'énergie d'un signal discret est définie par

$$E[x] = \sum_{k \in \mathbb{Z}} |x(n)|^2.$$

On ne s'intéresse qu'aux signaux d'énergie finie, en particulier un signal à durée déterminée est d'énergie finie.

Exemples et remarques :

- Soit x_a un signal analogique, *i.e.* $x_a : \mathbb{R} \rightarrow \mathbb{R}$. On en déduit un signal discret en posant

$$x_d(n) = x_a(nT), \quad n \in \mathbb{Z}.$$

On dit que le réel T est la *période d'échantillonnage* du signal analogique x_a et $1/T$ est la *fréquence d'échantillonnage*.

- Signal rectangulaire (de largeur L), défini par

$$rect_L(n) = \begin{cases} 1 & , \text{ si } 0 \leq n \leq L-1 \\ 0 & , \text{ sinon} \end{cases}.$$

- Relations entre les signaux de base :

$$u(n) = \sum_{k=-\infty}^n \delta(k) = \sum_{k=0}^{+\infty} \delta(n-k),$$

$$\delta(n) = u(n) - u(n-1) \quad rect_L(n) = u(n) - u(n-L).$$

- Signal sinusoïdal (réel) : $x(n) = A \cos(\omega n + \phi)$, $\forall n \in \mathbb{Z}$, où A , ω et ϕ sont des réels.

On appelle A l'*amplitude*, ω la *fréquence* et $\omega n + \phi$ la *phase* du signal, ϕ est le *déphasage*.

- Signal exponentiel (complexe) : $x(n) = Ae^{i(\omega n + \phi)}$, $\forall n \in \mathbb{Z}$, ici $A \in \mathbb{R}_+$, ω et ϕ étant des réels quelconques, on utilise le même vocabulaire que dans le cas réel.

Soit $\omega' = \omega + 2\pi k$, alors $x(n) = Ae^{i(\omega n + \phi)} = Ae^{i(\omega' n + \phi)}$, c'est-à-dire que l'on ne peut pas distinguer deux signaux exponentiels discrets dont les fréquences sont identiques modulo 2π .

Il suffit donc de considérer les signaux de la forme $x(n) = Ae^{i(\omega n + \phi)}$,

resp. $x(n) = A \cos(\omega n + \phi)$, pour des fréquences ω appartenant à un intervalle de longueur 2π (par exemple $[-\pi, \pi[$ ou $[0, 2\pi[$).

- Déterminons les signaux exponentiels discrets périodiques de période $N \in \mathbb{Z}$.

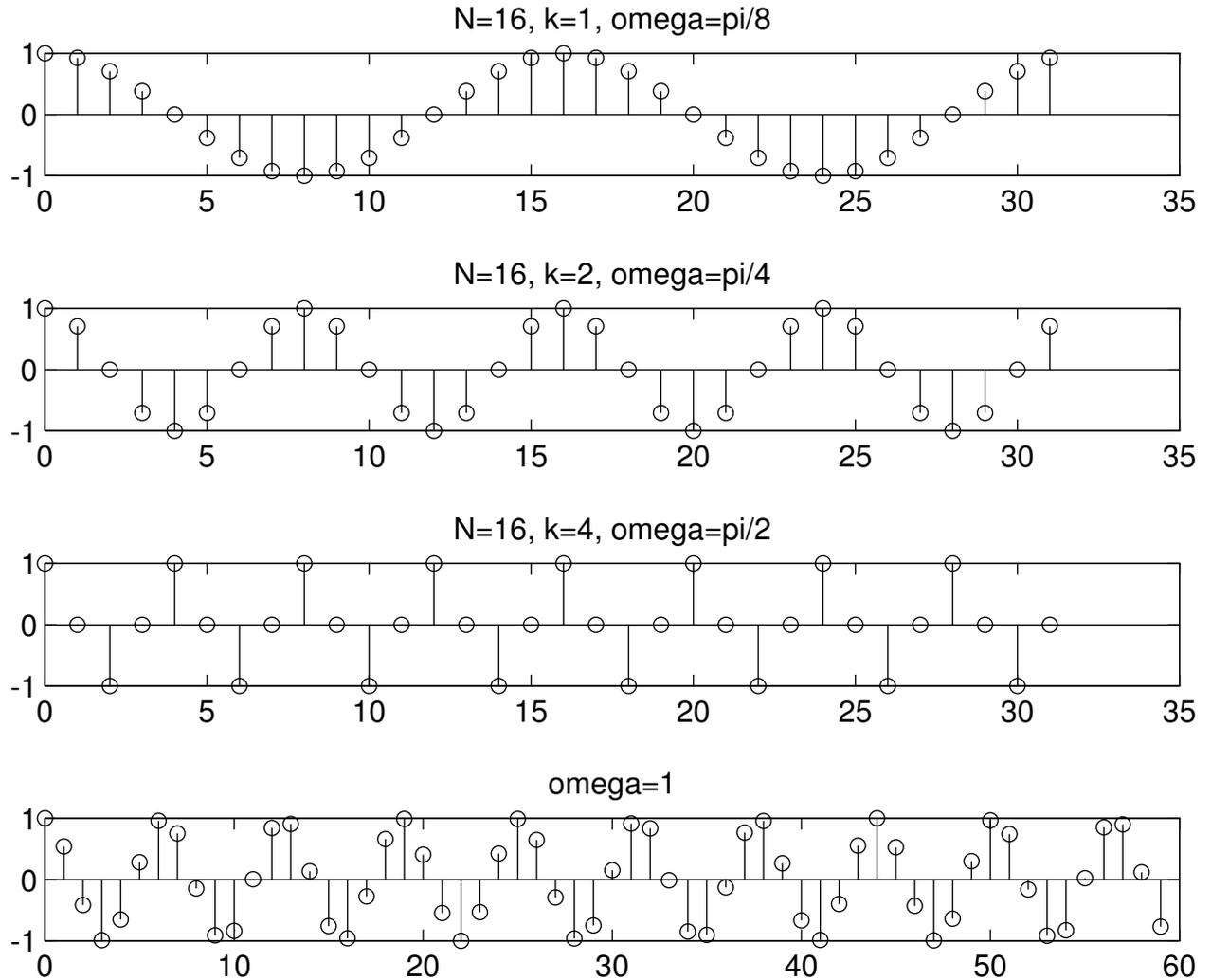
La relation $x(n) = x(n+N)$, $\forall n$, entraîne $\omega N = 2\pi k$, $k \in \mathbb{Z}$.

Suivant la fréquence ω on n'a pas nécessairement un signal périodique de période N .

Pour $\omega = 1$, par exemple, il n'existe pas d'entiers k et N tel que $N = 2\pi k$.

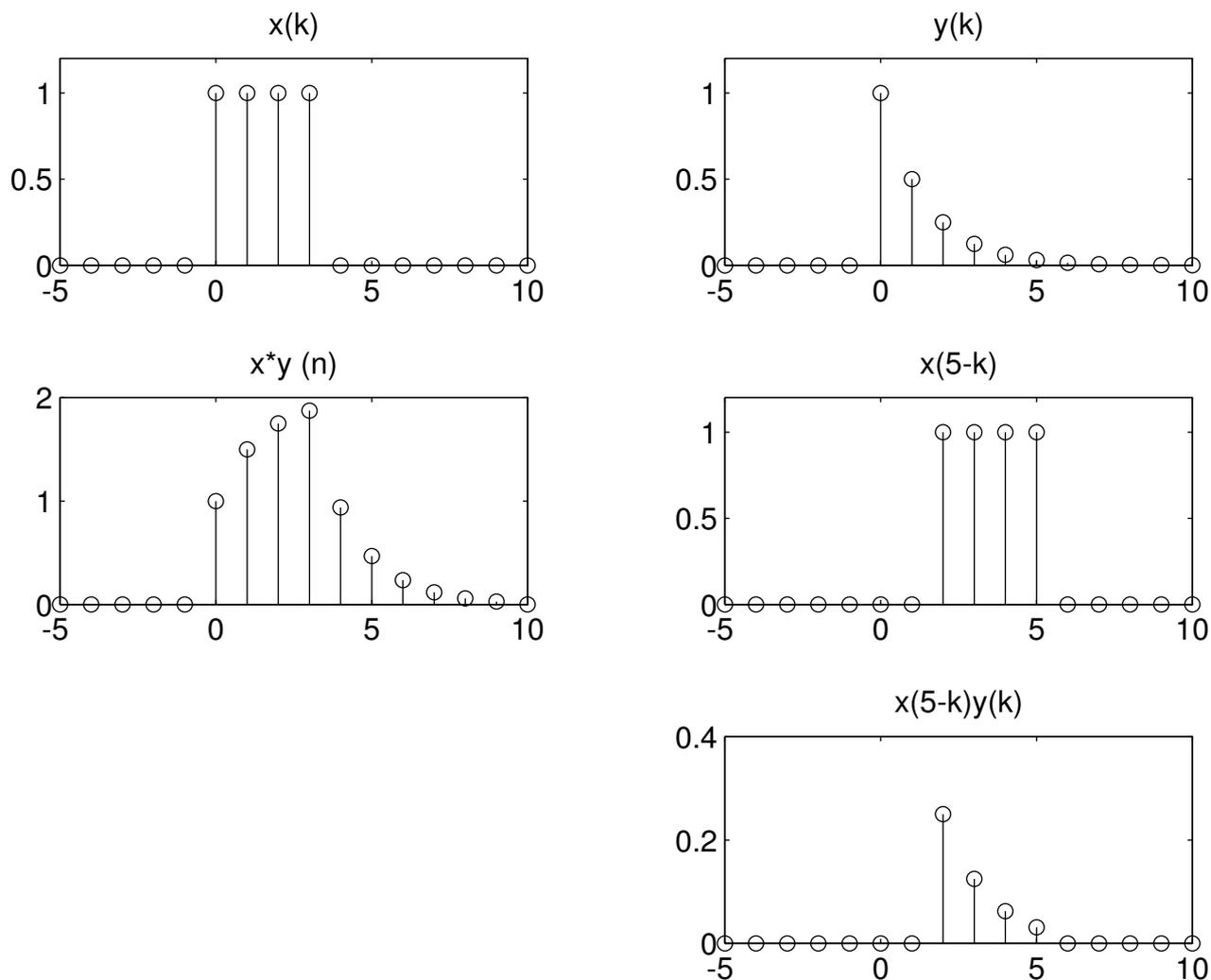
Par contre, pour une période N donnée, on trouve N fréquences distinctes, $\omega_k = 2\pi k/N, k = 0, 1, \dots, N-1$, telles que les signaux associés sont de période N et que l'on peut les distinguer sur la grille discrète.

- On remarque (voir figure) que le signal $x(n) = Ae^{i(\omega n + \phi)}$ oscille de plus en plus pour ω allant de 0 vers π et que les oscillations diminuent pour ω allant de π vers 2π . On dit qu'au voisinage de $2\pi k$ on a les «basses fréquences» et au voisinage de $\pi + 2\pi k$ on a les «hautes fréquences».



Sur la figure ci-dessus on a représenté la fonction $x(n) = \cos(\omega n), n \geq 0$, en prenant comme valeurs pour ω : $\pi/8, \pi/4, \pi/2$ et 1.

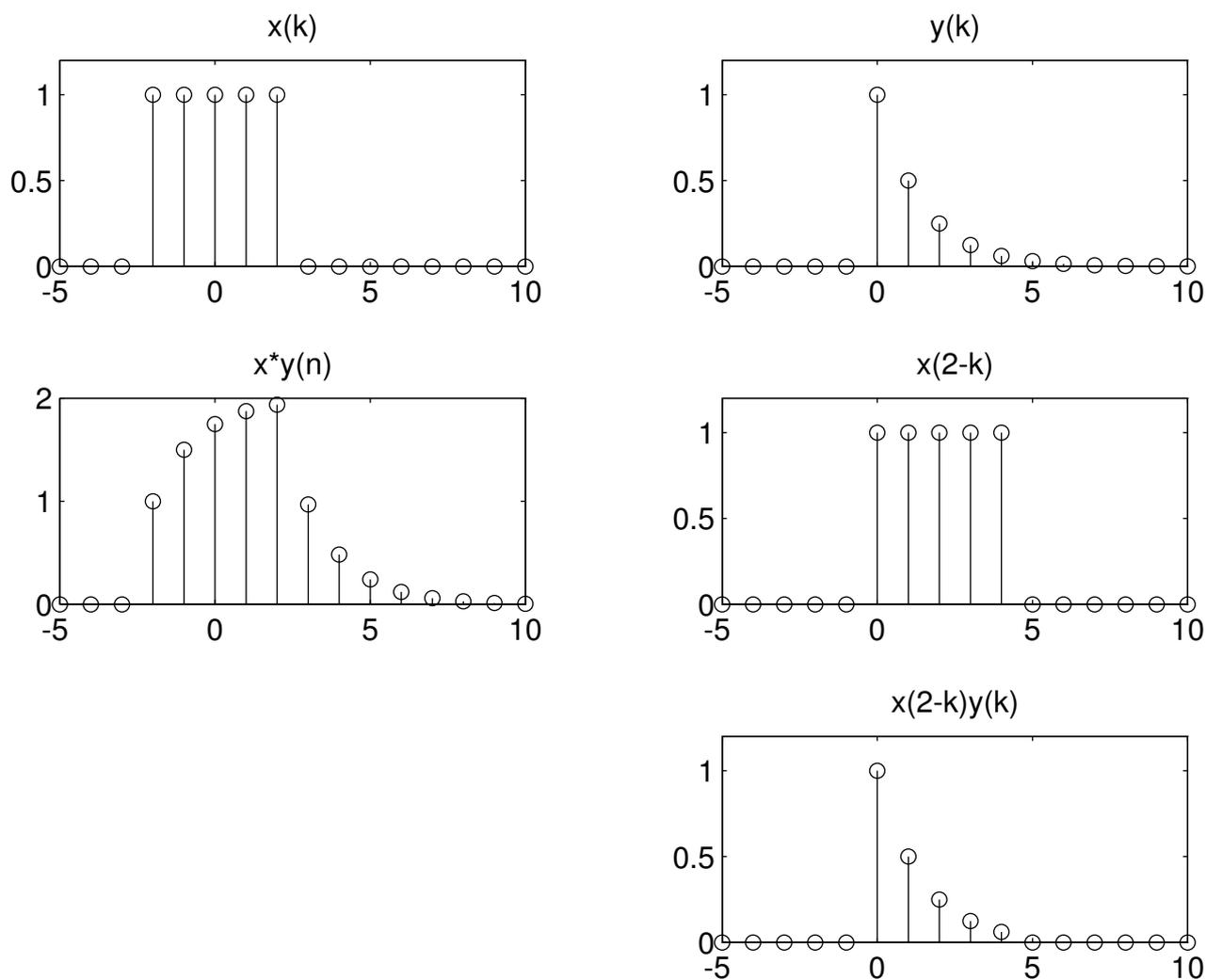
Dans les trois premiers cas on obtient des signaux discrets périodiques (de période 16). Remarquer aussi que le signal associé à $\omega = \pi/8$ est le même que celui associé à $\omega = 15\pi/8$. La quatrième figure représente un signal discret sinusoïdal qui n'a pas de période entière ($\omega = 1$).



Convolution de signaux discrets monodimensionnels :

$$x(k) = \text{rect}_4(k) , \quad y(k) = u(k)/2^k , \quad \text{où } u(k) \text{ est l'échelon unitaire.}$$

Remarque : On obtient le signal $x(n-k)$ en deux étapes. D'abord on construit $x(-k)$ par symétrie à partir de $x(k)$, ensuite on translate l'origine de $x(-k)$ en $k = n$.



Convolution de signaux discrets monodimensionnels :

$$x(k) = \text{rect}_5(k + 2), \quad y(k) = u(k)/2^k, \quad \text{où } u(k) \text{ est l'échelon unitaire.}$$

On va dès à présent introduire deux transformations de signaux, très importantes, et qui permettent de représenter un signal dans un autre «domaine» que celui du temps.

Transformée de Fourier : La transformation de Fourier d'un signal discret x , notée \hat{x} , est définie par :

$$\hat{x}(\omega) = \sum_{k \in \mathbb{Z}} x(k) e^{-ik\omega}, \quad 0 \leq \omega < 2\pi.$$

On associe à x une fonction périodique de période 2π à valeurs dans \mathbb{C} , en effet $\hat{x}(\omega) = \hat{x}(\omega + 2\pi)$. Donc tout intervalle de longueur 2π est suffisant pour décrire \hat{x} .

Pour un signal discret à support compact on obtient un polynôme trigonométrique.

Cette transformation est bien définie si $\sum_{k \in \mathbb{Z}} |x(k)| < \infty$ (c'est une condition suffisante!).

On a la transformation réciproque

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{x}(\omega) e^{in\omega} d\omega.$$

En général \hat{x} est une fonction complexe, on peut donc considérer sa partie réelle et sa partie imaginaire, ou encore l'exprimer comme suit

$$\hat{x}(\omega) = |\hat{x}(\omega)| e^{i \text{Arg}(\hat{x}(\omega))} = A(\omega) e^{i\Phi(\omega)}.$$

On dit que $A(\omega)$ est le *spectre d'amplitude*, il représente la répartition fréquentielle de l'amplitude du signal ; $\Phi(\omega)$ est le *spectre de phase*, il exprime la répartition fréquentielle de la phase du signal. On appelle *spectre d'énergie* la fonction $A(\omega)^2 = |\hat{x}(\omega)|^2$, elle exprime la répartition fréquentielle de l'énergie du signal.

L'intervalle des fréquences positives contenant les valeurs non nulles du spectre d'amplitude est appelé *largeur de bande* du signal.

Propriétés :

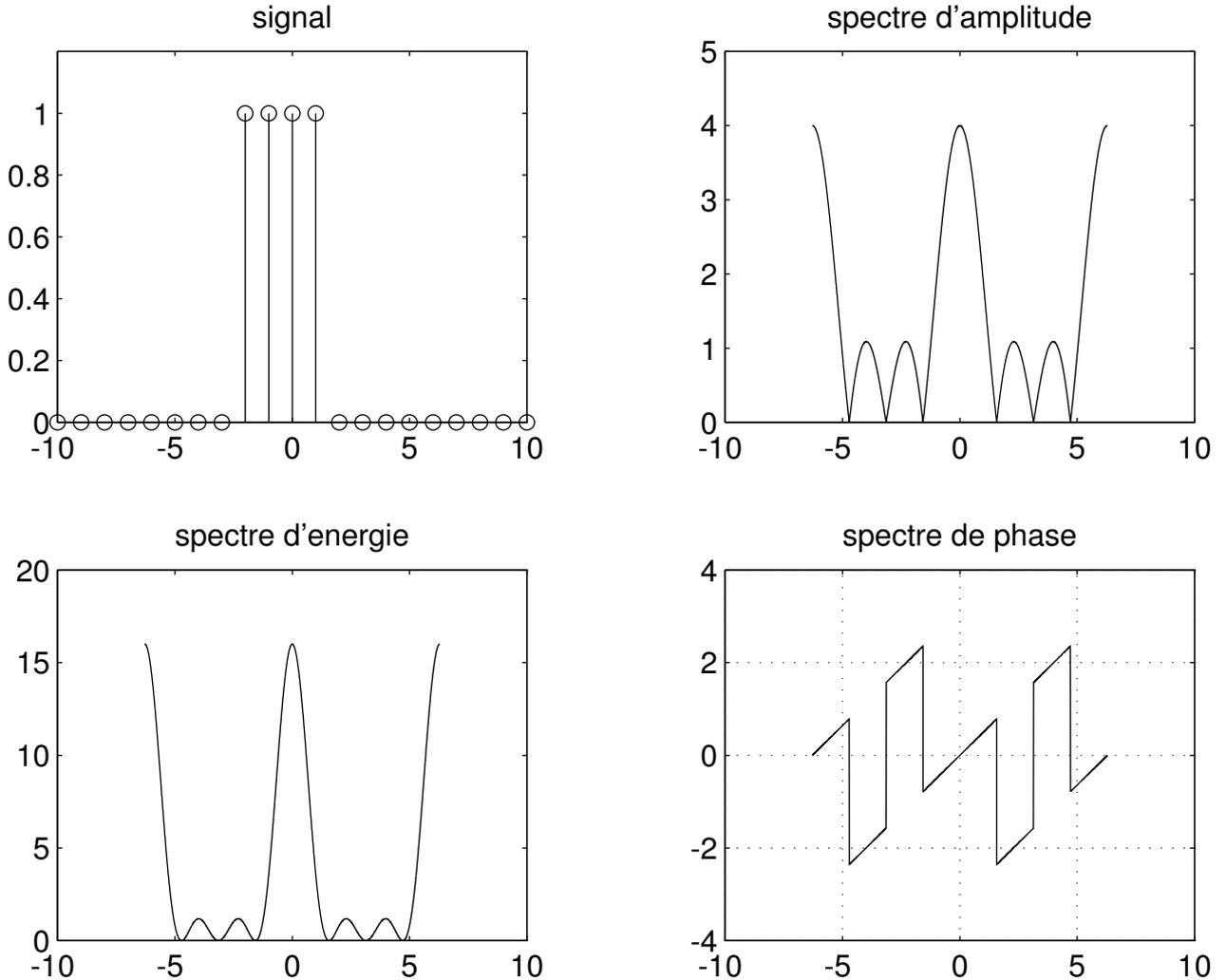
- La transformée de Fourier est linéaire.
- On a $\widehat{x \star y} = \hat{x} \cdot \hat{y}$.
- On a $\widehat{\tau_m x} = \hat{x} e^{-im\omega} = A(\omega) e^{i[\Phi(\omega) - m\omega]}$.
- Si $z = x \cdot y$, alors $\hat{z}(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{x}(\vartheta) \hat{y}(\omega - \vartheta) d\vartheta$.
- Si x est réel on a

$$\hat{x}(\omega) = \overline{\hat{x}(-\omega)}, \quad |\hat{x}(\omega)| = |\hat{x}(-\omega)| \quad \text{et} \quad \text{Arg}(\hat{x}(-\omega)) = -\text{Arg}(\hat{x}(\omega)).$$

Exemple :

On considère le signal $x(n) = \text{rect}_L(n + E(L/2))$, où E est la partie entière. Calculer \hat{x} .

Sur la figure suivante on a représenté $x(n)$, $|\hat{x}(\omega)|$, $\text{Arg}(\hat{x}(\omega))$ et $|\hat{x}(\omega)|^2$ pour $L = 4$ et $\omega \in [-2\pi, 2\pi]$.



Transformée en Z : La transformation en Z d'un signal discret x , notée X , est définie par :

$$X(z) = \sum_{k \in \mathbb{Z}} x(k) z^{-k} \quad z \in \mathbb{C}.$$

C'est une généralisation de la transformation de Fourier. La transformée en Z permet de considérer des fractions rationnelles à la place des suites. Avec un signal discret quelconque on obtient une série (de Laurent) dont il faut étudier le domaine de convergence. Pour un signal discret à support compact on obtient une fraction rationnelle en la variable complexe z .

La transformation réciproque suppose une intégration dans \mathbb{C} .

Propriétés :

- La transformée en Z est linéaire.
- Si $y = \tau_m x$, alors $Y(z) = z^{-m} X(z)$.
- Si $y(n) = a^n x(n)$, alors $Y(z) = X(z/a)$.
- Si $z = x * y$, alors $Z(z) = X(z) \cdot Y(z)$.

Exemples :

1. Soit $x(n) = \text{rect}_L(n + E(L/2))$, alors $X(z) = z^{-E(L/2)} \sum_{n=0}^{L-1} z^n$.
2. Soit $u(n)$, l'échantillon unitaire, alors $X(z) = \frac{z}{z-1}$, si $|z| > 1$.

Dans la section suivante on va généraliser aux signaux bidimensionnels les formules précédentes.

Signaux discrets bidimensionnels.

Un signal discret bidimensionnel est défini sur \mathbb{Z}^2 .

Une translation de $(\alpha, \beta) \in \mathbb{Z}^2$ du signal x , notée $\tau_{(\alpha, \beta)}$, est définie comme suit

$$\forall (i, j) \in \mathbb{Z}^2 : \tau_{(\alpha, \beta)} x(i, j) = x(i - \alpha, j - \beta).$$

De même le signal x sera périodique de période $(N, M) \in \mathbb{Z}^2$ si et seulement si il est périodique par rapport à chacune des variables, c'est-à-dire si et seulement si pour tout $(n, m) \in \mathbb{Z}^2$

$$\forall (i, j) \in \mathbb{Z}^2 : x(i, j) = x(i + n \cdot N, j + m \cdot M).$$

Le signal se répète dans \mathbb{Z}^2 suivant un rectangle de taille $N \times M$, par exemple le rectangle $[0, N-1] \times [0, M-1]$.

Les opérations de base se généralisent facilement, donnons les écritures en deux dimensions de la convolution (circulaire), de la TF et de la transformée en Z :

Convolution de deux signaux d.b. x et y , $z = x * y$,

$$\forall (i, j) \in \mathbb{Z}^2 : z(i, j) = \sum_{k, l \in \mathbb{Z}} x(i - k, j - l) \cdot y(k, l).$$

Si x a comme support un rectangle de taille $M_1 \times M_2$ et y un support de taille $N_1 \times N_2$, alors le support de z aura comme taille $(M_1 + N_1 - 1) \times (M_2 + N_2 - 1)$.

Convolution circulaire de deux signaux d.b. périodiques de période (N, M) , $z = x \star_c y$,

$$\forall (i, j) \in \mathbb{Z}^2 : z(i, j) = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} x(i - k, j - l) \cdot y(k, l).$$

Transformée de Fourier d'un signal d.b. x ,

$$\hat{x}(\omega_1, \omega_2) = \sum_{k, l \in \mathbb{Z}} x(k, l) e^{-ik\omega_1} e^{-il\omega_2}, \quad 0 \leq \omega_1 < 2\pi, 0 \leq \omega_2 < 2\pi.$$

On associe à x une fonction périodique de période $(2\pi, 2\pi)$ à valeurs dans \mathbb{C} , la réciproque s'écrit

$$x(n, m) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{2\pi} \hat{x}(\omega_1, \omega_2) e^{in\omega_1} e^{im\omega_2} d\omega_1 d\omega_2.$$

Les variables k et l représentent en général des distances, pour les variables ω_1 et ω_2 on parle dans le cas bidimensionnel de *fréquences spatiales*.

Transformée en Z d'un signal d.b. x :

$$X(z_1, z_2) = \sum_{k, l \in \mathbb{Z}} x(k, l) z^{-k} z^{-l} \quad z_1, z_2 \in \mathbb{C}.$$

Remarque : Les mêmes restrictions de validité que dans le cas 1D s'appliquent à ces formules.

Soient x_1 et x_2 deux signaux d.m., on obtient un signal discret bidimensionnel $x = x_1 \otimes x_2$ comme suit

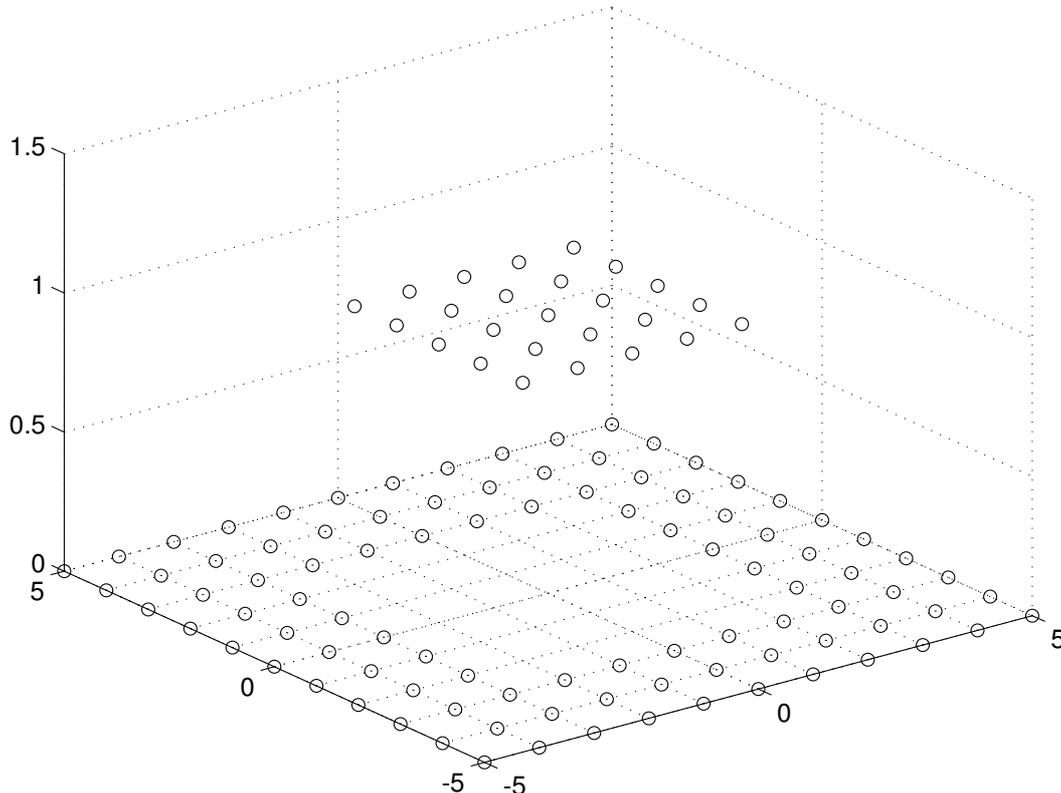
$$\forall (i, j) \in \mathbb{Z}^2 : x(i, j) = x_1(i) \cdot x_2(j).$$

Réciproquement, on dit qu'un signal bidimensionnel z est *séparable* si et seulement si il existe deux signaux monodimensionnels x_1 et x_2 tels que

$$\forall (i, j) \in \mathbb{Z}^2 x(i, j) = x_1(i) \cdot x_2(j).$$

Exemples : $\delta(k, l)$, $u(k, l)$ et $rect_{M,N}(k, l)$.

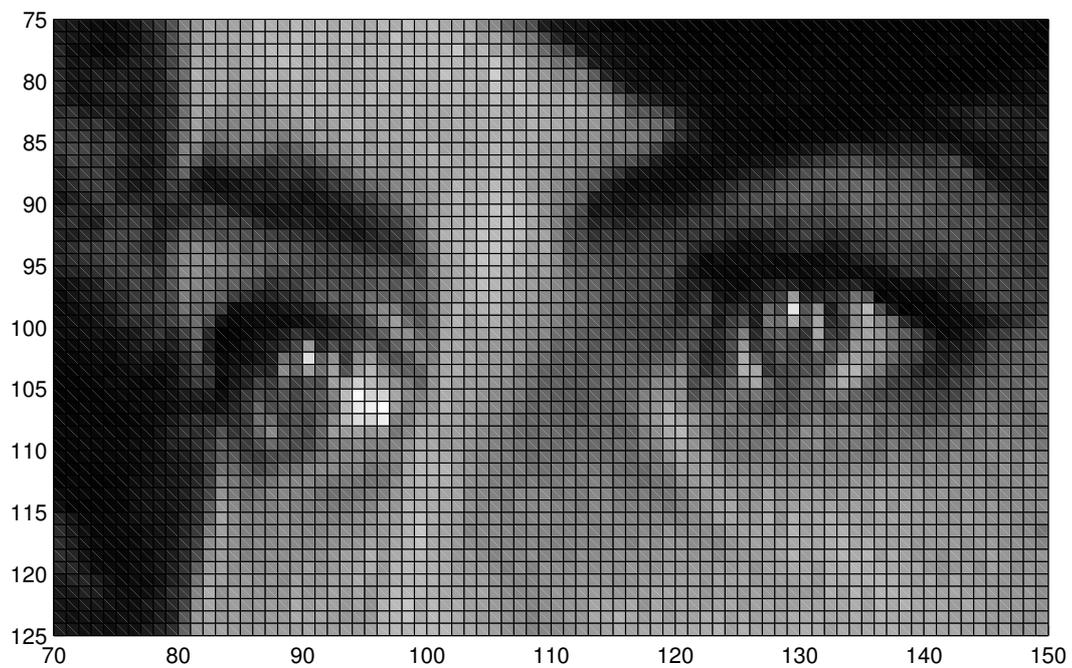
Sur la figure suivante on a représenté le signal bidimensionnel séparable $x(k, l) = rect_{5,5}(k+2, l+2) = rect_5(k+2) \otimes rect_5(l+2)$.



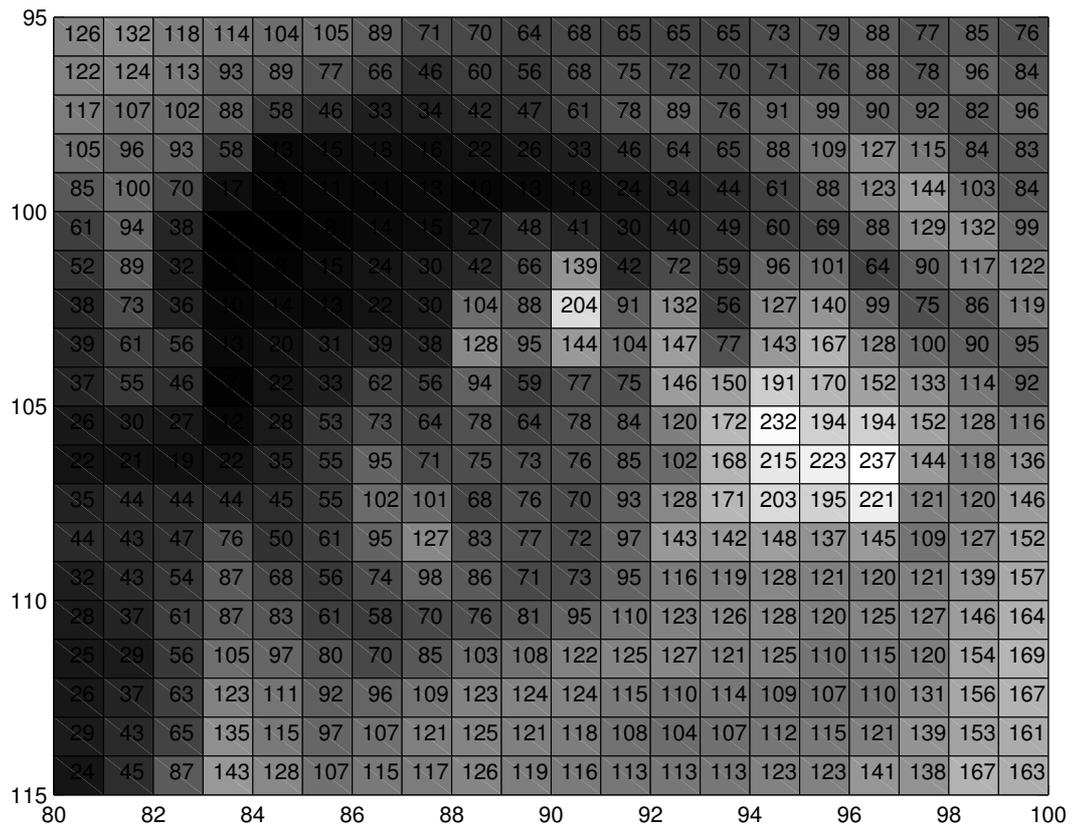
Un signal bidimensionnel plus «réaliste» est représenté sur la figure suivante :



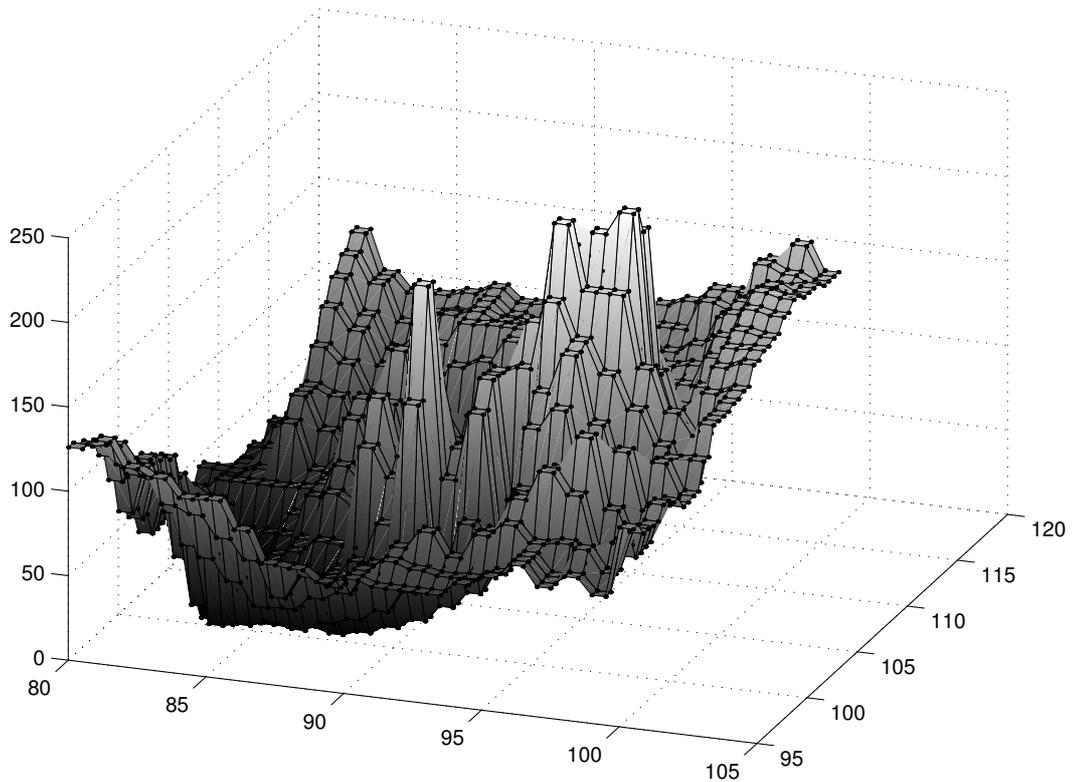
Si l'on regarde de plus près :



Chaque pixel $x(i, j)$ est représenté par un nombre interprété comme niveau de gris :



Ou on considère la valeur de $z_{ij} = x(i, j)$ comme «altitude» en trois dimensions :



2.2 Systèmes discrets.

Définition : Un *système discret* \mathcal{H} , qui à un signal numérique x en entrée fait correspondre un signal numérique y , est défini par :

$$y = \mathcal{H}[x].$$

Noter que la valeur de y en n , $y(n)$, peut dépendre de toutes les valeurs $x(k)$, $k \in \mathbb{Z}$.

Exemples :

1. Le système *retard* défini par : $y(n) = x(n - r)$, $\forall n \in \mathbb{Z}$. Le retard r est un entier positif (sinon on obtient un système en avance).
2. Le système *moyenne glissante* défini par :

$$\forall n \in \mathbb{Z}, \quad y(n) = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x(n - k).$$

Ce système calcule la valeur de $y(n)$ en prenant la moyenne des $M_1 + M_2 + 1$ valeurs de x autour de n .

On caractérise les systèmes grâce à leurs propriétés.

- * Systèmes *sans mémoire* : pour tout $n \in \mathbb{Z}$, $y(n)$ ne dépend que de $x(n)$.
Par exemple $y(n) = (x(n))^2$, $\forall n \in \mathbb{Z}$, définit un système sans mémoire ; par contre le système retard a de la mémoire si $r \in \mathbb{Z}^*$.
- * Un système \mathcal{H} est *linéaire* si le principe de superposition est vérifié, c'est-à-dire pour deux signaux quelconques x_1 et x_2 , deux constantes réelles (ou complexes) quelconques a et b , on a :

$$\mathcal{H}[ax_1 + bx_2] = a\mathcal{H}[x_1] + b\mathcal{H}[x_2],$$

Le système *accumulateur*, $y(n) = \sum_{k=-\infty}^n x(k)$, est linéaire.

- * Un système \mathcal{H} est dit *invariant* ou *stationnaire* si un décalage (*i.e.* translation) de l'entrée résulte en un décalage identique de la sortie,

$$y = \mathcal{H}[x] \Rightarrow \tau_a y = \mathcal{H}[\tau_a x].$$

Par exemple $y(n) = x(Mn)$, (M entier positif), n'est pas invariant.

- * Un système est *causal* si au temps n_0 , la valeur de y , *i.e.* $y(n_0)$, ne dépend que des valeurs de x pour des temps $n \leq n_0$. Soient $y_1 = \mathcal{H}[x_1]$ et $y_2 = \mathcal{H}[x_2]$,

$$\text{si } x_1(n) = x_2(n) \text{ pour } n \leq n_0, \text{ alors } y_1(n) = y_2(n) \text{ pour } n \leq n_0.$$

On dit encore que le système n'anticipe pas.

Le système $y(n) = x(n+1) - x(n)$ n'est pas causal ; le système $y(n) = x(n) - x(n-1)$ est causal.

- * Un système est dit *stable* si à une entrée bornée il fait correspondre une sortie bornée (BIBO en anglais). Soit $y = \mathcal{H}[x]$,

$$\text{si } \exists A \in \mathbb{R}_+ : |x(n)| \leq A \quad \forall n \in \mathbb{Z}, \text{ alors } \exists B \in \mathbb{R}_+ : |y(n)| \leq B \quad \forall n \in \mathbb{Z}.$$

Par exemple le système accumulateur n'est pas stable, mais le système moyenne glissante et le système retard le sont.

Remarque : Ces propriétés sont des propriétés des systèmes, et non pas des entrées, c'est-à-dire que l'on doit pouvoir vérifier la propriété en question pour chaque entrée, réciproquement si l'on trouve un seul signal entrée pour lequel la propriété n'est pas vérifiée, le système ne possède pas la propriété.

Dans la suite on va principalement s'intéresser à une classe particulière de systèmes discrets, ceux qui sont linéaires et invariants.

2.2.1 Systèmes linéaires invariants.

Définition : Un système linéaire et invariant est appelé *filtre*.

Théorème des filtres : Soit \mathcal{H} un filtre, alors si y est la réponse à x , $y = \mathcal{H}[x]$, il existe une suite $(h(n))_{n \in \mathbb{Z}}$ telle que

$$\forall n \in \mathbb{Z} : y(n) = \sum_{k \in \mathbb{Z}} h(n-k)x(k),$$

c'est-à-dire

$$y = h * x.$$

On dit que h est la *réponse impulsionnelle* de \mathcal{H} : pour une entrée égale à δ , on obtient comme réponse du système le signal h .

La réponse impulsionnelle caractérise complètement un système linéaire invariant.

Définition : On distingue deux classes de filtres :

- les filtres à *réponse impulsionnelle finie* (RIF, resp. FIR en anglais), pour lesquels le support de h est fini ;
- les filtres à *réponse impulsionnelle infinie* (RII, resp. IIR en anglais), pour lesquels le support de h est infini.

Exemples : Moyenne glissante, accumulateur, etc...

Grâce à la modélisation particulièrement simple des systèmes linéaires invariants, on peut immédiatement énoncer plusieurs propriétés :

- L'opération de convolution est commutative, l'entrée x et la réponse impulsionnelle h peuvent donc échanger leurs rôles.
- Filtres en *chaîne* : soient h_1 et h_2 les r.i. de deux filtres en chaîne, grâce à l'associativité on a

$$y = h_2 * (h_1 * x) = (h_2 * h_1) * x.$$

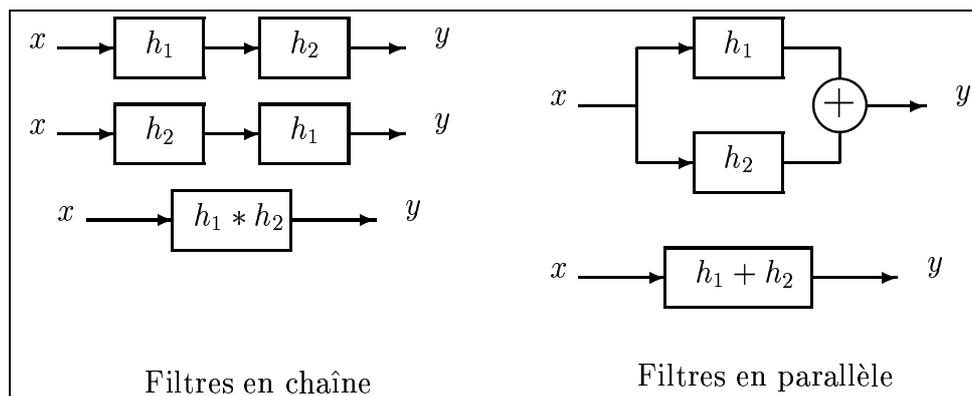
Deux filtres en chaîne se comportent comme un filtre de r.i. $h = h_1 * h_2$, l'ordre des filtres étant quelconque.

- Filtres en *parallèle* : soient h_1 et h_2 les r.i. de deux filtres en parallèle, grâce à la distributivité on a,

$$y = (h_1 * x) + (h_2 * x) = (h_1 + h_2) * x.$$

Deux filtres en parallèle se comportent comme un filtre de r.i. $h = h_1 + h_2$.

- Un filtre est stable si et seulement si $\sum_{k \in \mathbb{Z}} |h(k)| < \infty$.
- Un filtre est causal si et seulement si $h(n) = 0$, pour tout $n < 0$.

Exemples :

1. On considère les filtres $\mathcal{H}_1 : y(n) = x(n+1) - x(n)$, et $\mathcal{H}_2 : y(n) = x(n-1)$. Déterminer le filtre \mathcal{H} obtenu par la cascade $\mathcal{H}_1 \mathcal{H}_2$.
2. Même question pour le filtre accumulateur et le filtre \mathcal{H} .

Définition : Un filtre de r.i. h a un *filtre inverse*, de r.i. h_i , si on a :

$$h * h_i = h_i * h = \delta.$$

On parle encore de *déconvolution*.

Utilisation de la transformée de Fourier : étude fréquentielle d'un filtre.

Soit \mathcal{H} un filtre de r.i. h on a donc

$$y = h * x,$$

grâce aux propriétés de la transformation de Fourier on a

$$\hat{y} = \hat{h} \cdot \hat{x}.$$

La transformée de Fourier de la réponse impulsionnelle est appelé *réponse fréquentielle*.

Elle modifie le spectre d'amplitude et le spectre de phase de l'entrée suivant la forme de la fonction \hat{h} , le système agit sur la répartition fréquentielle de l'entrée.

Remarques :

1. On considère le signal complexe $x(n) = e^{i\omega_0 n}$, $n \in \mathbb{Z}$, avec $\omega_0 \in \mathbb{R}$, comme entrée d'un filtre de r.i. h .
La réponse du filtre est $y(n) = \hat{h}(\omega_0) \cdot x(n)$.
2. La fonction $\hat{h}(\omega)$ est 2π périodique, on se restreint d'habitude à l'intervalle $[-\pi, \pi]$. On parle de «basses fréquences» au voisinage de 0 ($+2k\pi$) et de «hautes fréquences» au voisinage de π ($+2k\pi$).
3. Le problème de la déconvolution s'exprime de façon simple grâce à la réponse fréquentielle. Pour trouver le filtre inverse au filtre de r.i. h , il faut déterminer h_i telle que $\hat{h} \cdot \hat{h}_i = 1$, c'est-à-dire $\hat{h}_i = 1/\hat{h}$.

Exemple :

Calculer la réponse $y(n)$, d'un filtre de r.i. réelle h , au signal $x(n) = A \cos(\omega_0 n + \phi)$.
Vérifier le résultat pour le filtre retard idéal.

Filtres idéaux.

Ce sont des filtres qui permettent de modifier le spectre d'amplitude d'un signal.

– Le *filtre passe-bas* dont la réponse fréquentielle est définie par

$$\hat{h}_{pb}(\omega) = \begin{cases} 1 & , \text{ si } |\omega| \leq \omega_c \\ 0 & , \text{ si } \omega_c < |\omega| \leq \pi \end{cases} .$$

Sa réponse impulsionnelle est $h_{pb}(n) = \frac{\sin(\omega_c n)}{\pi n}$.

– Le *filtre passe-haut* dont la réponse fréquentielle est définie par

$$\hat{h}_{ph}(\omega) = \begin{cases} 0 & , \text{ si } |\omega| \leq \omega_c \\ 1 & , \text{ si } \omega_c < |\omega| \leq \pi \end{cases} .$$

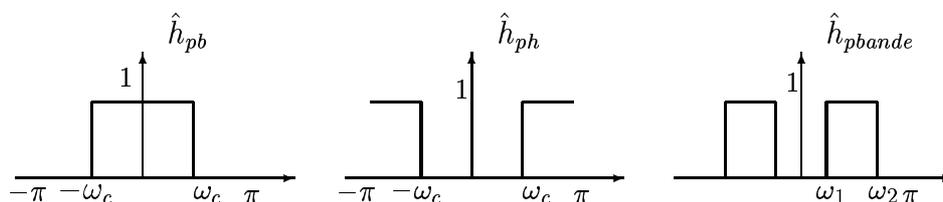
Sa réponse impulsionnelle est $h_{ph}(n) = \delta(n) - \frac{\sin(\omega_c n)}{\pi n}$.

– Le *filtre passe-bande* dont la réponse fréquentielle est définie par

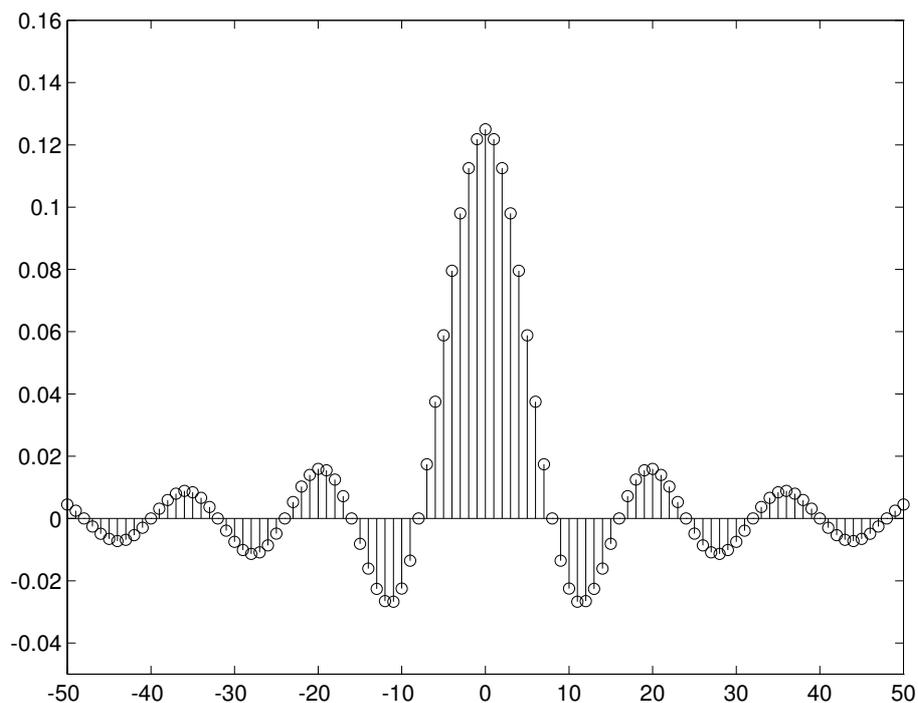
$$\hat{h}_{pbande}(\omega) = \begin{cases} 1 & , \text{ si } \omega_1 \leq |\omega| \leq \omega_2 \\ 0 & , \text{ si } |\omega| < \omega_1 \text{ ou } \omega_2 < |\omega| \leq \pi \end{cases} .$$

– Le *filtre coupe-bande* de réponse fréquentielle $\hat{h}_{cbande}(\omega) = 1 - \hat{h}_{pbande}(\omega)$.

Les fréquences de coupure ω_c , ω_1 et ω_2 , sont des réels positifs.



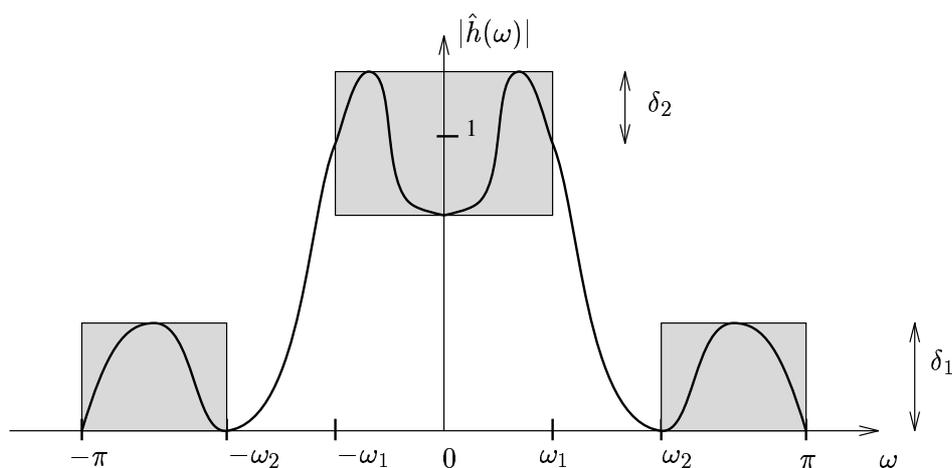
On constate que le support d'un filtre idéal s'étend de $-\infty$ à $+\infty$, ces filtres ne sont donc pas réalisables en pratique. Sur la figure suivante on a représenté la réponse impulsionnelle d'un filtre passe-bas, de fréquence de coupure $\omega_c = \pi/8$.



Pour des réalisations pratiques de ces filtres on est amené à considérer des filtres à phase non nulle et de donner des spécifications avec des tolérances sur l'amplitude de la réponse fréquentielle.

La phase est généralement spécifiée par des contraintes de causalité et de stabilité. Les contraintes sur l'amplitude sont données par ce que l'on appelle un *gabarit*.

La figure en dessous représente un gabarit et une réponse fréquentielle qui satisfait aux spécifications. La bande passante va de 0 à ω_1 et on y permet une erreur de δ_2 autour de l'amplitude idéale 1. Dans la région où l'on veut une atténuation des fréquences, on permet une erreur de δ_1 . On a une zone de transition entre ω_1 et ω_2 , où la réponse fréquentielle passe de la bande passante à la bande coupée.



Utilisation de la transformée en Z.

Soit \mathcal{H} un filtre de r.i. h on a donc

$$y = h * x,$$

en utilisant les propriétés de la transformée en Z on a

$$Y(z) = H(z) \cdot X(z).$$

La transformée en Z de la réponse impulsionnelle d'un système est appelée *fonction de transfert*.

Exemples :

1. Le filtre retard unité $y(n) = x(n-1)$ a comme réponse impulsionnelle $h(n) = \delta(n-1)$ et comme fonction de transfert $H(z) = z^{-1}$.
2. Le filtre accumulateur a comme réponse impulsionnelle $h(n) = u(n)$ et comme fonction de transfert $H(z) = \frac{1}{1-z^{-1}}$, pour $|z| > 1$.

La fonction de transfert est très utile pour étudier la causalité et la stabilité d'un filtre. De plus, elle sert à faire le lien entre la réponse impulsionnelle et les équations aux différences, présentées dans la section suivante.

Équations aux différences.

Un grand nombre de filtres sont définies par une équation aux différences

$$\sum_{k=0}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k). \quad (*)$$

Les coefficients $(a_k)_{k=0,\dots,N}$ et $(b_k)_{k=0,\dots,M}$ sont des constantes complexes.

Exemples :

1. Le système accumulateur, dont la r.i. u est à support infini, vérifie l'équation aux différences

$$y(n) = y(n-1) + x(n).$$

On peut donc calculer $y(n)$ en utilisant un nombre fini d'échantillons du signal en entrée $(x(n))_{n \in \mathbb{Z}}$.

2. Soit le système de r.i.

$$h(n) = \frac{1}{L} \sum_{k=0}^{L-1} \delta(n-k).$$

C'est le filtre moyenne glissante causal, il vérifie les équations aux différences

$$y(n) = \frac{1}{L} \sum_{k=0}^{L-1} x(n-k)$$

et

$$y(n) = y(n-1) + \frac{1}{L}(x(n) - x(n-L)).$$

On constate que pour un filtre donné on peut trouver plusieurs équations aux différences.

Soient $(x_p(n))_{n \in \mathbb{Z}}$ une entrée et $(y_p(n))_{n \in \mathbb{Z}}$ une réponse, solution particulière de l'équation (*). Soit de plus $(y_h(n))_{n \in \mathbb{Z}}$ vérifiant l'équation *homogène* $\sum_{k=0}^N a_k y(n-k) = 0$.

Si l'on pose $y(n) = y_h(n) + y_p(n)$, l'entrée $(x_p(n))_{n \in \mathbb{Z}}$ et la réponse $(y(n))_{n \in \mathbb{Z}}$ vérifient aussi l'équation aux différences (*).

On n'a donc pas une solution unique de l'équation (*).

Pour avoir unicité de la sortie y , il suffit d'avoir N *conditions initiales*; si on fixe $y(-1)$, $y(-2)$, \dots , $y(-N)$, par exemple, on obtient, en réarrangeant l'équation (*):

$$y(n) = \sum_{k=0}^M \frac{b_k}{a_0} x(n-k) - \sum_{k=1}^N \frac{a_k}{a_0} y(n-k) \quad , \text{ pour } n = 0, 1, 2, \dots$$

$$y(n) = \sum_{k=0}^M \frac{b_k}{a_N} x(n-k) - \sum_{k=0}^{N-1} \frac{a_k}{a_N} y(n-k) \quad , \text{ pour } n = -N-1, -N-2, \dots$$

Ceci permet de calculer les valeurs de y pour tout n , il faut néanmoins garantir la cohérence des conditions initiales avec les contraintes de linéarité et de stationnarité.

Remarque : Si l'on suppose que le système démarre du repos, c'est-à-dire si pour $x(n) = 0$ pour $n < n_0$, on impose $y(n) = 0$ pour $n < n_0$, alors on montre que l'équation (*) détermine de façon unique un système linéaire, invariant et causal.

Si $N = 0$, l'équation (*) s'écrit

$$y(n) = \sum_{k=0}^M \frac{b_k}{a_0} x(n-k) ,$$

cette équation, non récursive, détermine un filtre RIF :

$$h(n) = \sum_{k=0}^M \frac{b_k}{a_0} \delta(n-k) .$$

Plus généralement : comment trouver la réponse impulsionnelle associée à l'équation aux différences (*) ?

Supposons que l'équation (*) est vérifiée pour tout entier $n \in \mathbb{Z}$ et, grâce aux propriétés de la transformée en Z, on a

$$\left(\sum_{k=0}^N a_k z^{-k} \right) Y(z) = \left(\sum_{k=0}^M b_k z^{-k} \right) X(z) ,$$

Si on a un filtre, on sait que

$$Y(z) = H(z) \cdot X(z)$$

et ce filtre est entièrement déterminé par la fonction de transfert $H(z)$ (en n'oubliant pas le domaine de convergence de H). On trouve alors

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} .$$

Note : Pour une expression algébrique donnée de $H(z)$, on peut trouver différentes réponses impulsionnelles, en fonction du choix du domaine de convergence.

Réciproquement, sous quelles conditions et de quelle façon, peut-on associer à une réponse impulsionnelle donnée une équation aux différences ?

Soit un filtre de r.i. h , à partir de $y = h * x$ on trouve $Y(z) = H(z) \cdot X(z)$.

Seulement si l'on peut écrire H sous forme d'une fraction rationnelle $H(z) = \frac{P(z^{-1})}{Q(z^{-1})}$, où P et Q sont des polynômes, on trouve une équation aux différences.

Exemples :

1. Soit le filtre causal de réponse impulsionnelle $h(n) = a^n u(n)$, pour $n \in \mathbb{N}$.

On montre que sa fonction de transfert est $H(z) = \frac{1}{1 - az^{-1}}$, avec $|z| > |a|$.

Une équation aux différences est $y(n) = ay(n-1) + x(n)$.

2. Soit le filtre anticausal de réponse impulsionnelle $h(n) = -a^n u(-n-1)$, $n \in \mathbb{N}$.

Sa fonction de transfert est $H(z) = \frac{1}{1 - az^{-1}}$, pour $|z| < |a|$.

On obtient donc la même fonction de transfert que dans l'exemple précédent, mais avec un domaine de convergence différent. De même on aura la même équation aux différences, mais on aura d'autres conditions initiales pour le filtre anticausal que pour le filtre causal.

3. Soit le filtre de réponse impulsionnelle finie, $h(n) = \begin{cases} a^n & , \text{ si } 0 \leq n \leq M-1 \\ 0 & , \text{ ailleurs} \end{cases}$.

Alors la fonction de transfert est $H(z) = \sum_{n=0}^{M-1} a^n z^{-n} = \frac{1 - a^M z^{-M}}{1 - az^{-1}} = \frac{1}{z^{M-1}} \frac{z^M - a^M}{z - a}$,

avec $z \neq 0$.

On a les équations aux différences

$$y(n) = \sum_{k=0}^{M-1} a^k x(n-k)$$

et

$$y(n) = ay(n-1) + x(n) - a^n x(n-M).$$

Résumé sur les filtres :

Soit \mathcal{H} un filtre, il est entièrement caractérisé par sa réponse impulsionnelle h et

$$y(n) = \sum_{k \in \mathbb{Z}} h(n-k)x(k).$$

Souvent on caractérise un filtre aussi par sa réponse fréquentielle $\hat{h}(\omega)$ et

$$\hat{y}(\omega) = \hat{h}(\omega) \cdot \hat{x}(\omega),$$

où encore par sa fonction de transfert $H(z)$ et

$$Y(z) = H(z) \cdot X(z).$$

Si $H(z)$ s'écrit sous la forme d'une fraction rationnelle en z^{-1} , on obtient une équation aux différences

$$\sum_{k=0}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k).$$

On a ainsi regroupé les quatre principaux outils de l'étude des filtres et on peut résumer de la façon suivante :

- * Les filtres RIF sont réalisables de façon récursive et non récursive.
- * Les filtres RII sont soit réalisables de façon récursive, soit pas réalisables.

Note : On verra plus loin un moyen de calcul du produit de convolution pour un filtre RIF, la transformée de Fourier rapide (TFR), qui est plus rapide que le calcul direct non récursif.

2.2.2 Filtres discrets bidimensionnels.

Dans cette section on s'intéresse particulièrement aux systèmes linéaires et invariants par translation en dimension deux.

Rappelons d'abord que la variable $(i, j) \in \mathbb{Z}$ représente en général des coordonnées spatiales (*e.g.* coordonnées d'un pixel). Les opérations sur les signaux bidimensionnels ont été présentées dans la section 2.1 .

La linéarité et l'invariance par translation d'un système se généralisent facilement au cas de dimension deux et on obtient un énoncé du théorème des filtres équivalent à celui vu pour le cas monodimensionnel.

Un système linéaire et invariant par translation est entièrement caractérisé par sa réponse impulsionnelle $(h(i, j)_{i, j \in \mathbb{Z}})$ et on a

$$y(n, m) = \sum_{k, l \in \mathbb{Z}} h(n-k, m-l)x(k, l).$$

Remarques :

- L'invariance par translation impose au système d'agir sur un signal, par exemple une image, indépendamment de la position absolue du signal dans le plan.
- La linéarité implique qu'un filtre, appliqué à la somme de deux images, donne en résultat la somme des résultats obtenus pour les images filtrées indépendamment.

Or la propriété de linéarité, qui simplifie énormément la modélisation mathématique, n'est pas nécessairement souhaitable en traitement d'image. En effet, la somme de deux images (au sens addition de nombres) donne une nouvelle image où les objets, présents dans les images initiales, ne sont plus reconnaissables. Une bonne définition de «somme» de deux images tiendra compte des occultations et transparences que l'on obtient en superposant deux «scènes».

- Un système linéaire et invariant par translations n'est pas nécessairement stable pour les rotations ou les projections. Donc des images d'un même objet, prises de divers points de vue, seront traitées de façon différente (*e.g.* images aériennes).

- En dimension deux, la convolution d'un signal avec le filtre $h(k, l)$, au point (n, m) , s'obtient comme suit : on effectue une symétrie autour de $(0, 0)$ pour obtenir $h(-k, -l)$, suivie d'une translation de (n, m) , ce qui donne $h(-(k-n), -(l-m)) = h(n-k, m-l)$.
- Si h est un filtre de taille $M_1 \times M_2$, c'est-à-dire que l'on a M_1 colonnes et M_2 lignes, que l'on applique à un signal (image) de taille $NC \times NL$, on obtient un signal résultat de taille $(NC + M_1 - 1)(NL + M_2 - 1)$.

Filtres séparables.

Des filtres particulièrement simples à utiliser sont les *filtres séparables*, obtenus en multipliant deux réponses impulsionnelles monodimensionnelles h_1 et h_2 .

Soit $h(i, j) = h_1(i) \cdot h_2(j)$ pour tout $(i, j) \in \mathbb{Z}^2$, alors la sortie de ce filtre s'obtient en faisant deux convolutions en dimension un :

$$\begin{aligned}
 y(n, m) &= \sum_{k, l \in \mathbb{Z}} h(n-k, m-l) x(k, l) \\
 &= \sum_{k, l \in \mathbb{Z}} h_1(n-k) h_2(m-l) x(k, l) \\
 &= \sum_{k \in \mathbb{Z}} h_1(n-k) \left(\sum_{l \in \mathbb{Z}} h_2(m-l) x(k, l) \right) \\
 &= \sum_{k \in \mathbb{Z}} h_1(n-k) y_2(k, m).
 \end{aligned}$$

Considérons une image $(x(k, l))_{0 \leq k, l \leq N}$ et les filtres monodimensionnels h_1 , de taille M_1 , et h_2 , de taille M_2 . Le filtre bidimensionnel $h(i, j) = h_1(i) \cdot h_2(j)$ est donc de taille $M_1 \cdot M_2$, M_1 colonnes et M_2 lignes.

On veut calculer $y(n, m)$, pour $0 \leq n, m \leq N$ uniquement. On néglige ainsi les valeurs de y en dehors du support de x , en pratique on suppose que y est nul en dehors du support de x .

Si l'on calcule y avec la formule générale, *i.e.* en utilisant h , on a de l'ordre de $M_1 \cdot M_2$ opérations par point, donc de l'ordre de $N^2 M_1 M_2$ opérations en tout.

Par contre, si l'on détermine d'abord $y_2(k, m)$ ($0 \leq k, m \leq N$), grâce à la convolution «par lignes» h_2 , on effectue de l'ordre de $N^2 M_2$ opérations. Enfin, on obtient y à partir de y_2 , grâce à la convolution «par colonnes» h_1 , ce qui nécessite de l'ordre de $N^2 M_1$ opérations. En tout l'on aura exécuté de l'ordre de $N^2(M_1 + M_2)$ opérations pour obtenir y à partir de x .

En particulier si $N = M_1 = M_2$, on a de l'ordre de N^3 opérations dans le cas séparable, au lieu de N^4 dans le cas général.

Donc les filtres séparables sont plus faciles à implémenter et plus rapides que les filtres bidimensionnels non séparables.

Formulaire pour les filtres linéaires bidimensionnels.

On a vu la représentation d'un filtre par sa réponse impulsionnelle $(h(k, l))_{k, l \in \mathbb{Z}}$, comme dans le cas monodimensionnel, on peut aussi considérer la réponse fréquentielle du filtre

$$\hat{h}(\omega_1, \omega_2) = \sum_{k, l \in \mathbb{Z}} h(k, l) e^{-ik\omega_1} e^{-il\omega_2}, \quad -\pi \leq \omega_1 < \pi, \quad -\pi \leq \omega_2 < \pi,$$

et la fonction de transfert du filtre

$$H(z_1, z_2) = \sum_{k, l \in \mathbb{Z}} h(k, l) z_1^{-k} z_2^{-l} \quad z_1, z_2 \in \mathbb{C},$$

où une équation aux différences reliant l'entrée x à la sortie y :

$$y(n, m) + \sum_{k=1}^{N_1} \sum_{l=1}^{N_2} a_{kl} y(n-k, m-l) = \sum_{k=0}^{M_1} \sum_{l=0}^{M_2} b_{kl} x(n-k, m-l). \quad (*)$$

Les coefficients $(a_{kl})_{k=0, \dots, N_1, l=0, \dots, N_2}$ et $(b_{kl})_{k=0, \dots, M_1, l=0, \dots, M_2}$ sont des constantes complexes. Si $N_1 = N_2 = 0$ on a un filtre *non récursif*, sinon on a un filtre *récursif*.

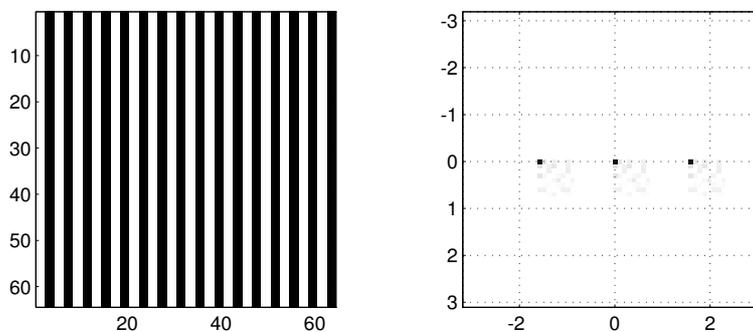
Les remarques que l'on a fait dans le cas monodimensionnel se généralisent facilement au cas bidimensionnel. On peut noter cependant que la causalité de signaux bidimensionnels ne se définit pas de façon aussi naturelle que dans le cas monodimensionnel.

Dans le paragraphe suivant, on s'intéresser plus particulièrement à l'information fréquentielle contenue dans un signal (image) ou filtre bidimensionnel.

Interprétation de la représentation fréquentielle :

Pour des signaux monodimensionnels, *e.g.* de la musique, on a l'habitude de parler d'amplitude et de fréquences. Pour comprendre le cas bidimensionnel considérons seulement une ligne d'une image, l'intensité lumineuse peut être représentée par ses valeurs en fonction de la position sur la grille ou par l'amplitude et la phase des fréquences spatiales.

Sur la figure suivante on a représenté, à gauche, l'image x , formée de bandes de largeur 2 pixels et, à droite, le module la réponse fréquentielle $|\hat{x}|$. On constate que le spectre d'amplitude de x a trois composantes principales : la valeur à fréquence zéro, et deux points symétriques correspondant à la fréquence des bandes verticales (0 =blanc et \max =noir).



Note : La fréquence maximale dans une direction est de $\frac{1}{2 \text{ pixel}}$, la fréquence des bandes étant de $\frac{1}{4 \text{ pixels}}$, les points qui la représentent se trouvent à moitié de la fréquence maximale de l'image (cf. TFD).

Rappelons que pour un signal réel x (p.ex. une image) on a :

$$\hat{x}(-\omega_1, -\omega_2) = \overline{\hat{x}(\omega_1, \omega_2)},$$

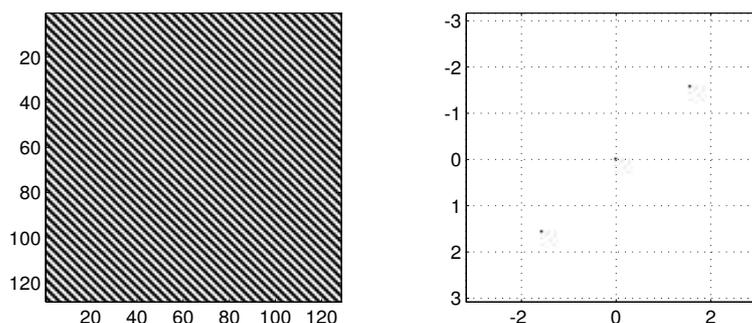
$$|\hat{x}(-\omega_1, -\omega_2)| = |\hat{x}(\omega_1, \omega_2)|, \quad \text{Arg}(\hat{h}(-\omega_1, -\omega_2)) = -\text{Arg}(\hat{h}(\omega_1, \omega_2)),$$

$$\hat{x}(0, 0) = \sum_{k, l \in \mathbb{Z}} h(k, l).$$

Ces relations se traduisent par des symétries dans la représentation du spectre d'amplitude et de la phase de \hat{h} et elles expliquent la grande valeur du spectre d'amplitude pour la fréquence $(0, 0)$.

Note : En anglais $\hat{h}(0, 0)$ est souvent appelé la composante DC (*direct current*) et les coefficients de \hat{h} pour les fréquences non nulles sont les coefficients AC (*alternating current*).

L'image suivante représente des bandes diagonales de largeur 2 pixels. Sur le spectre d'amplitude associé on trouve les principales fréquences.



L'interprétation du spectre d'amplitude peut se résumer ainsi :

- si le spectre d'amplitude comporte de grandes valeurs pour les hautes fréquences, alors l'image contient des données qui varient beaucoup sur une courte distance (textures) ;
- si le spectre d'amplitude comporte de grandes valeurs pour les basses fréquences, alors l'image contient des données qui varient peu (régions de luminosité presque constante).

Filtre passe-bas bidimensionnel.

Comme dans le cas monodimensionnel, il n'existe pas de filtre passe-bas idéal bidimensionnel, par contre l'on peut obtenir facilement des filtres «essentiellement» passe-bas.

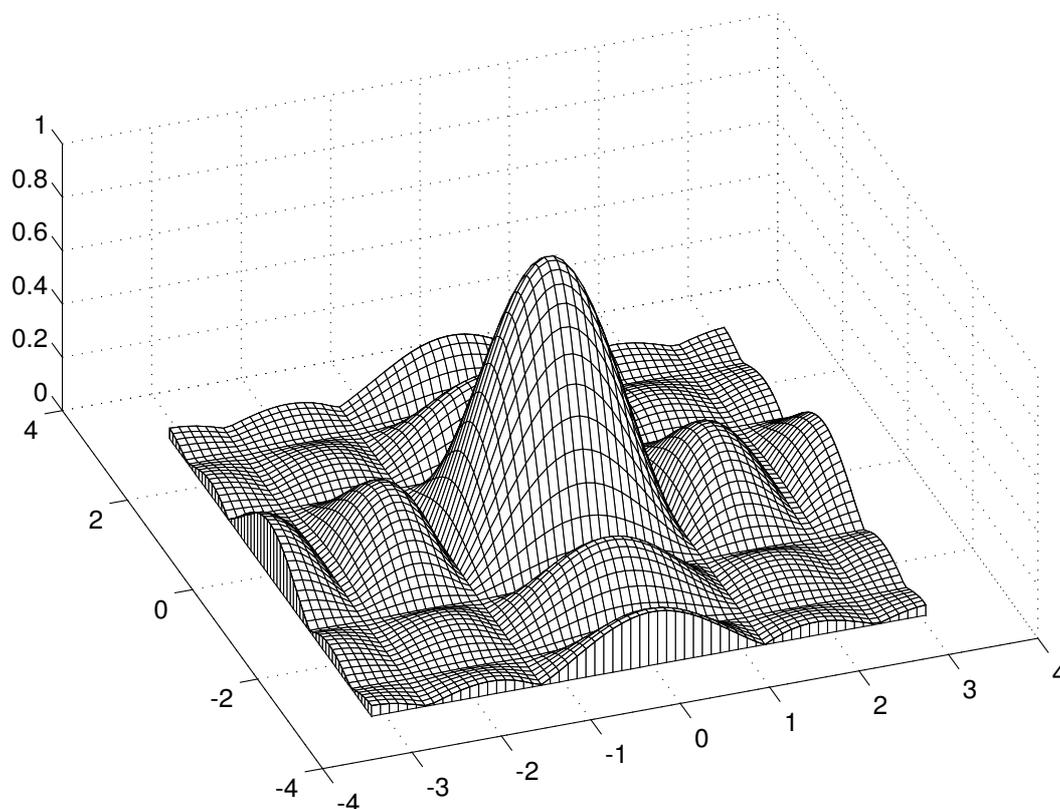
Soit par exemple le filtre moyenne glissante, de réponse impulsionnelle

$$h_5(i, j) = (1/25)rect_{5,5}(i + 2, j + 2).$$

Dans cet exemple la réponse fréquentielle s'obtient facilement à partir du cas monodimensionnel. En effet h et la transformation de Fourier étant séparables, on obtient

$$\hat{h}(\omega_1, \omega_2) = \frac{\sin(5\omega_1/2)}{\sin(\omega_1/2)} \frac{\sin(5\omega_2/2)}{\sin(\omega_2/2)}.$$

Sur la figure suivante on a représenté $|\hat{h}(\omega_1, \omega_2)|$, le spectre d'amplitude de h , pour $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$.



Si l'on applique ce filtre à une image bruitée, c'est-à-dire ayant des hautes fréquences correspondant aux pixels perturbés, on va diminuer le bruit.

Filtre inverse.

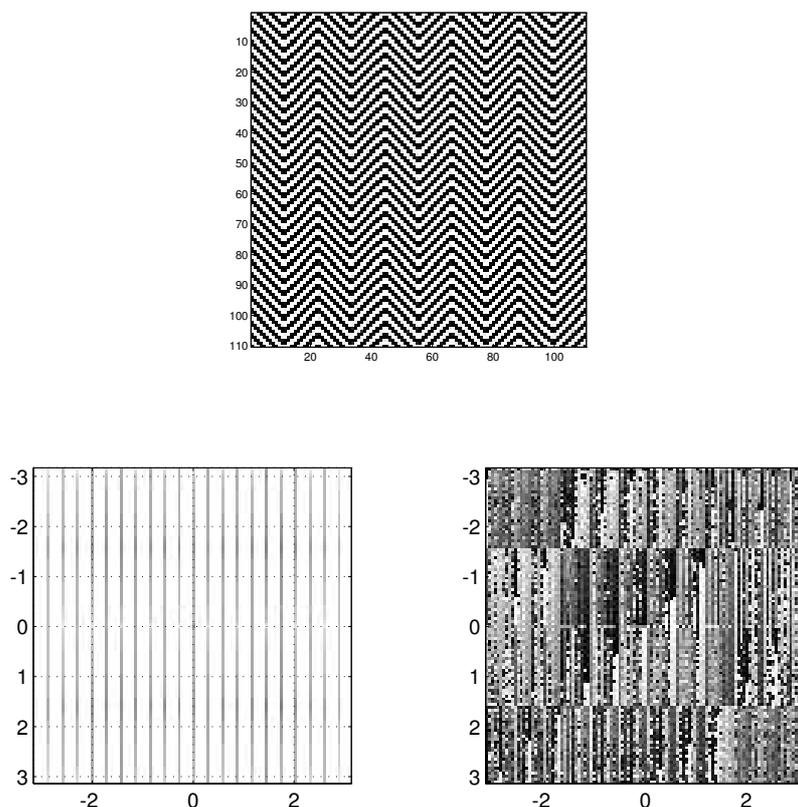
De nombreux phénomènes physiques réels qui perturbent les images, peuvent être modélisés grâce à des filtres bidimensionnels : turbulences atmosphériques, optique défectueuse, mouvement d'un satellite, etc...

Il est souvent possible de trouver les paramètres de ce filtre (système physique continue) et d'essayer de construire un filtre inverse, l'on pourra ainsi récupérer, dans le meilleur des cas, l'information qui a été dégradée.

Importance de la phase.

Très souvent on ne s'intéresse qu'au spectre d'amplitude d'une image, il ne faut cependant pas oublier le spectre de phase, nécessaire pour la reconstruction.

L'image suivante est caractérisée par son spectre d'amplitude, représenté en bas à gauche (0=blanc et max=noir) et son spectre de phase, en bas à droite, centré sur le niveau 128 ($-\pi = 0, 0 = 128$ et $\pi = 255$).



Sur les images suivantes on présente ce qui se passe si l'on échange les amplitudes et les phases de deux images.

En haut on a les images originales «bureau» et «femme», au centre la représentation de leurs spectres d'amplitude et de phase.

L'image «bureau-femme» est obtenue par transformée de Fourier inverse, en utilisant l'amplitude de l'image «bureau» et la phase de l'image «femme».

De même l'image «femme-bureau» est obtenue en utilisant l'amplitude de l'image «femme» et la phase de l'image «bureau».

On constate que la phase contient une grande partie de l'information visuelle.

Note : La taille des quatre images centrales a été réduite pour des raisons de mise en page.

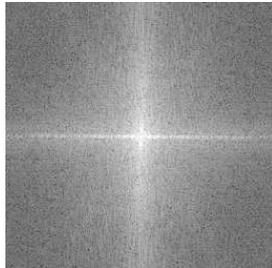
'bureau'



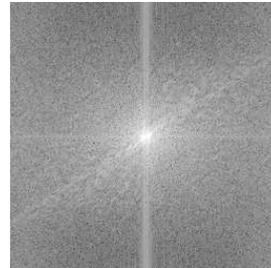
'femme'



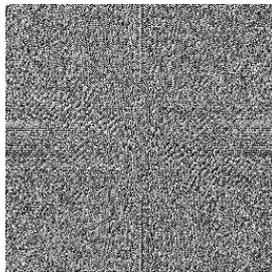
amplitude de 'bureau'



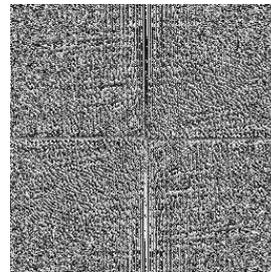
amplitude de 'femme'



phase de 'bureau'



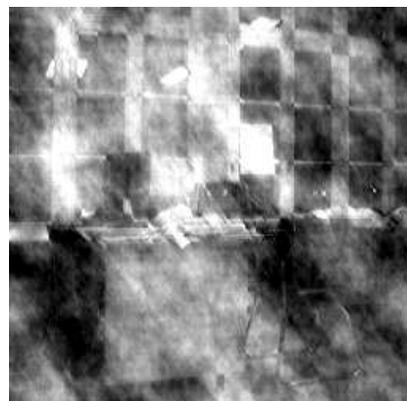
phase de 'femme'



"bureau"- "femme"



"femme"- "bureau"



2.3 Échantillonnage et quantification

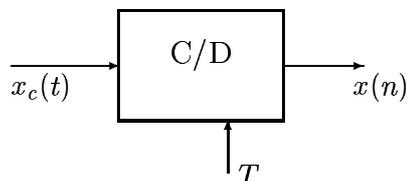
Dans cette section s'intéresse à l'échantillonnage et la quantification de signaux analogiques et aux possibilités de retrouver l'image analogique à partir d'un signal discret.

2.3.1 Échantillonnage de signaux continus.

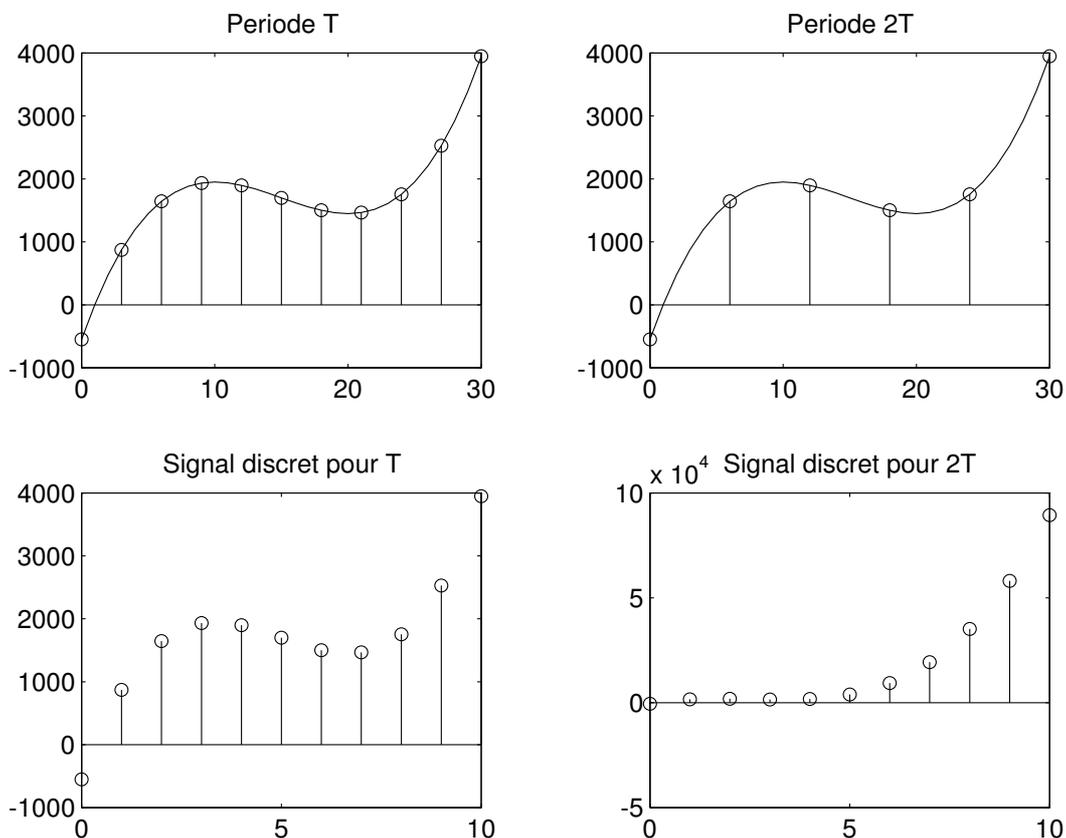
La méthode le plus souvent utilisée pour obtenir un signal discret en temps, $(x(n))_{n \in \mathbb{Z}}$, à partir d'un signal continue en temps, $x_c(t)$, est l'échantillonnage périodique défini par :

$$\forall n \in \mathbb{Z} : \quad x(n) = x_c(nT),$$

où T est la *période d'échantillonnage* et $f_e = 1/T$ est la *fréquence d'échantillonnage*. On représente cette opération grâce à un convertisseur C/D (continu/discret).



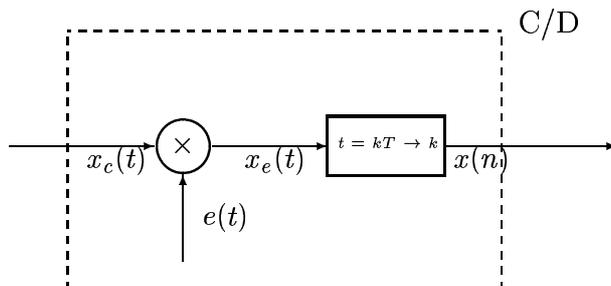
Pour des cadences d'échantillonnage différentes, on obtient des signaux discrets distincts, voir la figure suivante. On a représenté sur la première ligne un signal continu avec des échantillons aux points $(kT)_{k \in \mathbb{Z}}$, resp. $(2kT)_{k \in \mathbb{Z}}$. Sur la deuxième ligne on affiche les signaux discrets associés, chaque fois de $x(0)$ à $x(10)$ (noter en particulier le changement d'échelle en bas à droite).



En général il est impossible de retrouver $x_c(t)$ à partir de $(x(n))_{n \in \mathbb{Z}}$, sans plus d'informations. De plus des signaux continus distincts peuvent donner la même suite d'échantillons. Ce n'est qu'en considérant une classe restreinte de signaux continus que l'on peut lever l'ambiguïté entre l'entrée et la sortie du convertisseur C/D.

Modélisation mathématique du convertisseur C/D idéal.

Sur la figure suivante on a représenté l'algorithme mathématique qui se cache derrière un convertisseur C/D idéalisé.



Remarque : Ce schéma ne correspond pas à un appareil existant, mais il nous permettra de mieux comprendre l'échantillonnage d'un signal continu.

En un premier temps on multiplie $x_c(t)$ par le signal $e(t)$ qui est une somme d'impulsions localisées en les multiples entiers de T :

$$e(t) = \sum_{k \in \mathbb{Z}} \delta(t - kT).$$

On obtient $x_e(t)$, un signal partout nul, sauf aux points $(kT)_{k \in \mathbb{Z}}$:

$$x_e(t) = x_c(t) \cdot e(t) = \sum_{k \in \mathbb{Z}} x_c(kT) \delta(t - kT).$$

Voir aussi la figure de la page précédente, où l'on a représenté un signal continu, $x_c(t)$, et deux signaux $x_e(t)$ ayant des cadences d'échantillonnage différentes ($1/T$ et $1/(2T)$).

Finalement, on obtient le signal discret x , en prenant comme valeur au point k la valeur du k -ième échantillon, c'est-à-dire $x(k) = x_e(kT) = x_c(kT)$.

Ce sont les signaux représentés dans la deuxième ligne de la figure de la page précédente.

On remarque que x ne contient pas d'information explicite sur la période d'échantillonnage T .

Étude fréquentielle du convertisseur C/D idéal.

On va étudier la représentation fréquentielle du modèle introduit au dessus pour obtenir des informations sur une perte éventuelle d'information.

On admet les résultats sur la transformée de Fourier de signaux continus définie, pour $f \in \mathbb{R}$, par

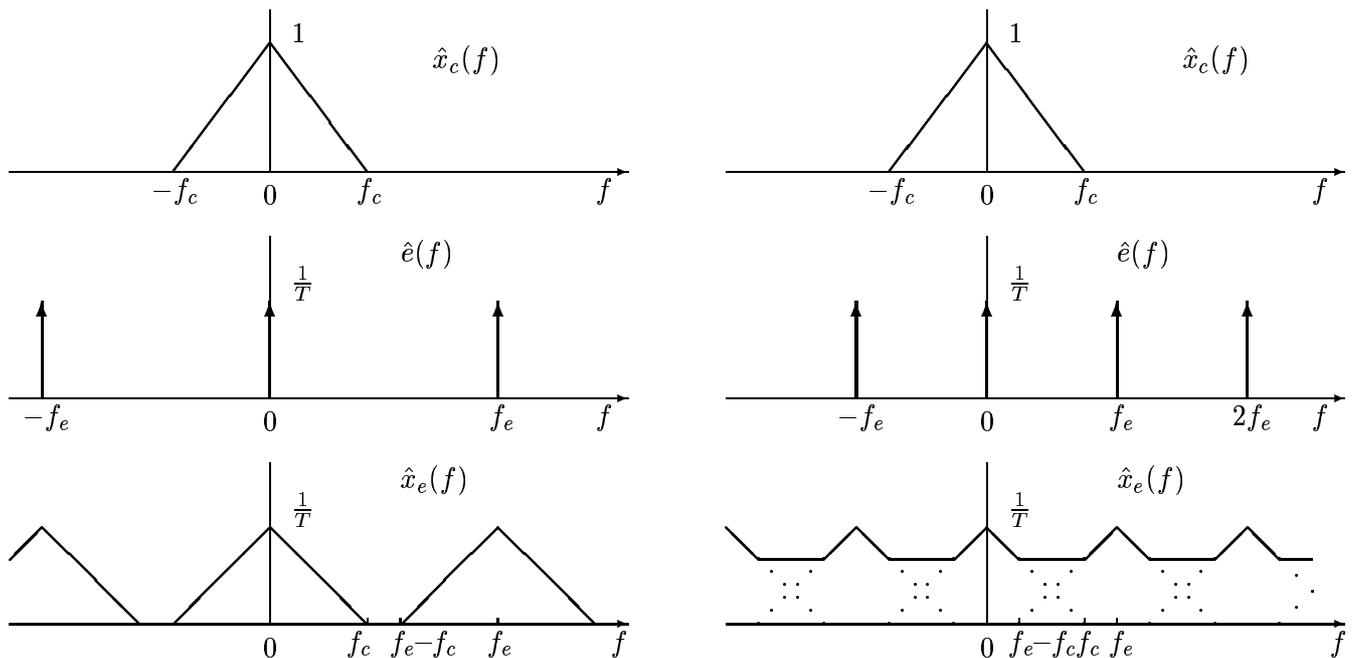
$$\hat{x}_c(f) = \int_{\mathbb{R}} x_c(t) e^{-2\pi i t f} dt.$$

En particulier, on montre qu'entre la transformée de Fourier de $x_c(t)$ et la transformée de Fourier de $x_e(t)$ on a la relation suivante :

$$\hat{x}_e(f) = \frac{1}{T} \sum_{k \in \mathbb{Z}} \hat{x}_c\left(f - \frac{k}{T}\right).$$

Interprétation : On obtient \hat{x}_e en superposant des copies de \hat{x}_c , translatées de multiples entiers de $f_e = 1/T$. On dit que l'on a périodisé le spectre de x_c .

Sur la figure suivante on a représenté le spectre d'un signal continu $x_c(t)$ à bande limitée, c'est-à-dire $\hat{x}_c(f) = 0$ pour $|f| > f_c$. Suivant la valeur de T deux cas se présentent pour le spectre périodisé $\hat{x}_e(f)$.



On a donc les deux cas suivants à traiter :

- Soit $f_e - f_c > f_c$, c'est-à-dire $f_e > 2f_c$ ou $T < \frac{1}{2f_c}$.

Les copies de \hat{x}_a ne se chevauchent pas et on garde une copie exacte de $\hat{x}_a(f)$ à l'intérieur de l'intervalle $[-f_c, f_c]$. Il suffit d'appliquer un filtre passe-bas idéal pour retrouver $x_c(t)$ à partir de $\hat{x}_e(f)$.

- Soit $f_e - f_c \leq f_c$, c'est-à-dire $f_e \leq 2f_c$ ou $T \geq \frac{1}{2f_c}$.

Les copies de $\hat{x}_a(f)$ se chevauchent et s'ajoutent pour donner le spectre de $x_e(t)$. On parle de *repliement* des fréquences ou *aliasing*.

On a ainsi démontré le

Théorème : (Nyquist 1928, Shannon 1949)

Soit $x_c(t)$ un signal continu à bande limitée, c.à.d. $\hat{x}_c(f) = 0$ si $|f| > f_c$.

Alors $x_c(t)$ est déterminé de façon unique par le signal discret $x(n) = x_c(nT)$, $n \in \mathbb{Z}$, si et seulement si $\frac{1}{T} = f_e > 2f_c$.

Ce qui s'énonce aussi de la façon suivante :

Pour représenter un signal continu par une suite d'échantillons, la fréquence d'échantillonnage doit être deux fois plus grande que la fréquence maximale présente dans le signal continu.

Remarque : Avant de passer à la suite, précisons le lien entre $\hat{x}_c(f)$, la transformée de Fourier du signal continu, et $\hat{x}(\omega)$, la transformée de Fourier du signal discret.

On a

$$\hat{x}_e(f) = \sum_{k \in \mathbb{Z}} x_c(kT) e^{-i2\pi kTf} \quad \text{et} \quad \hat{x}(\omega) = \sum_{k \in \mathbb{Z}} x_c(kT) e^{-ik\omega}.$$

En remarquant que $\hat{x}(\omega) = \hat{x}_e\left(\frac{\omega}{2\pi T}\right)$, il vient

$$\hat{x}(\omega) = \frac{1}{T} \sum_{k \in \mathbb{Z}} \hat{x}_c\left(\frac{\omega}{2\pi T} - \frac{k}{T}\right).$$

En utilisant $\boxed{\omega = 2\pi T f}$ on a : si $f \in \left[-\frac{1}{2T}, \frac{1}{2T}\right]$, alors $\omega \in [-\pi, \pi]$.

On obtient donc une normalisation de la fréquence :

\hat{x}_e est périodique de période $f_e = 1/T$ et \hat{x} est périodique de période 2π .

Ceci traduit le fait que la distance entre les échantillons dans le signal $x_e(t)$ est T et dans le signal $x(n)$ elle est 1.

Note : f est mesurée en *Hertz*, $1\text{Hz} = 1\text{s}^{-1}$, tandis que l'unité de ω est le *radian*.

Reconstruction. Convertisseur D/C idéal.

On a vu que si un signal à bande limité, x_c , est échantillonné avec une fréquence $f_e > 2f_c$, alors on peut le reconstruire à partir de la suite $((x(k))_{k \in \mathbb{Z}}$, où $x(k) = x_c(kT)$.

De façon plus précise, si $f_e > 2f_c$, on a $f_c < \frac{f_e}{2} < f_e - f_c$ (voir la figure de la page précédente).

Il suffit donc de considérer \hat{x}_e sur l'intervalle $[-f_e/2, f_e/2]$, et d'appliquer un filtre passe-bas idéal, de fréquence de coupure $f_e/2$, à $x_e(t)$:

$$\hat{h}_{pb}(f) = \begin{cases} T & , \text{ si } |f| < \frac{1}{2T} \\ 0 & , \text{ sinon} \end{cases} \quad \text{et} \quad h_{pb}(t) = \frac{\sin(\pi t/T)}{\pi t/T}.$$

On obtient la formule de reconstruction pour $x_r(t)$:

$$x_r(t) = x_e(t) * h_{pb}(t) = \sum_{k \in \mathbb{Z}} x(k) \frac{\sin\left[\pi\left(\frac{t}{T} - k\right)\right]}{\pi\left(\frac{t}{T} - k\right)}.$$

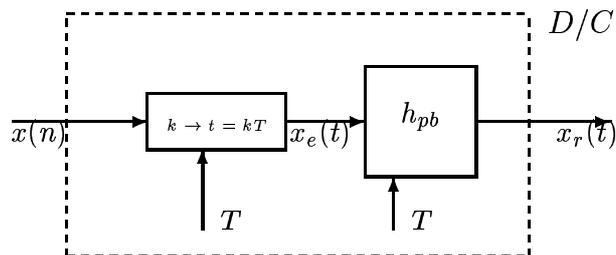
On reconstruit donc un signal continu $x_r(t)$, à partir d'une suite discrète, grâce à une interpolation par convolution continue.

Par ailleurs, on vérifie que l'on obtient les valeurs $x_r(kT) = x_c(kT)$ pour tout entier k . Le signal reconstruit prend donc les mêmes valeurs que le signal original aux points échantillonnés, et ceci indépendamment de la valeur de f_e .

L'interpolation donnera $x_r(t) = x_c(t)$, pour tout t , seulement si l'on est sous les hypothèses du théorème de Shannon, comme l'a montré l'analyse fréquentielle ci-dessus.

Remarquons aussi que le signal reconstruit, $x_r(t)$, est toujours à bande limitée. La plus grande fréquence présente dans x_r est au plus la fréquence de coupure de h_{pb} , c'est-à-dire en général la moitié de la fréquence d'échantillonnage.

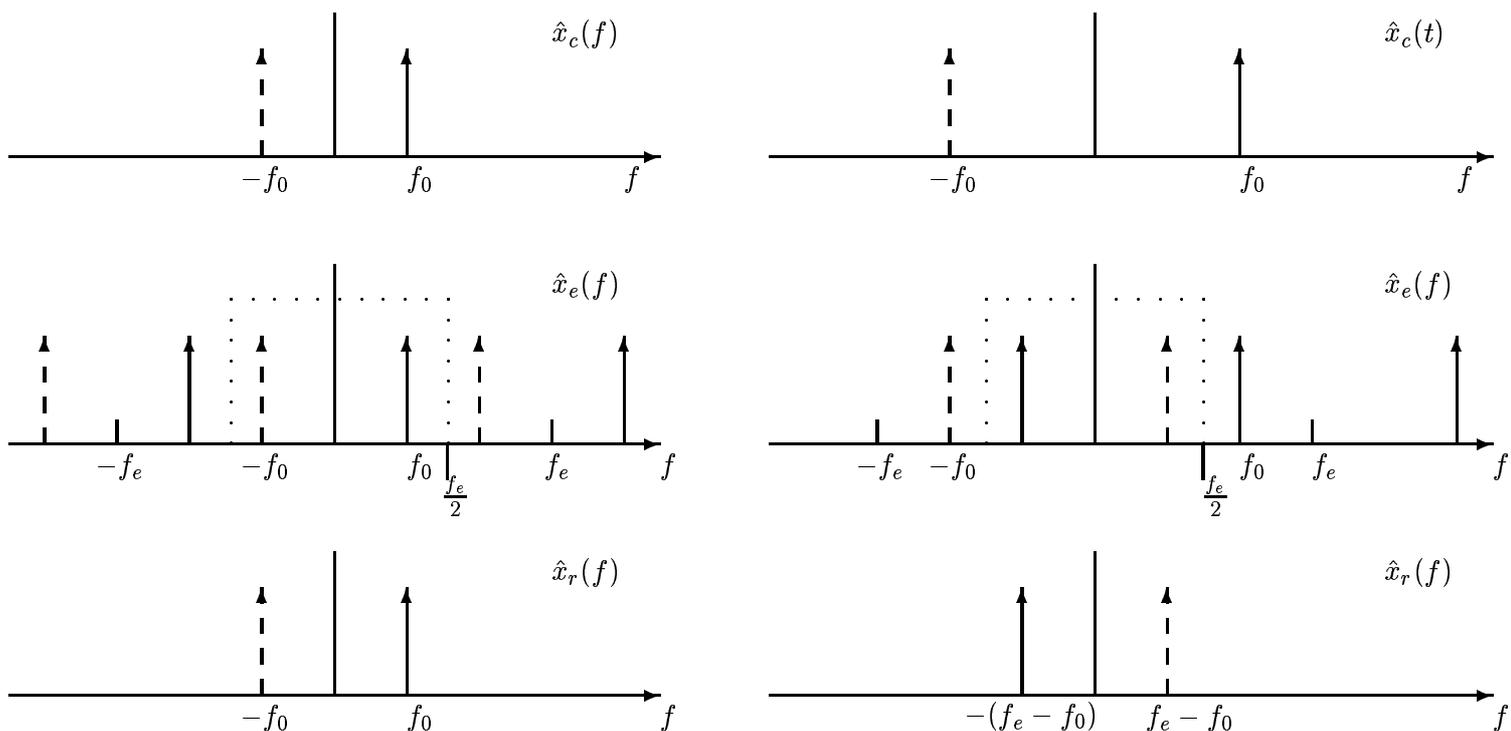
L'opération de reconstruction est schématisée par un convertisseur D/C, dont une description est donnée par le schéma suivant :



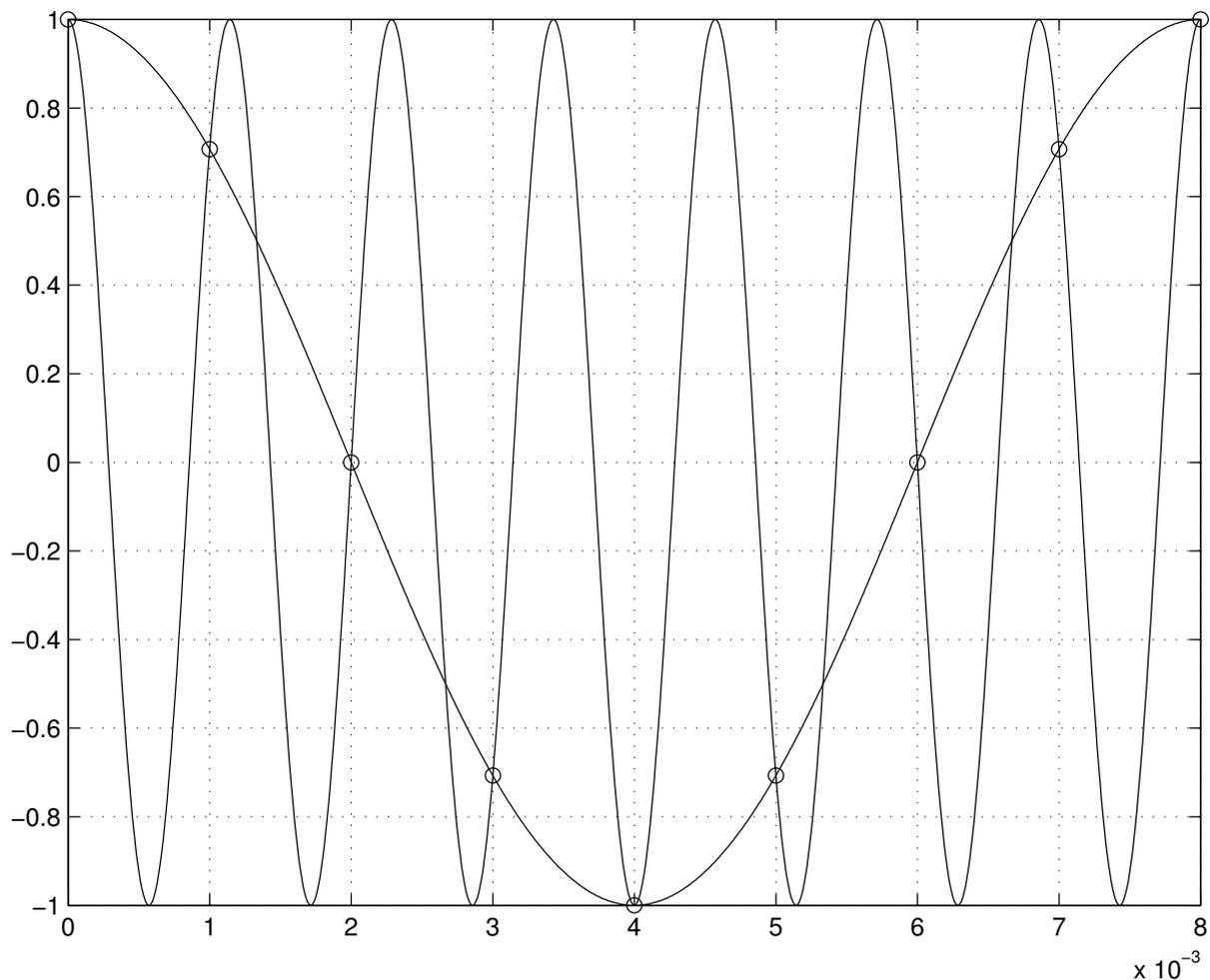
Exemple : Considérons le signal continu périodique $x_c(t) = 2 \cos(2\pi f_0 t)$, de période $1/f_0$. C'est un signal à bande limitée, $f_c = f_0$, dont la transformée de Fourier s'écrit $\hat{x}_c(f) = \delta(f + f_0) + \delta(f - f_0)$.

Sur la figure en bas on a représenté les deux cas possibles. La figure de gauche montre le cas où $f_e > 2f_0$, on obtient le signal reconstruit $x_r(t)$ de fréquence f_0 (i.e. $x_r(t) = x_c(t)$).

Si par contre $f_e < 2f_0$, on remarque que la fréquence f_0 est réfléchiée par $f_e/2$ pour donner une fréquence $f_e - f_0$. Le signal reconstruit sera $x_r(t) = 2 \cos(2\pi(f_e - f_0)t)$.



Application : Le signal discret $x(n) = \cos(\pi n/4)$, $n \in \mathbb{Z}$ a été obtenu en échantillonnant le signal continu $x_c(t) = \cos(2\pi f_0 t)$ à une cadence de mille échantillons par seconde. Quelles sont les valeurs de f_0 qui ont pu donner le signal $x(n)$?



La période de x_c est $p = 1/f_0$ et la fréquence d'échantillonnage est de $f_e = 1000 = 1\text{kHz}$. On trouve donc $x_c(t) = \cos(250\pi t)$ ou $x_c(t) = \cos(1750\pi t)$.

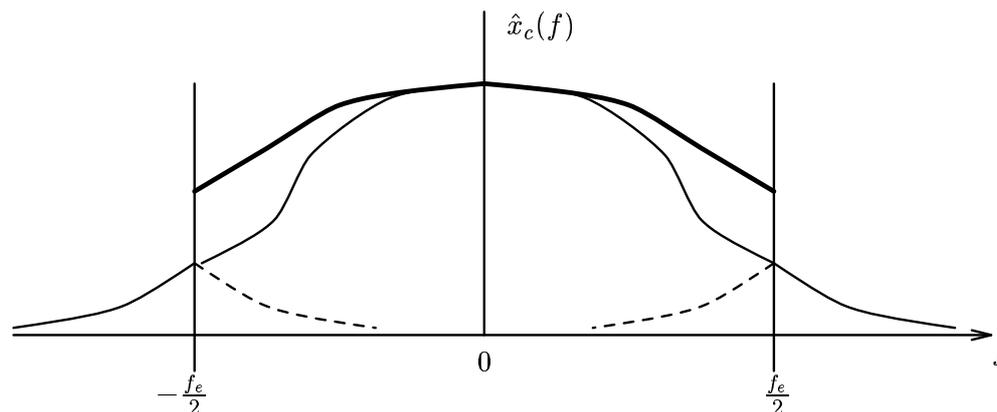
Ce qui donne pour les périodes $p_1 = \frac{1}{125} = 0.008$ ou $p_2 = \frac{1}{875} = \frac{1}{7 \cdot 125} = 0.001142 \dots$

On a représenté $x(0), x(1), \dots, x(8)$, par des \circ et les deux signaux continus en trait continu.

Quelques aspects pratiques.

- Si l'échantillonnage se fait à une fréquence en dessous de la fréquence critique (sous-échantillonnage) la reconstruction ne sera pas bonne. On aura du *crénelage* (angl. *aliasing*) suite à l'apparition de fréquences (angl. *folding frequencies*) et on peut perdre des fréquences du signal original.

Note : L'origine du mot «aliasing», vient du fait que des hautes fréquences sont réfléchies par la fréquence $f_e/2$ et deviennent des basses fréquences, c'est-à-dire qu'elles ont un «alias», *i.e.* un nom d'emprunt.



- En pratique les signaux continus ne sont pas à bande limitée. Or souvent les composantes haute fréquences d'un signal proviennent de bruit ou on décide qu'elles ne sont pas nécessaires, par exemple la transmission de la parole par téléphone n'a pas les mêmes besoins que l'enregistrement sur support numérique de musique.

Pour éviter les problèmes de crénelage on élimine ces hautes fréquences avant l'échantillonnage grâce à un filtre passe-bas.

On dit que l'on applique un filtre *anti-aliasing* avant d'échantillonner.

- Comme un filtre passe-bas idéal n'est pas réalisable, on doit faire la reconstruction grâce à des filtres dont la réponse fréquentielle approche celle du filtre passe-bas.

On peut par exemple utiliser une interpolation linéaire, si la fréquence d'échantillonnage est suffisamment grande on obtient une bonne approximation du signal reconstruit.

- Si la fréquence d'échantillonnage est au dessus de la fréquence critique (sur-échantillonnage) on n'améliore pas la qualité de la reconstruction par le filtre passe-bas idéal.

En pratique il peut être utile d'échantillonner à une cadence plus élevée que la fréquence critique afin d'améliorer les performances du filtre de reconstruction utilisé. Dans ce cas il faudra plus d'espace mémoire et de capacité de calcul.

Dans les applications réelles, il faut trouver un compromis entre une bonne reconstruction et les moyens et le temps à mettre en oeuvre.

2.3.2 Quantification de signaux analogiques

Un convertisseur C/D idéal transforme un signal continu en une suite discrète $(x(n))_{n \in \mathbb{Z}}$, or les éléments de cette suite sont des réels, connus avec une précision infinie. Pour pouvoir traiter un signal analogique dans un système numérique il faut quantifier les valeurs $x(n)$.

Il faut donc associer à tout $x(n)$ une valeur $x_q(n)$ telle que la suite $(x_q(n))_{n \in \mathbb{Z}}$ ne prend qu'un nombre fini de valeurs $\{q_1, q_2, \dots, q_N\}$. En général on fait une quantification uniforme, c'est-à-dire telle que $q_{i+1} - q_i = \Delta q$.

Les valeurs q_i seront alors codées en binaire pour pouvoir être représentées dans un ordinateur ou un circuit numérique.

L'opération de quantification n'est pas réversible, on introduit toujours une *erreur de quantification*, $e(n) = x(n) - x_q(n)$.

Exemple : On veut représenter un signal $v(t)$, qui prend ses valeurs dans l'intervalle $[0, 5]$, à l'aide de deux bits.

On a quatre possibilités (00, 01, 10, 11) de coder une valeur $v(t)$, on va donc partager l'intervalle $[0, 5]$ en quatre : $[0, 1.25[$, $[1.25, 2.50[$, $[2.50, 3.75[$ et $[3.75, 5]$.

Dans chaque intervalle les valeurs sont quantifiées par : $q_1 = 0.625$, $q_2 = 1.875$, $q_3 = 3.125$ et $q_4 = 4.375$.

Finalement q_1 est codé par 00, q_2 par 01, q_3 par 10, et q_4 par 11.

Par exemple la valeur $v(t) = 2.1$ va être codée par 01.

Si on a une valeur négative on est en *dépassement inférieur* (angl. *underflow*), et si $v(t) > 5$ on est en *dépassement supérieur* (angl. *overflow*).

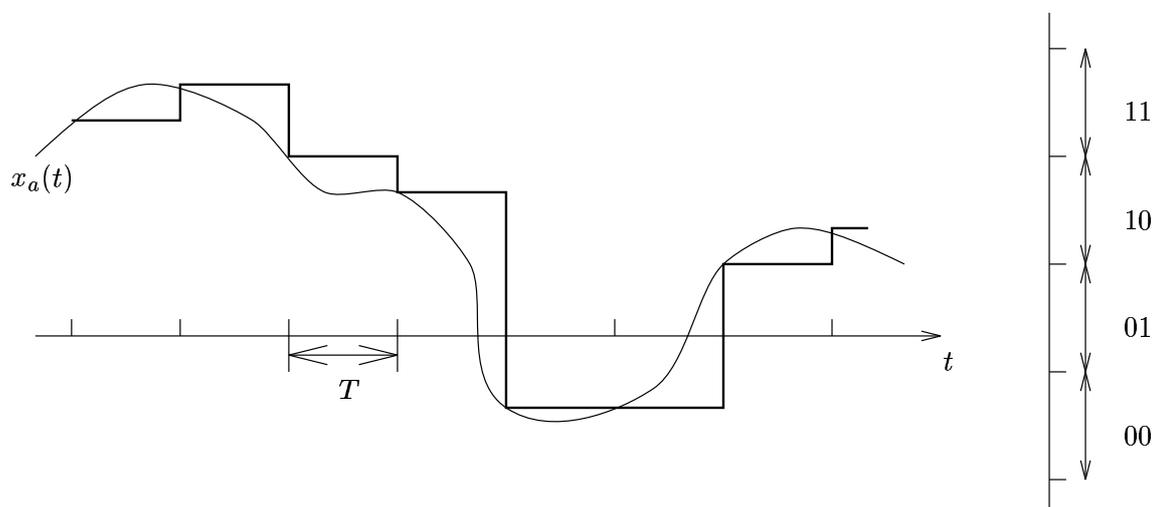
L'erreur de quantification, par échantillon, est majorée par $|e(t)| = |v(t) - v_q(t)| < 0.625$ dans cet exemple.

2.3.3 Convertisseur A/D.

Comme l'on a déjà noté, les convertisseurs C/D et D/C ne correspondent pas à des appareils existants, ce sont des outils pour représenter schématiquement les opérations mathématiques.

En pratique on utilise des convertisseurs A/N (analogique/numérique) et N/A (numérique/analogique).

Un convertisseur A/N est un appareil qui prend en entrée un signal analogique, $x_a(t)$, et donne en sortie une suite discrète échantillonnée et quantifiée, $(x_q(n))_{n \in \mathbb{Z}}$. Son fonctionnement est schématisé sur la figure suivante.



En un premier temps le signal analogique est transformé en un signal en escalier. Pour cela le convertisseur A/N prend une valeur «instantanée» $x_a(t)$ et la maintient pendant une durée T . Pendant ce temps l'appareil doit quantifier et coder en binaire la valeur actuelle de la fonction en escalier.

Dans l'exemple de la figure on a une période d'échantillonnage de T et on code les valeurs du signal sur 2 bits.

Les performances d'un convertisseur A/N dépendent de sa capacité de choisir une valeur aussi instantanée que possible et de la maintenir aussi constante que possible, jusqu'à ce qu'elle est quantifié et codé aussi vite que possible.

Un convertisseur N/A va opérer dans le sens inverse. À partir des valeurs discrètes et quantifiées on crée un signal continu en escalier. Ce dernier donnera un signal analogique après lui avoir appliqué un filtre de reconstruction (interpolation) approprié.

Remarque : Pour une bonne restitution du signal on peut jouer sur l'échantillonnage et la quantification. Ainsi un fort échantillonnage peut équilibrer une grossière quantification. Le standard CD Audio stéréo de 1982 utilise une fréquence d'échantillonnage de 44.1 kHz et une quantification à 16 bits. Les derniers développements en DVD Audio stéréo permettent une quantification de 24 bits échantillonnés à 96 kHz, tandis que les débuts du format SACD permettent 1 bit échantillonné à 2,8224 MHz (*i.e.* 64 fois 44.1 kHz) et une extrême ressemblance avec le signal analogique initial.

2.4 Transformation de Fourier discrète

Dans cette section on va introduire une version discrète de la transformation de Fourier des signaux discrets.

Rappelons que pour un signal discret $(x(n))_{n \in \mathbb{Z}}$, on a considéré sa transformée de Fourier $\hat{x}(\omega)$ qui est une fonction de la variable continue ω . Pour pouvoir étudier le spectre d'un signal discret sur ordinateur il faut donc échantillonner $\hat{x}(\omega)$.

2.4.1 Signaux périodiques.

Soit $(x_p(n))_{n \in \mathbb{Z}}$ un signal discret périodique de période N , avec $N \in \mathbb{N}$, c'est-à-dire $x_p(n) = x_p(n + rN)$, pour tout entier r .

Rappelons que l'on peut trouver seulement N signaux exponentiels discrets et périodiques de période N . Ce sont les signaux de la forme $e_k(n) = Ae^{i\omega_k n}$, où $\omega_k = 2\pi k/N$ et $k = 0, 1, \dots, N-1$.

Par ailleurs on montre qu'une suite N -périodique peut être représentée par une somme de signaux exponentiels dont les fréquences sont des multiples entiers de la fréquence fondamentale $2\pi/N$.

Un signal N -périodique s'écrit donc comme somme des signaux e_k , où $k = 0, 1, \dots, N-1$:

$$\forall n \in \mathbb{Z} : x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_p(k) e^{i2\pi kn/N},$$

où les $\hat{x}(k)$, $k = 0, 1, \dots, N-1$, sont les *coefficients de Fourier* du signal x_p .

On montre la formule inverse

$$\hat{x}_p(k) = \sum_{n=0}^{N-1} x_p(n) e^{-i2\pi kn/N}.$$

On remarque que $\hat{x}_p(k) = \hat{x}_p(k + rN)$ et on peut considérer que l'équation précédente ne donne pas seulement N coefficients $\hat{x}_p(k)$, mais définit en fait une suite périodique.

Conclusion : À une suite N -périodique $(x_p(n))_{n \in \mathbb{Z}}$, on associe une suite N -périodique $(\hat{x}_p(k))_{k \in \mathbb{Z}}$, et réciproquement.

On a ainsi deux façons discrètes de représenter un signal discret périodique.

Convolution circulaire.

Pour les signaux périodiques on avait introduit la notion de convolution circulaire. Rappelons que pour deux signaux périodiques de période N , x_p et y_p , on définit $z_p = x_p *_c y_p$ par

$$z_p(n) = x_p(n) *_c y_p(n) = \sum_{m=0}^{N-1} x_p(n-m) y_p(m).$$

On montre que z_p est N -périodique, on peut donc lui associer sa suite de coefficients de Fourier $(\hat{z}_p(k))_{k \in \mathbb{Z}}$. En fonction des coefficients de Fourier de x_p et y_p on trouve

$$\hat{z}_p(k) = \hat{x}_p(k) \cdot \hat{y}_p(k).$$

Conclusion : La convolution circulaire des signaux périodiques correspond à la multiplication des suites de coefficients de Fourier associées.

Exercices :

1. On considère le signal 10-périodique x_p défini par $x_p(n) = 1$ pour $0 \leq n \leq 4$ et $x_p(n) = 0$ pour $5 \leq n \leq 9$.

Représenter $x_p(n)$. Calculer et représenter $\hat{x}_p(k)$.

2. Montrer que

$$\sum_{r \in \mathbb{Z}} \delta(n + rN) = \frac{1}{N} \sum_{k=0}^{N-1} e^{i2\pi kn/N}.$$

On ne va pas étudier plus en détail les propriétés des coefficients de Fourier associés à un signal périodique. Dans la section suivante on va utiliser la description par coefficients de Fourier des signaux périodiques et l'on verra leurs propriétés à ce moment.

2.4.2 Signaux à durée limitée.

Dans cette section on considère exclusivement des signaux à durée limitée. Soit $x(n)$ un signal de longueur N tel que $x(n) = 0$ si n n'appartient pas à $\{0, 1, \dots, N-1\}$.

Note : Un signal de longueur $M < N$ peut toujours être complété par des zéros en un signal de longueur N .

A tout signal de longueur N on peut associer un signal périodique x_p défini par

$$x_p(n) = \sum_{r \in \mathbb{Z}} x(n + rN) = x(n \bmod N).$$

Réciproquement, à partir du signal N -périodique x_p on retrouve le signal de longueur N en posant

$$x(n) = \begin{cases} x_p(n) & , \text{ si } 0 \leq n \leq N-1, \\ 0 & , \text{ sinon.} \end{cases}$$

On a vu dans la section précédente que l'on peut représenter le signal $x_p(n)$ par un signal N -périodique, formé par les coefficients $\hat{x}_p(k)$. Pour garder la dualité on va associer au signal $x(n)$, de longueur N , une période, de longueur N , de la suite des $\hat{x}_p(k)$.

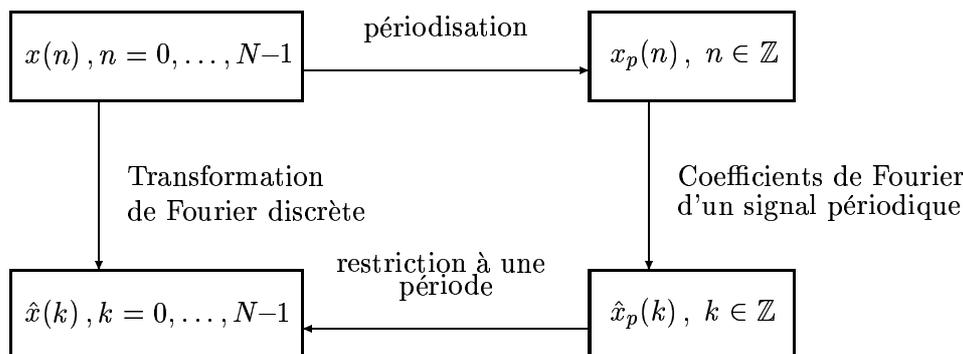
Notons $\hat{x}(k)$ les N coefficients choisis, on a

$$\hat{x}(k) = \begin{cases} \hat{x}_p(k) & , \text{ si } 0 \leq k \leq N-1, \\ 0 & , \text{ sinon,} \end{cases}$$

et réciproquement on obtient une suite périodique en posant

$$\hat{x}_p(k) = \hat{x}(k \bmod N).$$

Conclusion : En utilisant un prolongement périodique d'un signal à durée limitée N , on trouve un signal de durée N , composé des coefficients de Fourier du signal initial périodisé. Le schéma suivant résume ces développements :



On pose souvent $W_N = e^{i\frac{2\pi}{N}}$, donc en particulier $W_N^{kn} = e^{i2\pi kn/N}$, et on peut écrire les équations suivantes :

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) W_N^{-kn}, \quad k = 0, 1, \dots, N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) W_N^{kn}, \quad n = 0, 1, \dots, N-1$$

Définitions :

- $\{\hat{x}(k)\}_{k=0,1,\dots,N-1}$ est appelée *transformée de Fourier discrète (TFD)* de $x(n)$.
- La première équation est appelée *équation d'analyse* du signal.
- La deuxième équation est appelée *équation de synthèse* du signal.

Remarque :

Les formules supposent que $x(n)$ et $\hat{x}(k)$, valent zéro en dehors des points $0, 1, \dots, N-1$. En effet on est seulement intéressé par N valeurs : le signal initial est de longueur N et N coefficients de Fourier sont suffisants pour représenter x .

Mais il ne faut pas oublier la périodicité de x_p , resp. \hat{x}_p , qui est toujours sous-jacente et qui explique un grand nombre de propriétés de la TFD.

Note : Les équations d'analyse et de synthèse se trouvent sous des formes diverses dans la littérature, grâce à la propriété de dualité, que l'on verra plus loin, ces formulations sont équivalentes.

En particulier le facteur $\frac{1}{N}$ de l'équation de synthèse est souvent remplacé par un facteur $\frac{1}{\sqrt{N}}$ dans chacune des équations.

TFD et transformée de Fourier

Soit x un signal de longueur N . On vient de voir que l'on peut lui associer sa TFD

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n)e^{-i2\pi kn/N}, k = 0, 1, \dots, N-1.$$

On a introduit la transformée de Fourier d'un signal comme fonction 2π -périodique de la variable continue ω , et qui s'écrit

$$\hat{x}(\omega) = \sum_{n \in \mathbb{Z}} x(n)e^{-i\omega n} = \sum_{n=0}^{N-1} x(n)e^{-i\omega n}.$$

En posant $\omega_k = \frac{2\pi}{N}k$, on remarque que les $\hat{x}(k)$ sont des échantillons de $\hat{x}(\omega)$ sur l'intervalle $[0, 2\pi]$ distants de $T = \frac{2\pi}{N}$.

Conclusion : On vient de montrer que les coefficients de Fourier du signal périodisé $x_p(n)$ sont les échantillons ($T = 2\pi/N$) de la transformée de Fourier du signal à durée limitée $x(n)$.

Note : On a utilisé la même notation \hat{x} pour la TFD et la transformée de Fourier (avec une variable continue). Suivant le contexte on considère $\hat{x}(k)$ ou $\hat{x}(\omega)$ et il n'y aura pas d'ambiguïté.

Remarque : Dans l'interprétation des coefficients de la TFD on peut noter que les points près de $k = 0$ où $k = N - 1$ correspondent aux basses fréquences et les points autour de $k = N/2$ (division entière) correspondent aux hautes fréquences du signal $x(n)$, de longueur N .

Cas d'un signal de durée infinie.

Soit $x(n)$ un signal de durée infinie qui admet une transformée de Fourier $\hat{x}(\omega)$. On procédant comme plus haut on peut échantillonner cette fonction 2π -périodique sur l'intervalle $[0, 2\pi]$ avec une période de $2\pi/N$.

On obtient ainsi N coefficients $\hat{x}_p(k) = \hat{x}(2\pi k/N)$, que l'on peut considérer comme les coefficients de Fourier du signal N -périodique $x_p(n)$ défini par

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_p(k) W_n^{kn}.$$

En utilisant la définition de $\hat{x}(\omega)$ aux points $2\pi k/N$, on obtient

$$\begin{aligned} x_p(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{m \in \mathbb{Z}} x(m) W_N^{-km} \right) W_N^{kn} \\ &= \sum_{m \in \mathbb{Z}} x(m) \left(\frac{1}{N} \sum_{k=0}^{N-1} W_N^{(n-m)k} \right) = \sum_{m \in \mathbb{Z}} x(m) \left(\sum_{r \in \mathbb{Z}} \delta(n - m + rN) \right) \\ &= \sum_{r \in \mathbb{Z}} x(n + rN) \end{aligned}$$

Il y a donc un phénomène d'aliasing dans le domaine temporel, ceci est à comparer à la situation traitée dans le cas de l'échantillonnage des signaux à spectre borné.

La TFD, interprétée comme échantillonnage fréquentiel, a pour effet de périodiser le signal $x(n)$, on obtient le signal N -périodique $x_p(n)$. A partir de ce signal on ne peut plus retrouver le signal initial $x(n)$.

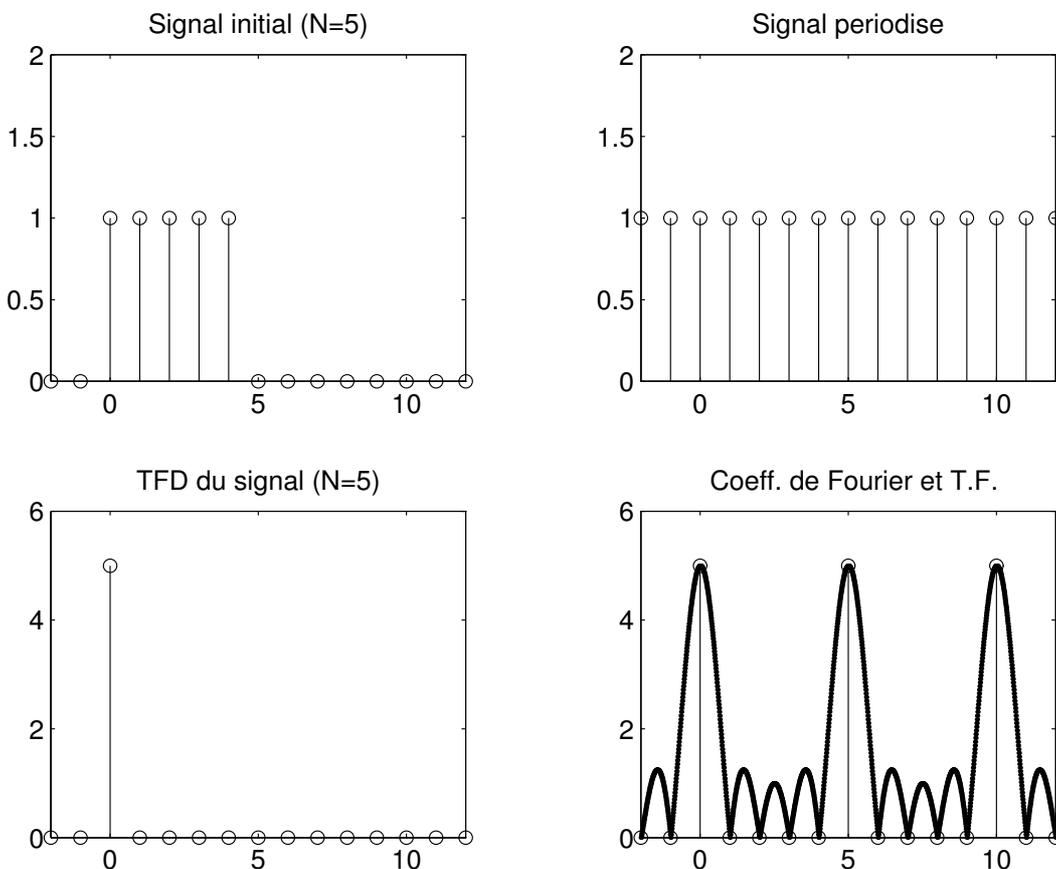
Conclusion : La transformée de Fourier discrète (analyse et synthèse) permet de retrouver le signal initial (*i.e.* $x(n) = x_p(n)$) seulement si $x(n)$ est à durée limitée.

Exemple : (ambiguïté de la TFD)

On définit le signal x pour tout $n \in \mathbb{Z}$ par $x(n) = \begin{cases} 1 & \text{si } 0 \leq n \leq 4 \\ 0 & \text{ailleurs} \end{cases}$.

- Si l'on considère que x est un signal de longueur $N = 5$, on lui associe le signal périodique x_p défini pour tout n par $x_p(n) = x(n \bmod 5) = 1$. On rappelle qu'un signal de durée limitée est supposé commencer en $n = 0$.

La formule d'analyse s'écrit $\hat{x}(k) = \sum_{n=0}^4 x(n)W_5^{-kn}$, on trouve $\hat{x}(k) = 0$ pour $1 \leq k \leq 4$ et $\hat{x}(0) = 5$. Sur la figure suivante on a représenté x , x_p , $\hat{x}(k)$ et $\hat{x}_p(k)$ (les \circ) avec $|\hat{x}(5\omega/2\pi)|$ (trait continu).



- Par contre si l'on considère que x est un signal de longueur $N = 10$, on lui associe le signal 10-périodique x_p défini sur une période par $x_p(n) = 1$ pour $0 \leq n \leq 4$ et $x_p(n) = 0$ pour $5 \leq n \leq 9$.

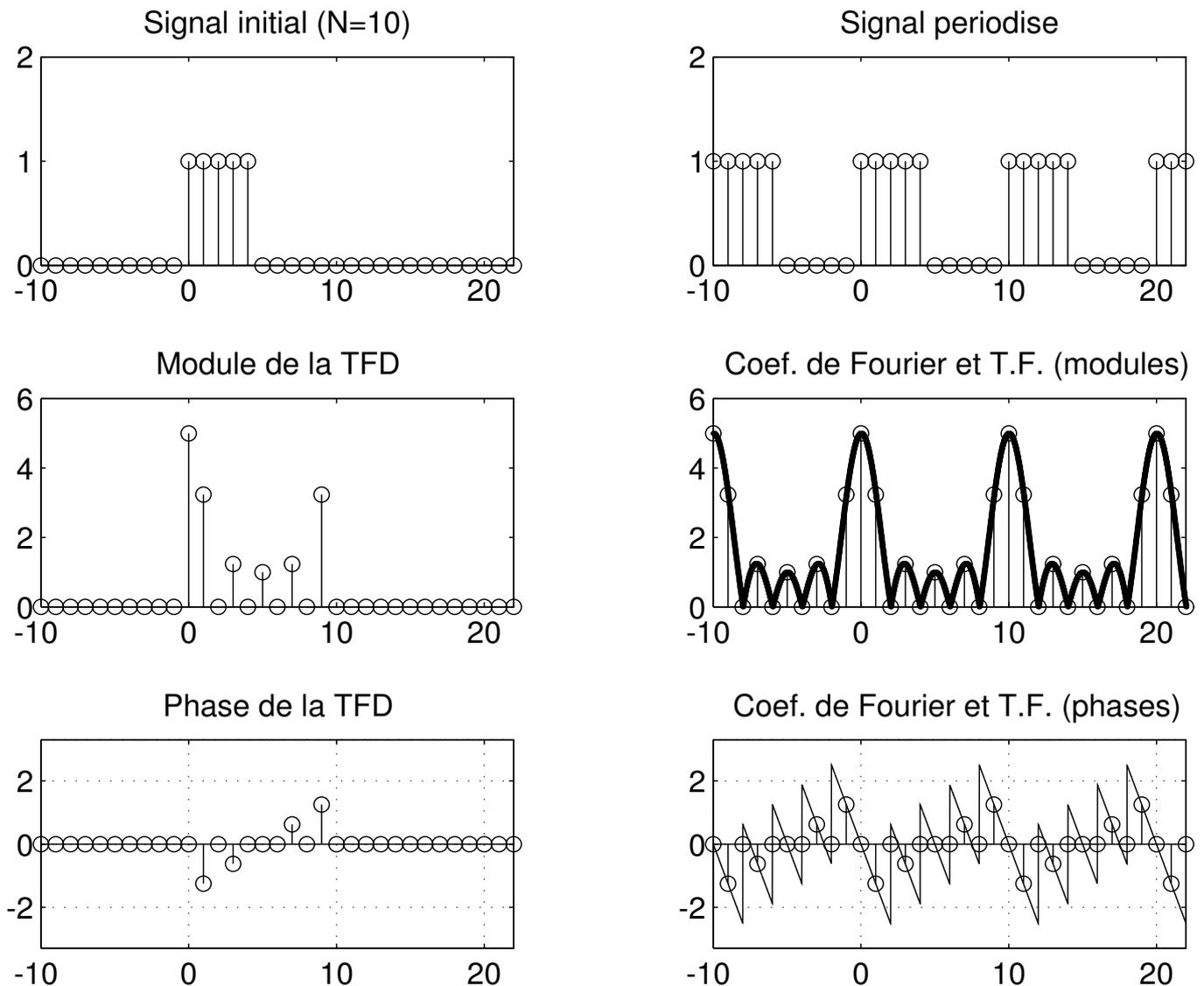
On montre dans ce cas, pour $k \in \{0, 1, \dots, N - 1\}$:

$$\hat{x}(k) = \sum_{n=0}^9 x(n)W_{10}^{-kn} = \sum_{k=0}^4 W_{10}^{-kn} = e^{-i2\pi k/5} \frac{\sin(\pi k/2)}{\sin(\pi k/10)}$$

et on a toujours pour la T.F. du signal $x(n)$

$$\hat{x}(\omega) = e^{i2\omega} \frac{\sin(5\omega/2)}{\sin(\omega/2)}.$$

On a représenté la TFD de $x(n)$ (module et phase) et, sur le même graphique, les modules et phases de $\hat{x}_p(k)$ (avec des \circ) et $\hat{x}(10\omega/2\pi)$ (trait continu).



2.4.3 Propriétés de la transformation de Fourier discrète.

Les propriétés suivantes sont utiles dans les calculs :

$$\begin{aligned} W_N^{kN} &= 1, & W_N^2 &= W_{N/2}, & W_N^{kn} &= W_N^{k(n+N)} = W_N^{(k+N)n}, \\ W_N^{N/2} &= -1, & W_N^{k+N/2} &= -W_N^k, & W_N^{-kn} &= W_N^{k(N-n)} = \overline{W_N^{kn}}. \end{aligned}$$

Dans ce qui suit on ne considère que des signaux de longueur N , la relation entre le signal et sa TFD est représenté par

$$x(n) \longleftrightarrow \hat{x}(k).$$

Les indices n et k vont de 0 à $N - 1$.

- La TFD est linéaire :

$$\alpha x_1(n) + \beta x_2(n) \longleftrightarrow \alpha \hat{x}_1(k) + \beta \hat{x}_2(k), \quad \alpha, \beta \in \mathbb{C}$$

- Dualité :

$$\hat{x}(n) \longleftrightarrow Nx((-k) \bmod N)$$

- Décalage cyclique, $n_0, k_0 \in \{0, 1, \dots, N - 1\}$:

$$x((n - n_0) \bmod N) \longleftrightarrow W_N^{-kn_0} \hat{x}(k)$$

$$W_N^{nk_0} x(n) \longleftrightarrow \hat{x}((k - k_0) \bmod N)$$

- Symétrie :

$$\overline{x(n)} \longleftrightarrow \overline{\hat{x}((-k) \bmod N)}$$

Si $x(n)$ est un signal réel :

$$\hat{x}(k) = \overline{\hat{x}((-k) \bmod N)}$$

$$|\hat{x}(k)| = |\hat{x}((-k) \bmod N)|$$

$$\text{Arg}(\hat{x}(k)) = -\text{Arg}(\hat{x}((-k) \bmod N))$$

En particulier $|\hat{x}(k)| = |\hat{x}(N - k)|$, $1 \leq k \leq N - 1$.

De plus $\hat{x}(0) = \sum_{n=0}^{N-1} x(n)$ est réel et, si N est pair, $\hat{x}(N/2)$ est réel aussi.

- Convolution circulaire :

$$x(n) = \sum_{m=0}^{N-1} x_1((n - m) \bmod N) x_2(m) \longleftrightarrow \hat{x}(k) = \hat{x}_1(k) \hat{x}_2(k)$$

$$x(n) = x_1(n) x_2(n) \longleftrightarrow \hat{x}(k) = \frac{1}{N} \sum_{h=0}^{N-1} \hat{x}_1((k - h) \bmod N) \hat{x}_2(h).$$

Exercices : Ces exercices peuvent se faire grâce à des représentations graphiques et permettront d'illustrer les propriétés (admisses) de la TFD.

1. On donne le signal de longueur $N = 8$ défini par $x(n) = 1$ pour $0 \leq n \leq 4$ et $x(n) = 0$ pour $5 \leq n \leq 7$. Représenter $x(n)$ et la TFD de $\hat{x}(k)$.
2. Considérer le signal de longueur $N = 4$ défini par $x(0) = 1$, $x(1) = 3/4$, $x(2) = 1/2$ et $x(3) = 1/4$.

Représenter $x(n)$, $x_p(n)$ et le signal $y(n)$ qui est tel que $\hat{y}(k) = W_4^{-2k} \hat{x}(k)$.

3. Soit x_1 tel que $x_1(2) = 1$ et $x_1(n) = 0$ ailleurs, et x_2 tel que $x_2(0) = 1$, $x_2(1) = 3/4$, $x_2(2) = 1/2$ et $x_2(3) = 1/4$. Représenter $x_1(n)$ et $x_2(n)$.

En considérant x_1 et x_2 comme signaux de longueur 4, construire leur produit de convolution circulaire. Est-ce que l'on pouvait prévoir le résultat ?

Construire le produit de convolution circulaire de x_1 et x_2 en les considérant comme signaux de longueur 6. Comparer avec le résultat de la convolution linéaire $x_1(n) * x_2(n)$.

4. Comparer le résultat de la convolution circulaire de longueur 4 et 8 des signaux des signaux x_1 et x_2 définis par $x(n) = 1$ pour $0 \leq n \leq 4$ et 0 ailleurs, et x_2 tel que $x_2(0) = 1$, $x_2(1) = 3/4$, $x_2(2) = 1/2$ et $x_2(3) = 1/4$ et 0 ailleurs.

Comparer aussi avec le résultat de la convolution linéaire.

Convolution linéaire et convolution circulaire.

Dans l'exercice précédent on a vu que la convolution circulaire de deux signaux de longueur finie peut coïncider avec la convolution linéaire. Par ailleurs on a vu parmi les propriétés de la TFD que la convolution circulaire de longueur N correspond à la multiplication des coefficients de la TFD de longueur N .

On verra un peu plus loin qu'il existe des algorithmes très performants pour effectuer la TFD. Comme par ailleurs la convolution linéaire est fréquemment utilisée pour caractériser les filtres linéaires on voudrait effectuer la convolution linéaire par multiplication des coefficients de la TFD. Il s'agit donc de voir sous quelles conditions la convolution circulaire donne le même résultat que la convolution linéaire.

Soient $x_1(n)$ un signal de longueur N_1 et $x_2(n)$ un signal de longueur N_2 , avec $N_1 \geq N_2$. On sait alors que le produit de convolution linéaire $x_3 = x_1 * x_2$ est de longueur $N_1 + N_2 - 1$ et que dans le domaine fréquentiel on peut écrire $\hat{x}_3(\omega) = \hat{x}_1(\omega) \cdot \hat{x}_2(\omega)$.

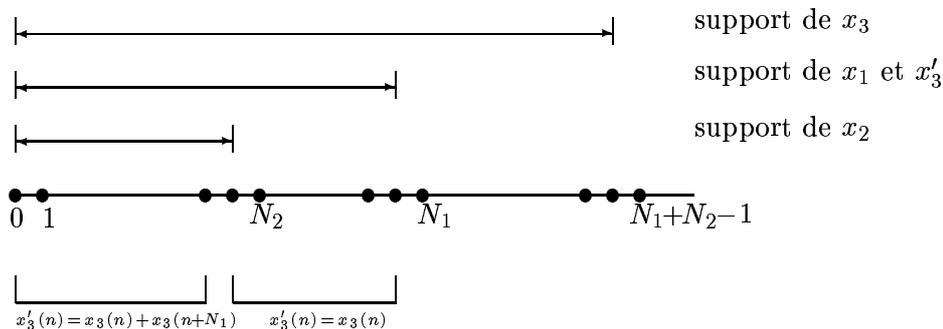
Considérons les échantillons de $\hat{x}_3(\omega)$ en les points $\omega_k = 2\pi k/N_1$, pour $k = 0, 1, \dots, N_1 - 1$. On a vu que l'on peut interpréter les valeurs obtenues comme étant la TFD de longueur N_1 du signal $x'_3(n)$, où $x'_3(n)$ est le résultat de la convolution circulaire (de longueur N_1) de $x_1(n)$ et $x_2(n)$, définie par

$$x'_3(n) = \begin{cases} \sum_{r \in \mathbb{Z}} x_3(n + rN_1) & , 0 \leq n \leq N_1 - 1 \\ 0 & , \text{ailleurs} \end{cases} .$$

Or $x_3(n) = 0$ pour $n < 0$ et $n \geq N_1 + N_2 - 1$, d'où

$$x'_3(n) = \begin{cases} x_3(n) + x_3(n + N_1) & , 0 \leq n \leq N_2 - 2 \\ x_3(n) & , N_2 - 1 \leq n \leq N_1 - 1 \\ 0 & , \text{ailleurs} \end{cases}$$

Donc $x'_3(n)$ est obtenu à partir de $x_3(n)$ de la façon suivante : on copie les valeurs de $x_3(n)$ de 0 à $N_1 - 1$ dans $x'_3(n)$, ensuite on ajoute les $N_2 - 1$ valeurs qui restent dans $x_3(n)$ à $x'_3(n)$, à partir de 0 (voir aussi la figure).



En particulier si $N_1 = N_2$, seulement $x'_3(N_1 - 1) = x_3(N_1 - 1)$, toutes les autres valeurs de x'_3 seront toutes perturbées par les valeurs de x_3 qui se trouvent au-delà de $N_1 - 1$.

D'après ce qui précède il suffit de prolonger les deux signaux x_1 et x_2 par des zéros jusqu'en $N_1 + N_2 - 1$ pour que $x'_3(n) = x_3(n)$ pour $n = 0, 1, \dots, N_1 + N_2 - 1$.

Conclusion : La convolution linéaire de deux signaux, de longueurs respectives N_1 et N_2 , correspond à la convolution circulaire si les signaux sont considérés de longueur au moins égale à $N_1 + N_2 - 1$.

À cet effet on complète chacun des signaux par des zéros (angl. *zero padding*).

Exercice : Comment peut-on utiliser la convolution circulaire pour calculer $y(n) = h(n) * x(n)$? Où $h(n)$ est la réponse impulsionnelle finie d'un filtre linéaire non causal, c'est-à-dire le support de $h(n)$ est $[-M_1, N_1]$, $M_1, N_1 \in \mathbb{N}$.

Filtrage d'un signal très long.

En pratique on va utiliser la TFD pour évaluer la réponse d'un filtre à une entrée $x(n)$. Or souvent le signal va être beaucoup plus long que la réponse impulsionnelle $h(n)$ du filtre et on a vu que l'on peut utiliser la TFD seulement si les deux signaux sont supposés de longueur égale à la longueur du produit de convolution linéaire.

Par exemple, pour effectuer le produit de convolution d'un signal de longueur 16.000 avec une réponse impulsionnelle de longueur 51, il faut une TFD de longueur 16050 au moins, multiplier les coefficients et ensuite faire une TFD inverse.

La mise en place de la méthode présentée au dessus nécessite en pratique des mémoires énormes, un temps de calcul prohibitif et, de plus, il faut attendre la fin des calculs pour obtenir la première valeur du résultat.

En pratique on va couper le signal en plusieurs morceaux, calculer le produit de convolution séparément et, au fur et à mesure, combiner les résultats partiels de manière approprié. On va présenter deux techniques de *convolution par blocs* qui sont utilisées le plus souvent.

Addition superposée : (angl. *overlap-add method*)

Supposons que $h(n)$ est de longueur N et $x(n)$ de longueur beaucoup plus grande. On peut découper $x(n)$ en blocs de la façon suivante

$$x(n) = \sum_{r \in \mathbb{N}} x_r(n - rL),$$

où

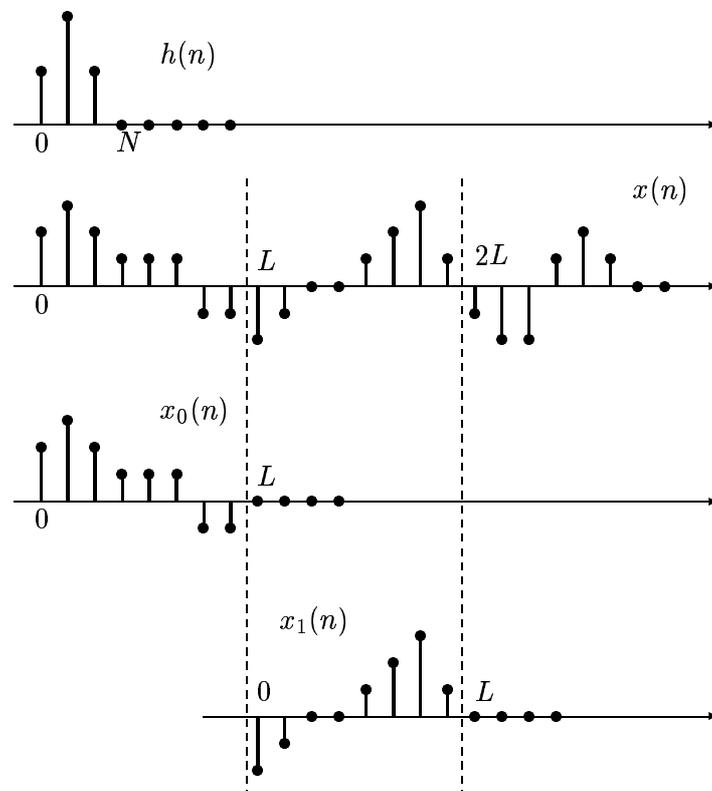
$$x_r(n) = \begin{cases} x(n + rL) & , 0 \leq n \leq L - 1 \\ 0 & , \text{ ailleurs} \end{cases}$$

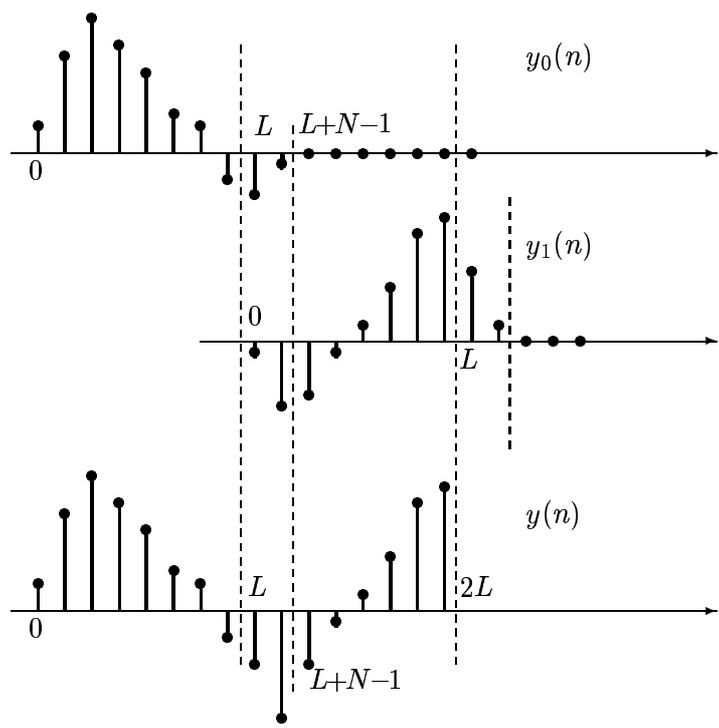
Chaque $x_r(n)$ est un signal de longueur L (commençant en 0) et $x_r(0) = x(rL)$. On obtient

$$y(n) = h(n) * x(n) = \sum_{r \in \mathbb{N}} (h(n) * x_r(n - rL)) = \sum_{r \in \mathbb{N}} y_r(n - rL),$$

où $y_r(n) = h(n) * x_r(n)$ est de longueur $N + L - 1$.

Il suffit donc d'utiliser une TFD de longueur $N + L - 1$, pour évaluer $y_r(n)$. Ensuite il faut ajouter les $y_r(n)$ en tenant compte du décalage L . On constate en particulier que les $N - 1$ premières valeurs d'un bloc sont ajoutées au $N - 1$ dernières valeurs du bloc précédent (voir figure).



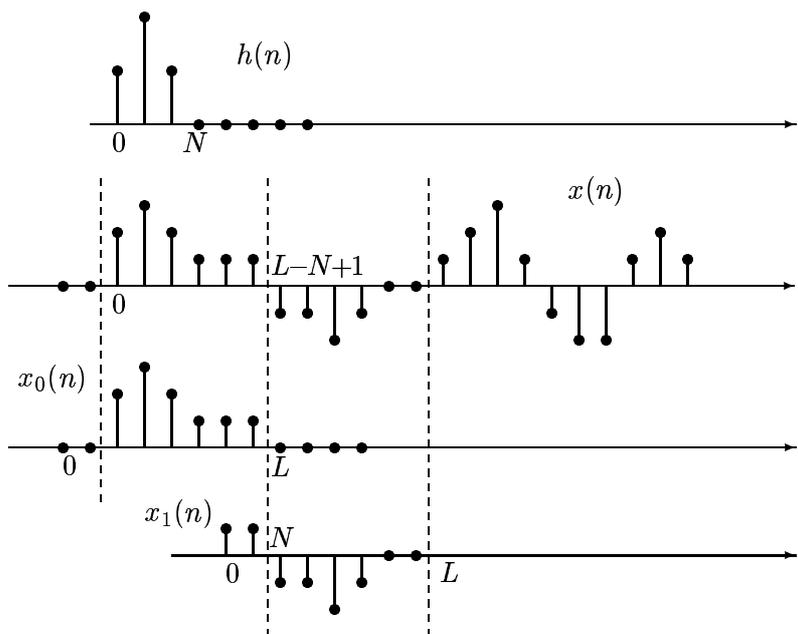


Juxtaposition : (angl. *overlap-save method*)

Dans cette méthode on va découper $x(n)$ en blocs de longueur L de façon à ce que chaque bloc recouvre le précédent sur $N - 1$ points. On utilise donc des sections de la forme

$$x_r(n) = x(n + r(L - N + 1) - (N - 1)), \quad 0 \leq n \leq L - 1.$$

Donc $x_r(n)$ est encore un signal de longueur L avec $x_r(0) = x(rL - (r + 1)(N - 1))$.



Pour chaque $x_r(n)$ on calcule une TFD de longueur L , c'est-à-dire la convolution circulaire de longueur L de $x_r(n)$ et $h(n)$, que nous noterons $y_{rp}(n)$. Or on a vu plus haut que les

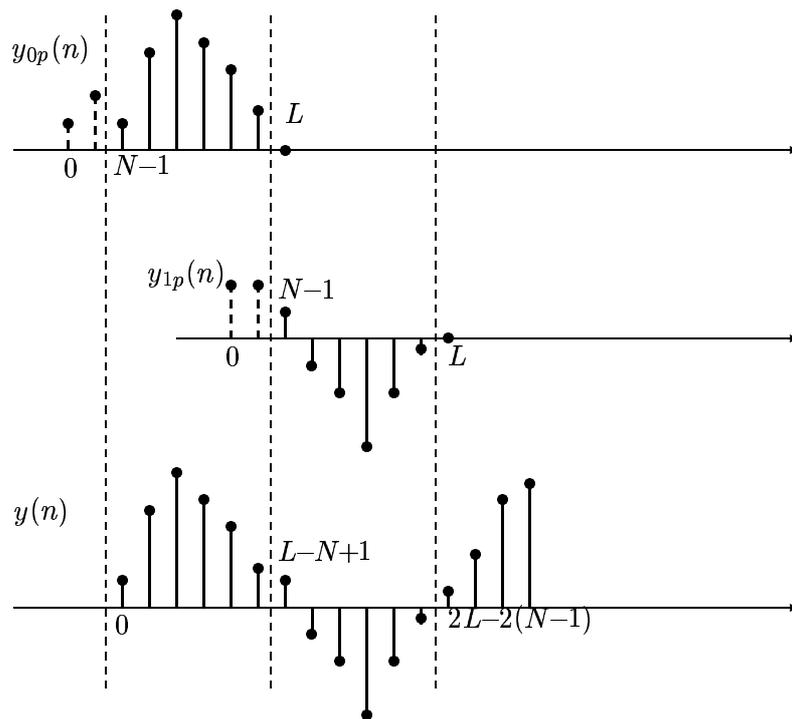
$N - 1$ premières valeurs de ce produit ne correspondent pas au résultat de la convolution linéaire de $x_r(n)$ et $h(n)$, on ne gardera donc que les valeurs $y_{rp}(N - 1)$ à $y_{rp}(L - 1)$.

Le signal en sortie $y(n)$ est donc obtenu en juxtaposant les $y_r(n)$ (voir figure) :

$$y(n) = \sum_{r \in \mathbb{N}} y_r(n - r(L - N + 1) + (N - 1))$$

avec

$$y_r(n) = \begin{cases} y_{rp}(n) & , N - 1 \leq n \leq L - 1 \\ 0 & , \text{ailleurs} \end{cases} .$$



2.4.4 Calcul rapide de la TFD (TFR/FFT)

En tenant compte des symétries qui existent dans les formules d'analyse et de synthèse on peut optimiser considérablement le calcul de la TFD de longueur N . Dans cette section on va présenter les techniques utilisées.

Remarquons d'abord qu'un calcul direct de la TFD nécessite de l'ordre de N^2 additions et multiplications complexes pour traiter un signal de longueur N . On verra que les algorithmes optimisés sont de l'ordre de $N \log(N)$. C'est l'existence de ces méthodes qui a rendu la TFD utilisable en pratique, en particulier pour des applications en temps réel.

Les différents algorithmes que l'on obtient sont regroupés sous l'appellation *transformée de Fourier rapide (TFR)* (angl. *FFT*).

Pour ces algorithmes on distingue en général deux approches. La première consiste à partager l'ensemble des N échantillons en plusieurs sous-ensembles, c'est la TFR *partagée en temps* (angl. *decimation-in-time FFT*). La deuxième consiste, de façon symétrique, à

partager les N coefficients de la transformé de Fourier discrète en sous-ensembles, c'est la TFR *partagée en fréquence* (angl. *decimation-in-frequency FFT*).

Pour illustrer ces méthodes on va en choisir une. En fait on va supposer dans la suite que $N = 2^\nu$ et utiliser la TFR partagée dans le temps. C'est d'ailleurs le plus souvent sous cette forme que la TFR est implémentée, en cas de besoin on complète le signal à traiter avec des zéros. Les méthodes pour une valeur de N quelconque, basées sur la décomposition en facteurs premiers de N , sont souvent plus compliquées et pas nécessairement plus rapides que la méthode que nous allons présenter.

Notons que les premières traces de la méthode présentée remontent à Danielson et Lanczos (1942), mais elle ne s'est répandue qu'à partir du milieu des années soixante avec les travaux de Cooley et Tuckey.

Le signal $x(n)$ étant formé d'un nombre pair de coefficients ($N = 2^\nu$), on peut séparer les coefficients d'indices pairs et les coefficients d'indices impairs :

$$\begin{aligned}
 \hat{x}(k) &= \sum_{n=0}^{N-1} x(n)W_N^{-kn} \\
 &= \sum_{n \text{ pair}} x(n)W_N^{-kn} + \sum_{n \text{ impair}} x(n)W_N^{-kn} \\
 &= \sum_{r=0}^{(N/2)-1} x(2r)W_N^{-2kr} + \sum_{r=0}^{(N/2)-1} x(2r+1)W_N^{-(2r+1)k} \\
 &= \sum_{r=0}^{(N/2)-1} x(2r)(W_N^2)^{-kr} + W_N^{-k} \sum_{r=0}^{(N/2)-1} x(2r+1)(W_N^2)^{-kr} \\
 &= \sum_{r=0}^{(N/2)-1} x(2r)W_{N/2}^{-kr} + W_N^{-k} \sum_{r=0}^{(N/2)-1} x(2r+1)W_{N/2}^{-kr} .
 \end{aligned}$$

Chacune des sommes représente une TFD de longueur $N/2$, la première somme est la TFD des coefficients d'indices pair, la deuxième somme est la TFD des coefficients d'indices impairs. On peut écrire

$$\hat{x}(k) = \hat{x}_p(k) + W_N^{-k} \hat{x}_i(k) .$$

Notons qu'il suffit de calculer $\hat{x}_p(k)$ et $\hat{x}_i(k)$ pour $0 \leq k \leq N/2 - 1$. En effet $\hat{x}_p(k)$ et $\hat{x}_i(k)$ sont $N/2$ -périodiques : pour obtenir les $N - 1$ valeurs $\hat{x}(k)$ chaque $\hat{x}_p(k)$ et $\hat{x}_i(k)$ sera utilisé deux fois.

Or $N/2$ étant un nombre pair, on peut répéter le procédé et on obtient

$$\hat{x}_p(k) = \hat{x}_{pp}(k) + W_{N/2}^{-k} \hat{x}_{pi}(k)$$

$$\hat{x}_i(k) = \hat{x}_{ip}(k) + W_{N/2}^{-k} \hat{x}_{ii}(k)$$

où les $\hat{x}_{pp}(k)$, $\hat{x}_{pi}(k)$, $\hat{x}_{ip}(k)$ et $\hat{x}_{ii}(k)$ sont des coefficients de TFD de longueur $N/4$.

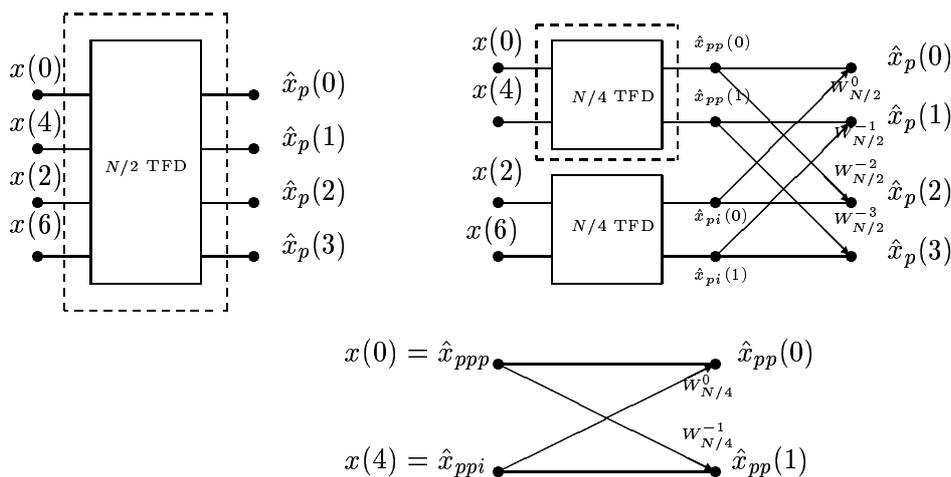
Comme $N = 2^\nu$ on peut appliquer ν fois ce procédé et on obtient N transformés de Fourier discrètes de longueur 1, de la forme $\hat{x}_{p_i p p \dots i p p}(k)$, où l'indice est une séquence de longueur ν , formée des lettres p et de i .

Or une transformé de Fourier discrète de longueur 1 ne dépend pas de k , c'est une constante, c'est-à-dire

$$\hat{x}_{p_i p p \dots i p p} = x(n).$$

Sur la figure suivante on a représenté une partie du calcul d'une TFD de longueur $N = 8$: D'abord on remplace le calcul de la N -TFD par le calcul de deux TFD de longueur $N/2 = 4$, dont une est représentée. Une $N/2$ -TFD est décomposée en deux TFD de longueur $N/4 = 2$, celles correspondant au pointillé du premier schéma ont été représentées. On va ainsi aboutir à huit TFD de longueur $N/8 = 1$, c'est-à-dire les huit coefficients du signal. Sur la figure on a représenté les $N/8$ -TFD correspondant au pointillé du schéma précédent.

Note : Sur ces schémas on utilise les conventions suivantes : un point où deux segments arrivent symbolise une addition des quantités à l'origine des segments, un nombre écrit à coté d'un segment signifie que l'on multiplie la quantités d'origine par ce nombre.



Soit s une séquence de longueur ν formée des lettres p et i , c'est-à-dire $s \in \{p, i\}^\nu$. Quel coefficient $x(n)$ correspond à la TFD de longueur 1 \hat{x}_s ?

Pour répondre à cette question il suffit de constater que si l'on applique m fois le procédé de décomposition, le coefficient $x(n)$ sera «classé p» si le m -ème chiffre de la représentation binaire de n est 0, si ce chiffre est 1, $x(n)$ sera «classé i».

On obtient donc l'indice de l'échantillon correspondant à \hat{x}_s en inversant la séquence s et en remplaçant p par 0 et i par 1.

Par exemple pour $N = 8$, $\hat{x}_{ppi} = x(n)$, avec $n = 100_2 = 4_{10}$.

On parle d'*inversion des bits* (angl. *bit-reversal*). Ceci explique l'ordre des coefficients utilisé dans les schémas précédents.

On trouve donc un algorithme simple, en ν étapes, pour calculer une TFD de longueur N . On part des N coefficients $x(n)$, chacun étant une TFD de longueur 1, on en déduit les coefficients des TFD de longueur 2, puis celles de longueur 4, ..., et ainsi de suite jusqu'à la TFD de longueur N .

Pour pouvoir calculer la TFR il reste donc à voir comment on passe des coefficients d'une TFD de longueur $N/2^m$ aux coefficients d'une TFD de longueur $N/2^{m-1}$.

Considérons une séquence $s \in \{p, i\}^{m-1}$, de longueur $m - 1$, on sait que $\hat{x}_s(k)$ est une TFD de longueur $N/2^{m-1} = 2^{\nu-m+1}$.

On a la relation suivante :

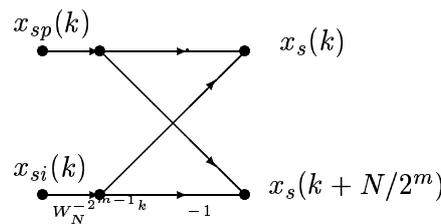
$$\hat{x}_s(k) = \hat{x}_{sp}(k) + W_{N/2^{m-1}}^{-k} \hat{x}_{si}(k).$$

Ici $\hat{x}_{sp}(k)$ et $\hat{x}_{si}(k)$ sont des TFD de longueur $N/2^m = 2^{\nu-m}$ et donc $N/2^m$ -périodiques, d'où par périodicité

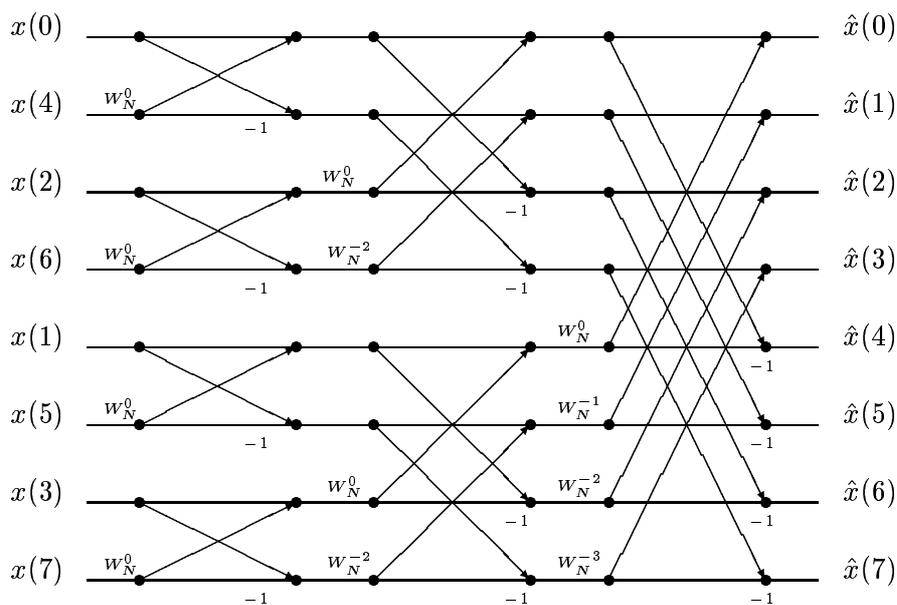
$$\hat{x}_s(k + N/2^m) = \hat{x}_{sp}(k) + W_{N/2^{m-1}}^{-k-N/2^m} \hat{x}_{si}(k).$$

De plus $W_{N/2^{m-1}}^{-k} = W_N^{-2^{m-1}k}$ et $W_{N/2^{m-1}}^{-k-N/2^m} = W_N^{-2^{m-1}k+N/2} = -W_N^{-2^{m-1}k}$.

Donc pour obtenir une TFD de longueur $N/2^{m-1}$ à partir d'une TFD de longueur $N/2^m$ on utilise le schéma en papillon suivant, avec $k = 0, \dots, N/2^m - 1$:



Sur le schéma suivant on a représenté une TFD de longueur $N = 8 = 2^3$ en utilisant les remarques précédentes.



Complexité de la TFR.

On vient de voir que le calcul d'un coefficient $\hat{x}_s(k)$ nécessite une multiplication et une addition (opérations complexes). A chaque étape on doit déterminer N coefficients à partir de l'étape précédente. On a donc $2N$ opérations complexes par étape.

Or on a $\nu = \log_2(N)$ étapes, donc de l'ordre de $N \log(N)$ opérations en tout.

Exercice : Soient $x_1(n)$ de longueur N_1 et $x_2(n)$ de longueur N_2 , deux signaux réels . On veut calculer le produit de convolution linéaire $x_3(n) = x_1(n) * x_2(n)$.

1. Quelle est la longueur de $x_3(n)$? Combien de multiplications (réelles) faut il effectuer pour déterminer $x_3(n)$?
2. Donner un algorithme utilisant la TFD pour calculer $x_3(n)$. Préciser la taille minimale des TFD directes et inverses utilisées.
3. On suppose $N_1 = N_2 = N/2$, où $N = 2^\nu$. Donner une formule pour calculer le nombre de multiplications réelles nécessaires dans l'algorithme du 2 si on utilise la TFR.
4. Comparer le nombre de multiplications de la méthode directe et indirecte. A partir de quelle valeur de N la méthode directe est-elle plus intéressante ?

Exemple de code en C.

L'exemple de code en C suivant est adapté de *Numerical Recipes in C*.

Le signal initial est dans la variable `data` de taille $2*nn$. Pour $i = 0, \dots, nn-1$, `data[2*i]` et `data[2*i+1]` sont respectivement la partie réelle et la partie imaginaire du coefficient d'indice i . On suppose que `nn` est une puissance de deux.

Si `isgn=1` on effectue une TFD directe (analyse), si `isgn=-1` on effectue une TFD inverse (synthèse).

Le résultat est retourné dans la variable `data`, pour $i = 0$ on a la composante à fréquence zéro du signal dans `data[0]` et `data[1]`. Pour $i = 1, \dots, nn/2 - 1$, `data[2*i]` et `data[2*i+1]` contiennent les fréquences positives croissantes, tandis que pour $i = nn/2 + 1, \dots, nn - 1$, on a les fréquences négatives croissantes (en module). Pour $i = nn/2$ on trouve la plus grande fréquence (positive et négative) dans `data[nn]` et `data[nn+1]`.

Dans cette implémentation on peut remarquer en particulier la façon dont sont effectuées les multiplications complexes et le calcul itératif des valeurs du coefficient W_N^{-k} .

Cette procédure peut être adaptée à l'évaluation de la TFD de signaux réels en exploitant les propriétés de symétrie vues plus haut.

```

1  #include <math.h>
2
3  #define SWAP(A,B)      tempr=(A);(A)=(B);(B)=tempr
4
5  void
6  four1(data,nn,isign)
7  float *data;
8  int    nn,isgn;
9  {
```

```

10  int n,mmax,m,i,j,istep;
11  double wtemp,wr,wi,wpr,wpi,theta;
12  float tempr,tempi;
13
14  n=nn << 1;
15
16  j=1;
17  for(i=1;i<n;i+=2) {                               /* bit-reversal */
18      if (j>i) {
19          SWAP(data->values[j-1],data->values[i-1]);
20          SWAP(data->values[ j ],data->values[ i ]);
21      }
22      m=n >> 1;
23
24      while (m >=2 && j>m) {
25          j=j-m;
26          m >>=1;
27      }
28      j=j+m;
29  }
30
31  mmax=2;
32  while(n>mmax){                                     /* Danielson-Lanczos iteration */
33      istep=2*mmax;                                  /* repeated ln(nn) times      */
34      theta=6.28318530717959/(-isign*mmax);
35      wtemp=sin(0.5*theta);
36      wpr=-2.0*wtemp*wtemp;
37      wpi=sin(theta);
38      wr=1.0;
39      wi=0.0;
40      for(m=1;m<mmax;m+=2) {
41          for(i=m-1;i<=n-1;i+=istep) {
42              j=i+mmax;
43              tempr=wr*data[ j ]-wi*data[j+1];
44              tempi=wr*data[j+1]+wi*data[ j ];
45              data[ j ]=data[ i ]-tempr;
46              data[j+1]=data[i+1]-tempi;
47              data[ i ]=data[ i ]+tempr;
48              data[i+1]=data[i+1]+tempi;
49          }
50          wr=(wtemp=wr)*wpr-wi*wpi+wr;
51          wi=wi*wpr+wtemp*wpi+wi;
52      }
53      mmax=istep;
54  }
55
56  if(isign== -1)
57      for(i=0;i<nn;i++) data[i] /= nn;
58  }

```

2.4.5 Transformation de Fourier discrète bidimensionnelle

Soit $x(m, n)$, $0 \leq m, n \leq N - 1$, un signal bidimensionnel fini. Pour obtenir une analyse de Fourier discrète on procède comme dans le cas monodimensionnel et on a les formules suivantes

$$\hat{x}(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) W_N^{-km} W_N^{-ln}, \quad k, l = 0, 1, \dots, N - 1$$

$$x(m, n) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \hat{x}(k, l) W_N^{km} W_N^{ln}, \quad m, n = 0, 1, \dots, N - 1$$

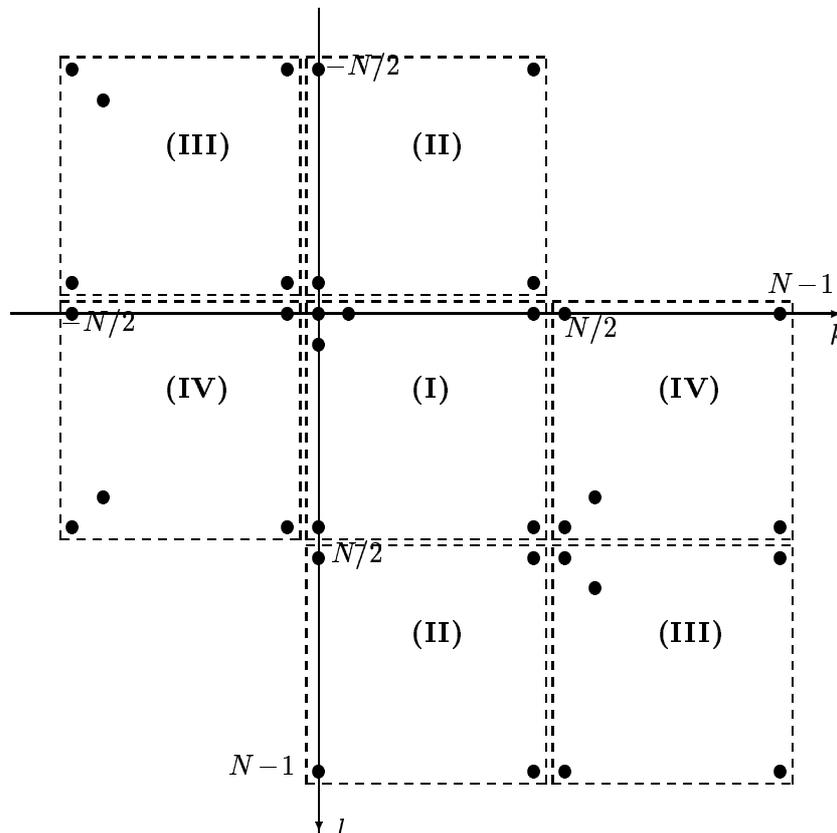
On a prolongé $x(m, n)$ et $\hat{x}(k, l)$ par N -périodicité par rapport à chacune des variables. Comme dans le cas monodimensionnel cette périodicité est sous-jacente et il faut en tenir compte en manipulant les coefficients d'une TFD. De plus la TFD bidimensionnelle est séparable.

Dans le cas où $x(m, n)$ est un signal réel (*e.g.* une image) on a

$$\hat{x}(k, l) = \overline{\hat{x}(N - k, N - l)}, \quad 1 \leq k, l \leq N - 1,$$

c'est-à-dire que N^2 coefficients réels caractérisent $\hat{x}(k, l)$.

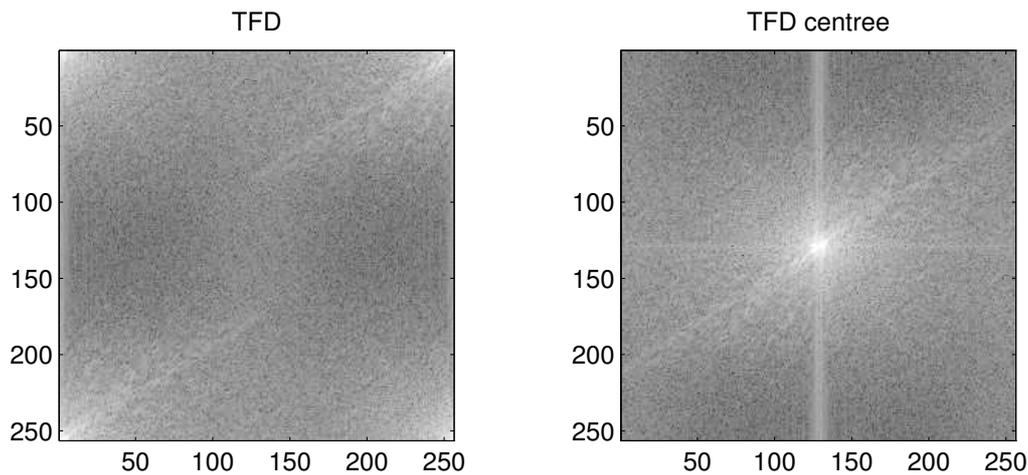
Sur la figure suivante on a représenté la distribution des coefficients de la TFD bidimensionnelle dans le plan orienté de la façon habituelle en traitement des images, c'est-à-dire «droite-bas». De plus on a supposé N pair. Grâce à la N -périodicité on retrouve les quatre blocs I, II, III et IV par une translation de (N, N) .



Un algorithme de TFD bidimensionnelle retournera comme résultat les coefficients sur $[0, N - 1] \times [0, N - 1]$, mais on représente souvent la TFD centrée, obtenue en permutant **I** avec **III** et **II** avec **IV**.

Dans ce cas la fréquence zéro, $\hat{x}(0, 0)$, se trouve en $(N/2 + 1, N/2 + 1)$, tandis que la plus grande fréquence, $\hat{x}(N/2, N/2)$ (réelle), sera placée en $(0, 0)$.

Ces remarques sont illustrées sur la figure suivante ($N = 256$) : à gauche se trouve la TFD avec l'origine en haut à gauche et à droite la TFD centrée, l'origine est au centre.



Complexité de l'algorithme : Comme la TFD bidimensionnelle est séparable on utilise la TFR monodimensionnelle pour la calculer. On applique $2N$ fois l'algorithme monodimensionnel, *i.e.* on a de l'ordre de $N^2 \ln(N)$ opérations à effectuer.

3 Traitement de l'image

3.1 Analyse d'images

Le but de l'analyse d'images est d'obtenir des informations quantitatives et/ou qualitatives sur les objets représentés dans une image numérique (*i.e.* un tableau de nombres).

Il s'agit donc de trouver une représentation *haut niveau* qui doit permettre d'identifier des objets, de les compter, mesurer, etc.

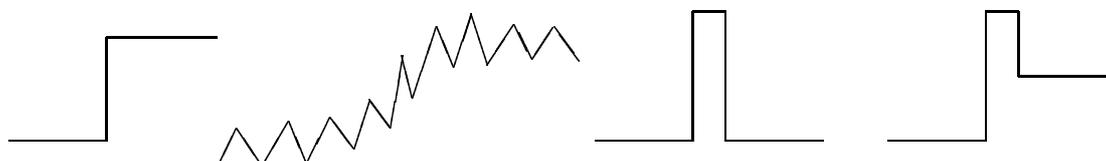
Note : un objet peut être une surface agricole, un nuage, une lettre, une cellule, un organe, une tumeur, un véhicule, une pièce mécanique, etc...

Dans le contexte de l'analyse d'images on parle aussi de *vision par ordinateur*.

Dans les sections suivantes, on propose de détecter un objet grâce à sa frontière avec d'autres objets ou avec le fond. Ceci est seulement une première approche, naïve, du problème. En effet, ainsi on ne traite pas les occultations (un objet couvre partiellement un autre) ou les effets de transparence et d'ombre. Des modèles plus sophistiqués tendent à tenir compte de ces phénomènes.

3.1.1 Détection de bords et de contours

Un bord dans une image est le plus simplement caractérisé par un changement d'intensité lumineuse. Sur la figure suivante on a schématisé différents types de changement d'intensité dans une direction.



On constate qu'un changement d'intensité peut être dû à un bord, à une structure monodimensionnelle (ligne/courbe) ou à un point. De plus on peut avoir une information perturbée par du bruit.

Il ne suffit donc pas de mesurer le changement d'intensité dans une seule direction. Il faut prendre en compte toutes les directions en un point donné, sauf les pixels perturbés par du bruit. De plus il faut fixer l'échelle à laquelle on regarde l'image, c'est-à-dire tenir compte de la hauteur du saut et de la raideur de la pente.

Opérateurs de gradient.

Considérons d'abord une image continue/analogique $g(x, y)$, c'est une fonction à deux variables réelles et à valeurs dans \mathbb{R} . Si l'on suppose que cette fonction est régulière, on sait que la direction de plus grande pente au point (x_0, y_0) est donné par le *gradient* de g :

$$\text{grad}(g)(x_0, y_0) = \begin{pmatrix} \frac{\partial g}{\partial x}(x_0, y_0) \\ \frac{\partial g}{\partial y}(x_0, y_0) \end{pmatrix},$$

et la norme euclidienne de $\text{grad}(g)(x_0, y_0)$ donne la valeur de la plus grande croissance de g au point (x_0, y_0) . On n'a pas besoin de calculer la pente dans toutes les directions, deux directions orthogonales suffisent.

Pour traiter une image discrète, il on remplace la dérivation par des différences finies. On détermine le gradient d'une image discrète en calculant les différences finies dans des directions orthogonales. Ceci est possible de diverses manières et de nombreux opérateurs ont été proposés.

On a par exemple les opérateurs décentrés suivants

$$\begin{aligned}(\Delta_x g)(m, n) &= g(m, n) - g(m - 1, n) \\ (\Delta_y g)(m, n) &= g(m, n) - g(m, n - 1),\end{aligned}$$

ils ont le défaut d'évaluer le saut en $(m + 1/2, n)$ et $(m, n + 1/2)$, et non pas en (m, n) .

Pour éviter cela on utilise les *différences centrées*

$$\begin{aligned}(\Delta_{2x} g)(m, n) &= g(m + 1, n) - g(m - 1, n) \\ (\Delta_{2y} g)(m, n) &= g(m, n + 1) - g(m, n - 1),\end{aligned}$$

où on ne fait pas intervenir la valeur $g(m, n)$.

Pour mesurer des changements dans la direction diagonale, on considère par exemple les *opérateurs de Roberts*

$$\begin{aligned}(\Delta_+ g)(m, n) &= g(m + 1, n + 1) - g(m, n) \\ (\Delta_- g)(m, n) &= g(m, n + 1) - g(m + 1, n),\end{aligned}$$

centrés au point $(m + 1/2, n + 1/2)$.

Ces opérateurs sont réalisables par convolution, avec une réponse impulsionnelle $h(m, n)$. Cette réponse impulsionnelle est en général représentée grâce à un *masque*. Il s'agit d'un tableau de nombres qui représente les *poids* des pixels dans les équations aux différences.

Attention : un masque représente $h(-m, -n)$ et non pas $h(m, n)$, c'est-à-dire le symétrique, par rapport à l'origine, de la réponse fréquentielle $h(m, n)$. On dit que l'on calcule l'*intercorrélacion* du masque et de l'image.

Pour correctement évaluer l'action du masque sur l'image, il faut connaître la position de l'origine de la grille et l'orientation des axes. Dans les exemples suivants on indiquera $h(0, 0)$ par \square et les axes sont orientés «droite-haut» (attention en TP ce sera «droite-bas» !).

Par exemple Δ_+ est représenté par $\begin{bmatrix} 0 & 1 \\ \square & 0 \end{bmatrix}$,

ce qui correspond à la réponse impulsionnelle $h_{\Delta_+}(m, n)$, définie par :

$h_{\Delta_+}(0, 0) = -1$, $h_{\Delta_+}(-1, -1) = 1$ et $h_{\Delta_+}(m, n) = 0$ ailleurs,

et on obtient l'image des différences dans la direction 45° , par

$$(\Delta_+ g)(m, n) = \sum_{k=0}^1 \sum_{l=0}^1 h_{\Delta_+}(m - k, n - l) g(k, l)$$

On représente Δ_- par le masque $\begin{bmatrix} 1 & 0 \\ \square & -1 \end{bmatrix}$.

Et de même pour les différences centrées $\Delta_{2x} : [-1 \quad \boxed{0} \quad 1]$ et $\Delta_{2y} : \begin{bmatrix} 1 \\ \boxed{0} \\ -1 \end{bmatrix}$.

On a ainsi différentes possibilités pour déterminer les coordonnées du vecteur gradient discret.

À partir de ces coordonnées on peut déterminer la direction de la plus grande pente. Pour évaluer la croissance dans cette direction, il faut calculer la norme du vecteur gradient.

En général, on utilise l'une des trois normes suivantes :

$$N_1(\text{grad}(g)) = |g_x| + |g_y|, \quad N_2(\text{grad}(g)) = \sqrt{g_x^2 + g_y^2}, \quad N_3(\text{grad}(g)) = \max(|g_x|, |g_y|).$$

On montre qu'elles sont équivalentes d'un point de vue théorique, pour les applications on tiendra compte du nombre d'opérations nécessaires et des problèmes dus à la grille de discrétisation.

En tout point (m, n) de l'image, on obtient ainsi une estimation de la pente. Si la valeur obtenue est au-dessus d'un seuil τ donné, on dira que l'on a un point du bord. Le résultat est l'image des bords de g , appelée en anglais *edge map* :

$$e(m, n) = \begin{cases} 1 & , \text{ si } N(\text{grad}(g)(m, n)) > \tau \\ 0 & , \text{ sinon} \end{cases}$$

Exercice :

Comparer les valeurs obtenues par N_1 , N_2 et N_3 , si l'on calcule le gradient discret avec Δ_{2x} et Δ_{2y} .

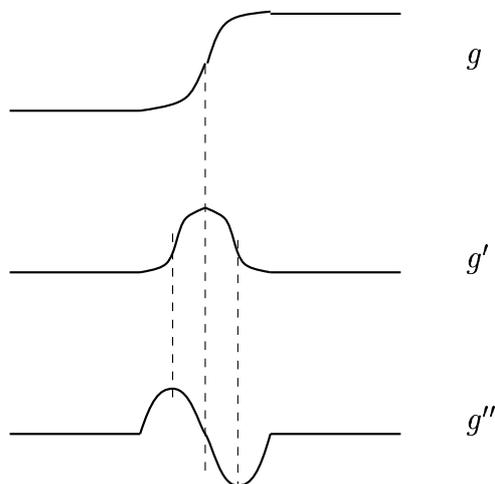
Même question si le gradient discret est obtenu grâce à Δ_+ et Δ_- .

En déduire que N_3 est la norme la mieux adaptée aux opérateurs Δ_+ , Δ_- et Δ_{2x} , Δ_{2y} .

Le Laplacien.

Les opérateurs de gradients donnent les meilleurs résultats si le changement d'intensité lumineuse est abrupte. Si la pente de la fonction est moins raide, on a intérêt à utiliser une autre technique.

Sur le schéma suivant on représente une coupe dans une direction de l'image $g(x, y)$, et les dérivées première et seconde dans la même direction.



On constate que la dérivée seconde réagit des deux cotés d'un bord, une fois avec un signe positif et une fois avec un signe négatif. Dans le cas de la figure, le passage par zéro (angl. *zero crossing*) de la dérivée seconde correspond exactement au centre de la pente de la fonction. L'idée est donc de déterminer l'emplacement d'un bord grâce au passage par zéro de la dérivée seconde.

Pour détecter les passages par zéro d'une fonction de deux variables réelles, on utilise un opérateur invariant par rotations. On montre que le plus simple de ces opérateurs est le Laplacien, noté Δg ou $\nabla^2 g$, défini par

$$\nabla^2 g(x, y) = \frac{\partial^2 g}{\partial x^2}(x, y) + \frac{\partial^2 g}{\partial y^2}(x, y).$$

Pour calculer le Laplacien discret on peut utiliser, comme pour le gradient discret, différentes approximations par différences finies. En utilisant une représentation par masques, on a par exemple

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & \boxed{4} & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & \boxed{8} & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & \boxed{4} & -2 \\ 1 & -2 & 1 \end{bmatrix}.$$

Exercice : Écrire les équations aux différences pour les trois masques précédents.

En pratique, on va appliquer un de ces masques et, grâce à un seuil, on détectera les passages par zéro du Laplacien discret.

On constate que le Laplacien discret est très sensible aux coins, lignes, fins de lignes et aux points isolés. Ceci entraîne en particulier que des pixels bruités vont donner une réponse forte au Laplacien discret et vont engendrer des faux bords. Pour des images réelles il faut donc envisager un débruitage avant la détection des bords.

Note : Considérons le masque centré m_5 , défini par $\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$.

Calculons la différence de $\delta(m, n)$ et $m_5(m, n)$:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

On obtient le masque de $1/5(\nabla^2 g)(m, n)$. Entre le filtre basse-bas m_5 et le filtre passe-haut ∇^2 on a la relation $\nabla^2 = 5(\delta - m_5)$.

Gradient discret pour images bruitées.

Pour réduire l'importance du bruit, on va régulariser l'image, avant d'extraire les bords grâce au gradient. Pour cela, on détermine les moyennes locales, avant de calculer les différences, ou, ce qui est équivalent, on calcule les différences des moyennes locales.

Des exemples sont les filtres de *Prewitt* :

$$\bar{\Delta}_{3x} : \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \bar{\Delta}_{3y} : \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix},$$

ou les filtres de *Sobel* :

$$\bar{s}_x : \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \bar{s}_y : \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix}.$$

A partir de ces filtres on peut déterminer la valeur de la pente en un point et l'on procède comme pour les opérateurs de gradient.

Il faut cependant noter que le fait de calculer des moyennes rend l'image floue, et, par conséquent, la réponse aux filtres sera plus faible. On aura dans ce cas tendance à utiliser les normes N_1 et N_2 pour calculer la pente. On remarque de plus la détection de l'emplacement des bords devient moins précise.

Exercice : Étudier le gradient discret de Frei-Chen, défini par les masques

$$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \quad \text{et} \quad \frac{1}{2 + \sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & 1 \end{bmatrix}.$$

Laplacien discret pour images bruitées.

Pour détecter les passages par zéro du Laplacien dans une image bruitée, on régularise d'abord l'image grâce au filtre gaussien $G_\sigma(x, y)$.

On rappelle que la Gaussienne s'écrit, en une dimension,

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

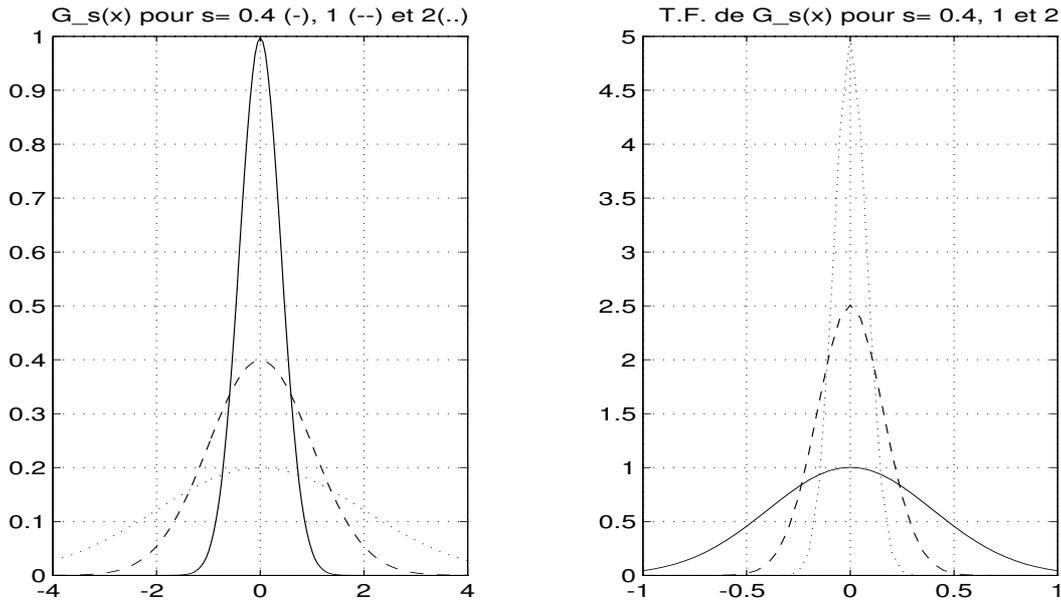
et, en deux dimensions,

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

Le paramètre réel σ contrôle la largeur de la Gaussienne et les coefficients sont choisis tels que l'intégrale sur \mathbb{R} , respectivement \mathbb{R}^2 , est égale à 1.

En appliquant la Gaussienne G_σ à une image, on enlève, par régularisation, les structures à une échelle (spatiale) plus petite que σ . Le filtre gaussien est un filtre passe-bas et l'on montre que c'est le filtre linéaire régulier réalisant le meilleur compromis de localisation en espace et en fréquence. Noter cependant que la Gaussienne rend l'image floue partout et dans toutes les directions, en particulier un bord sera effacé si σ est suffisamment grand.

Sur la figure suivante on a représenté $G_\sigma(x)$ et $\hat{G}_\sigma(f) = \sqrt{2\pi} \sigma e^{-2\pi^2 \sigma^2 f^2}$, pour différentes valeurs de σ .



On est ainsi amené à détecter les passages par zéro de $\nabla^2(G_\sigma * g)(x, y)$, ceci nécessite le choix d'un seuil pour le Laplacien, et d'une échelle σ .

En faisant varier σ , on trouve des bords à différentes échelles. Une analyse à petite échelle donnera beaucoup de bords, tandis qu'à grande échelle on ne verra que les bords importants. On fait une *analyse multiéchelle* de l'image.

La théorie du *scale space* de Witkin propose de détecter d'abord les bords à une grande échelle : on obtient les bords importants, sans bruit, mais avec une mauvaise précision. Ensuite, il faut suivre ces bords à travers les échelles, jusqu'aux petites échelles, où l'on a une bonne localisation, mais aussi du bruit.

Cette méthode qui, en théorie, permet de trouver les bords significatifs de façon précise, est difficile à appliquer. On ne peut pas suivre le paramètre réel σ vers 0, on est obligé de choisir un nombre fini d'échelles que l'on souhaite analyser.

Note : Un exemple d'analyse multiéchelle linéaire est l'analyse par ondelettes.

On peut simplifier la détection des passages par zéro du Laplacien en remarquant que

$$\nabla^2(G_\sigma * g)(x, y) = (\nabla^2 G_\sigma) * g(x, y) .$$

Il suffit donc de filtrer l'image avec $\nabla^2 G_\sigma$ afin d'obtenir les bords à l'échelle σ .

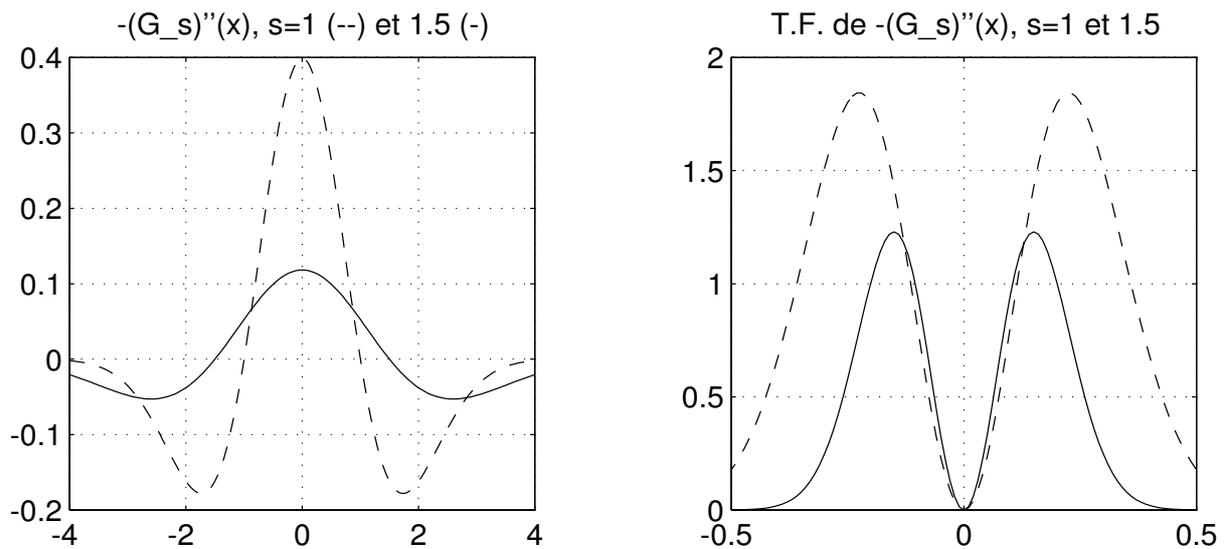
Dans le cas monodimensionnel on a

$$G_\sigma''(x) = -\frac{1}{\sqrt{2\pi} \sigma^3} \left(1 - \frac{x^2}{\sigma^2}\right) e^{-\frac{x^2}{2\sigma^2}} ,$$

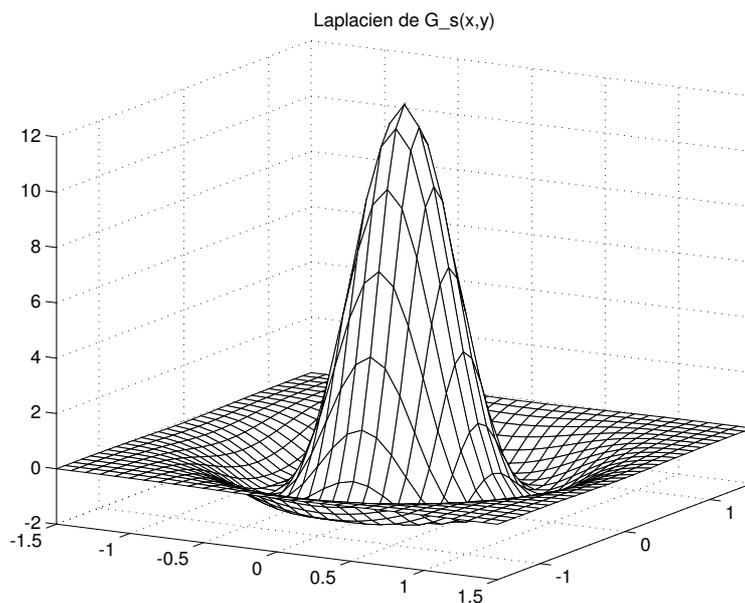
et

$$\widehat{G_\sigma''}(f) = -(2\pi)^{5/2} \sigma f^2 e^{-2\pi^2 \sigma^2 f^2} .$$

Sur la figure suivante on a représenté $-G_\sigma''(x)$ et sa réponse fréquentielle $-\widehat{G_\sigma''}(f)$, pour différentes valeurs de σ . On notera que la réponse fréquentielle est positive et correspond à un filtre passe-bande.



Sur la figure suivante on a représenté $\nabla^2 G_\sigma(x, y)$, ce filtre est souvent appelé le *chapeau mexicain*.



En pratique on restreint $\nabla^2 G_\sigma$ à un support fini, en effet, la décroissance de la Gaussienne étant très rapide, on peut donner un poids nul aux points qui sont loin du centre, tout en gardant une bonne approximation numérique.

On obtient le masque gaussien $MG_{(d,\sigma)}$, de taille $(2d + 1)^2$ et de largeur σ

$$MG_{(d,\sigma)}(m, n) = \begin{cases} C \left(1 - \frac{m^2+n^2}{2\sigma^2}\right) e^{-\frac{m^2+n^2}{2\sigma^2}} & , \text{ si } -d \leq m, n \leq d \\ 0 & , \text{ ailleurs} \end{cases}$$

où C est une constante qui normalise la somme des coefficients du masque à 1.

Comme on a déjà remarqué, la Gaussienne rend l'image floue et on a à déplorer une perte de précision dans la localisation des bords. Il existe d'autres opérateurs, non linéaires, qui sont plus performants pour le débruitage et ne perturbent pas les bords.

Filtre de Canny-Deriche.

Les détecteurs de bords présentés plus haut ont été introduit de façon heuristique. Canny a proposé une approche analytique, il a cherché un filtre de réponse impulsionnelle anti-symétrique, optimal pour la détection de bords dans une image bruitée.

La réponse impulsionnelle a été optimisée sous trois contraintes. Le filtre doit donner une réponse maximale pour le saut d'une fonction en escalier qui a été perturbée par un bruit blanc gaussien. De plus le filtre doit bien localiser les bords et ne pas donner plusieurs réponses pour un même bord.

Deriche a repris le problème en utilisant des conditions au limites différentes que Canny. La réponse impulsionnelle qu'il trouve dans le cas monodimensionnel s'écrit

$$h(x) = -cxe^{-\alpha|x|} \sin(\omega x),$$

où c , α et ω sont des paramètres réels.

A partir du filtre monodimensionnel, il déduit deux filtres séparables bidimensionnels, pour obtenir les composantes du gradient de l'image.

Filtrage récursif.

Il est souvent possible d'implémenter les filtres de réponse impulsionnelle infinie (filtre gaussien, filtre de Deriche) de façon récursive. Ceci permet de réduire le nombre d'opérations élevé que nécessite la convolution.

Détection de contours.

Dans la section précédente on a introduit les opérateurs de détection de bord. Le but de cette opération étant souvent de trouver les contours d'un objet, on cherche à obtenir des bords fermés. Or en pratique les détecteurs de bords donnent des morceaux de contours. Ceci peut être dû au bruit où à des changements d'intensité lumineuse du fond. De plus il n'y a pas que des contours fermés dans une image.

Pour obtenir des contours fermés on utilise des algorithmes de *fermeture de contours* qui, en partant de l'edge map, essayent de prolonger les bords. Le principe est le suivant :

A partir d'un point de terminaison d'un morceau de contour, on cherche les points de terminaison d'autres morceaux qui sont proches et donc candidats pour être reliés au morceau initial. Parmi ces points, on choisit celui qui semble donner le prolongement le plus «naturel» (*e.g.* droit, régulier). On relie ensuite les deux morceaux de contour par une courbe régulière.

Mis à part la complexité algorithmique, cette méthode est confrontée à beaucoup de cas ambigus ; elle ne donne des résultats satisfaisant que pour des scènes pas trop complexes.

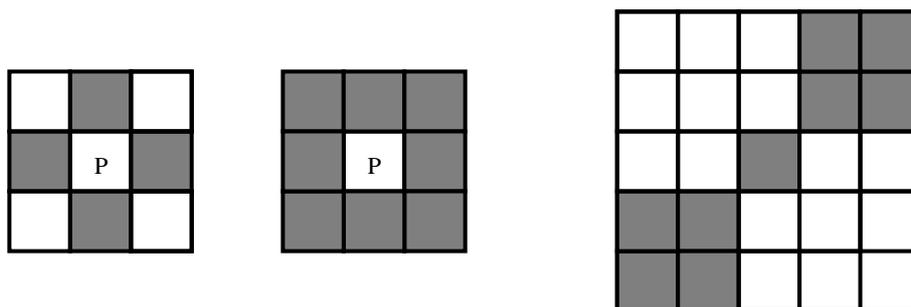
Dans le cas d'images très simples, par exemple binaires, on peut utiliser des algorithmes de *suivi de contours*. Le principe en est le suivant :

On suppose que l'on a trouvé deux pixels de couleurs différentes, c'est-à-dire une partie du bord de deux régions. On va suivre la frontière entre les deux régions de façon à ce que une des régions se trouve toujours à gauche du bord. Il suffit pour cela de faire des tests sur les pixels voisins de la position courante. Dans le cas d'une image binaire on va soit finir par retourner au point de départ, soit arriver au bord de l'image.

Remarque : Voisinage d'un pixel .

Sur une grille rectangulaire on utilise principalement deux notions de *voisinage* : le 4-voisinage, ce sont les quatre pixels voisins horizontaux et verticaux ; le 8-voisinage formé des huit pixels voisins horizontaux, verticaux et diagonaux.

Sur la figure suivante on a représenté à gauche et au centre le 4-voisinage, respectivement le 8-voisinage, du pixel P.



A droite on a représenté une configuration qui donne lieu à une ambiguïté suivant que l'on utilise les 4-voisinages ou les 8-voisinages pour le fond et/ou l'objet.

Si le fond est blanc et que l'on utilise des 8-liaisons on obtient 3 objets noirs reliés, si l'on utilise des 4-voisinages, le fond est partagée en deux par trois objets qui ne sont pas reliés. Il faut donc utiliser, par exemple, le 4-voisinages pour le fond et les 8-voisinages pour les objets. Si le noir devient le fond, on trouve une situation duale.

3.1.2 Segmentation d'images

Dans la section précédente, on a abordé le problème de détection de bords et d'extraction de contours d'un objet. Dans la segmentation d'une image g , on essaye d'analyser toute la scène représentée et d'aboutir à une image simplifiée u .

On peut dire que segmenter une image c'est :

- remplacer l'original par un «cartoon»,
- découper l'image en zones homogènes, séparées par des bords et traversées par des failles,
- faire une partition de l'image suivant un critère de ressemblance donné,

On distingue différentes approches :

- Les méthodes faibles ou approche bords.
- Les méthodes fortes ou approche région (partition, classification).
- Les méthodes mixtes.

Le problème commun à toutes ces techniques est que l'on veut garder les bords nets et bien localisés, tout en régularisant l'intérieur des régions.

Exemple : Un des premiers exemples que l'on trouve dans la littérature du traitement d'images est l'algorithme de Brice et Fennema (1970).

C'est un exemple d'un algorithme de *croissance de régions*. Le critère de ressemblance de deux régions voisines dépend du saut de la fonction le long de leur bord commun.

On procède comme suit :

1. Les pixels adjacents de même intensité lumineuse sont réunis en une même région.
2. Soient O et O' deux régions adjacentes, $\partial(O, O')$ leur bord commun et $\partial_w(O, O')$ la partie «faible» du bord commun, c'est-à-dire la partie où le saut de l'intensité lumineuse est «petit» . On réunit O et O' si et seulement si

$$\frac{\ell(\partial_w(O, O'))}{\ell(\partial(O, O'))} > \beta$$

avec $0 < \beta < 1$ paramètre réel et $\ell(\cdot)$ la longueur.

On dit encore que l'on a fait un *merging*.

3. Recommencer le 2 jusqu'à ce que ne plus aucun merging soit possible.

Algorithmes et modèles mathématiques.

Un algorithme de segmentation doit permettre de comparer deux régions (ensembles de pixels) et aboutir à une représentation structurée (haut niveau) de l'image.

Le critère de ressemblance peut être très simple (moyenne des niveaux de gris), où compliqué à évaluer (ajuster une surface paramétrée).

Les structures de données sont en général assez compliquées.

On peut trouver une quantité énorme d'algorithmes de segmentation, très souvent développés pour une application et un type particulier d'images. Ces méthodes sont un mélange de détection de bords, formation de régions (par croissance et/ou découpage) et des améliorations successives par itérations.

Ces algorithmes ne sont pas basés sur des modèles théoriques et il est difficile de prévoir le résultat, respectivement de comparer les résultats, de plus, il faut souvent ajuster un grand nombre de paramètres.

En 1984, S. Geman et D. Geman ont proposé un modèle stochastique a priori, Mumford et Shah ont proposé leur modèle déterministe en 1985. Ce sont les premiers modèles mathématiques a priori pour la segmentation d'images.

Les frères Geman et Mumford et Shah définissent une énergie E sur l'image g , dans cette énergie on réunit les contraintes de ressemblance de régions, du nombre de régions etc.

Un minimum u de l'énergie est une image qui donne la meilleur approximation de g , tout en tenant compte des contraintes de l'énergie.

Grâce à cette modélisation, on pourra comparer deux segmentations :

u_1 sera meilleure que u_2 si $E(u_1) < E(u_2)$.

En fait beaucoup d'algorithmes par croissance de régions «classiques» ont une énergie sous-jacente, donnons quelques exemples.

Exemples :

1. L'algorithme de Brice et Fennema est basé sur une minimisation d'une énergie.

Leur critère de ressemblance s'écrit $\frac{\ell(\partial_w(O, O'))}{\ell(\partial(O, O'))} > \beta$

ou, plus précisément,

$$\beta \cdot \ell(\partial(O, O')) - \ell \left\{ x \in \partial(O, O') \middle/ \left| \frac{\partial g}{\partial n} \right| < \delta \right\} < 0$$

Si l'on note K l'ensemble des bords des régions O_i , on a

$$\begin{aligned} E((O_i)_i) &= - \sum_{i,j} \beta \ell(\partial(O_i, O_j)) \\ &\quad + \sum_{i,j} \ell \left\{ (x, y) \in \partial(O_i, O_j) \middle/ \left| \frac{\partial g}{\partial n} \right| < \delta \right\} \\ &= -2\beta \ell(K) + 2 \ell \left\{ (x, y) \in K \middle/ \left| \frac{\partial g}{\partial n} \right| < \delta \right\} \\ &= 2(1 - \beta) \ell(K) - 2 \ell \left\{ (x, y) \in K \middle/ \left| \frac{\partial g}{\partial n} \right| > \delta \right\}. \end{aligned}$$

En éliminant les seuils on obtient la fonctionnelle

$$E_\lambda(K) = \lambda \cdot \ell(K) - \int_K \left| \frac{\partial g}{\partial n} \right| d\sigma.$$

Cette énergie tend à minimiser la longueur des bords, tout en maintenant les bords avec un grand saut de l'intensité lumineuse.

La fonctionnelle d'énergie modélise donc bien l'algorithme initial de Brice et Fenema. Noter qu'il reste un seul paramètre λ qui fixe l'importance relative de la longueur totale des bords et de la longueur des bords «importants».

2. Un deuxième algorithme que l'on va citer, est celui de Beaulieu et Goldberg (1980). Les auteurs se sont inspirés de techniques de taxonomie numérique : l'algorithme réunit deux régions voisines O_i et O_j si la quantité $\text{Var}(O_i \cup O_j) - \text{Var}(O_i) - \text{Var}(O_j)$ est minimale. On arrête si l'on a atteint le nombre de régions désiré.

Pour une segmentation donnée, on pose comme énergie $E((O_i)_i) = \sum_i \text{Var}(O_i)$,

qu'il faut minimiser pour $\text{Card}((O_i)_i)$ fixé.

Alors l'algorithme proposé par Beaulieu et Goldberg minimise l'énergie suivante

$$E((O_i)_i) = \sum_i \text{Var}(O_i) + \lambda \text{Card}((O_i)_i).$$

Le paramètre d'échelle λ permet d'obtenir le résultat cherché : en effet on commence par minimiser E pour λ petit, le minimum trouvé sera la donnée initiale pour minimiser E avec λ plus grand. On continue ainsi à faire croître λ , jusqu'à obtenir le nombre de régions désiré.

Remarques :

1. Le modèle du premier exemple n'utilise que des informations au bords, tandis que le deuxième modèle ne tient compte que de l'intérieur des régions. On va étudier dans la suite un modèle qui réunit ces deux informations.
2. Pour des images simples, on peut trouver les régions grâce à des seuillages des niveau de gris des objets.
3. D'autres méthodes de segmentation utilisent des techniques d'analyse de données ou probabilistes pour modéliser l'homogénéité des région.

Modèle de Mumford et Shah.

Nous allons présenter ici la version simplifiée de la fonctionnelle de Mumford et Shah. On cherche une approximation constante par morceaux de l'image initiale, pour cela posons

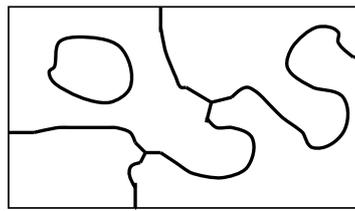
$$E(K) = \int_{\Omega \setminus K} (u - g)^2 + \lambda \ell(K),$$

où g est l'image originale définie sur Ω , u une fonction constante par morceaux et ℓ la longueur des bords K .

Minimiser cette énergie c'est trouver une image u , constante par morceaux, qui est proche de g sans avoir trop de bords.

On montre que la restriction de u , à une région homogène O , est la moyenne de g sur cette région, c'est-à-dire $u = \frac{1}{|O|} \int_O g$.

Une étude mathématique de cette fonctionnelle démontre qu'il existe au moins un minimum et que K , l'ensemble des bords de ce minimum, est formé de courbes régulières, sauf aux points d'intersection avec le bord de l'image et aux points triples. La figure suivante donne un exemple d'ensemble de bords K .



Le paramètre λ est un paramètre d'échelle :

Pour des petites valeurs on obtient un grand nombre de bords (*segmentation fine* ou *sous-segmentation*), pour des grandes valeurs on pénalise les bords et on en obtient peu de bords (*segmentation grossière* ou *sur-segmentation*).

L'exemple suivant montre que le minimum de la fonctionnelle n'est pas unique.

$g = \chi_D$		$\lambda < \lambda_0 : (u, K)_\lambda = (g, \partial D)$ $\lambda > \lambda_0 : (u, K)_\lambda = ((\int_\Omega g)/ \Omega , \emptyset)$ $\lambda = \lambda_0 : \text{deux solutions}$
--------------	--	---

On dit que K est une *segmentation 2-normale* si pour tout couple de régions adjacentes O et O' , la segmentation K' obtenue en réunissant O et O' vérifie $E(K) < E(K')$.

On démontre que le minimum de la fonctionnelle est atteint dans l'ensemble des segmentations 2-normales. Ce résultat permet d'affirmer que les algorithmes de croissance de régions donnent de « bonnes » segmentations.

Segmentation de données multicanaux.

Pour l'instant, on n'a traité que la segmentation des images en niveau de gris, c'est-à-dire $g(m, n) \in \mathbb{R}$. On peut facilement généraliser l'énergie de Mumford et Shah comme suit

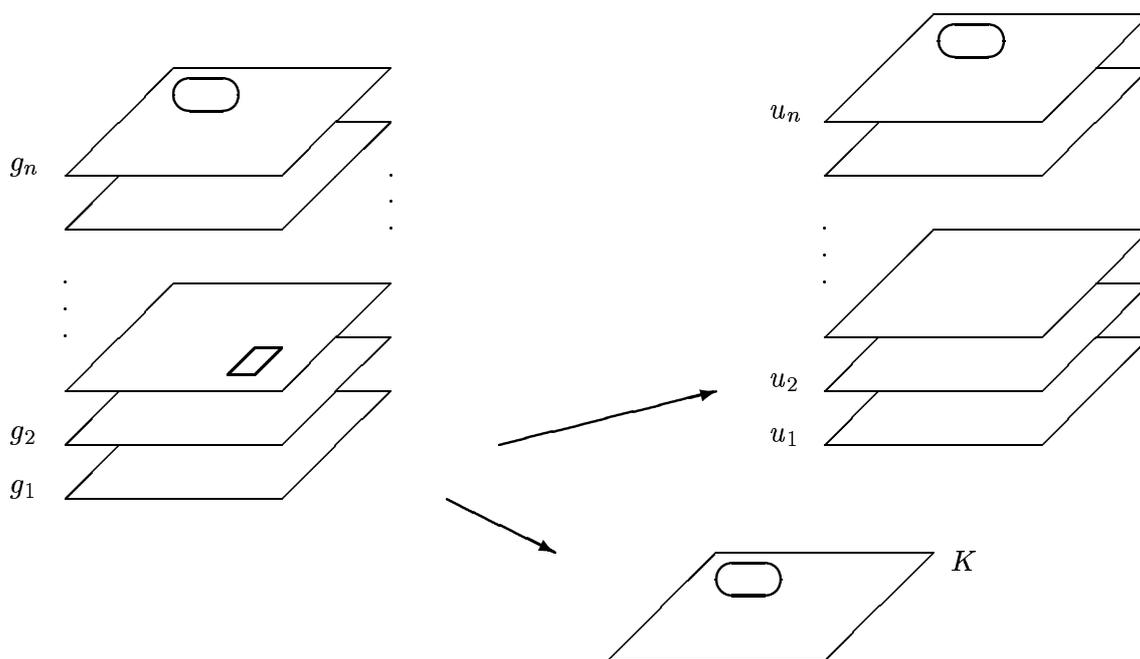
$$E(K) = \int_{\Omega \setminus K} \|u - g\|^2 + \lambda \ell(K),$$

où $g : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$ est l'image originale, multicanaux, u la fonction approximante constante par morceaux et $\ell(K)$ la longueur des bords K . On note $\|\cdot\|$ la norme que l'on utilise sur \mathbb{R}^n .

Les images multicanaux sont fréquents en imagerie satellite (couleurs, infrarouge, textures) et en imagerie médicale.

En chaque point (x, y) on a un vecteur de $\mathbb{R}^n : g(x, y) = (g_1(x, y), g_2(x, y), \dots, g_n(x, y))$, que l'on veut remplacer par $u(x, y) = (u_1(x, y), u_2(x, y), \dots, u_n(x, y))$, où g_i est une fonction scalaire et u_i la fonction approximante constante par morceaux.

Remarquer que l'ensemble des bords K est commun à tous les canaux u_i et que la norme $\|\cdot\|$ régit les interactions entre canaux. Sur la figure suivante sont schématisées les entrées et sorties d'un algorithme de segmentation multicanaux.



Pour deux régions voisines O_i et O_j , on a le critère de ressemblance suivant :

$$E(K \setminus \partial(O_i, O_j)) - E(K) < 0$$

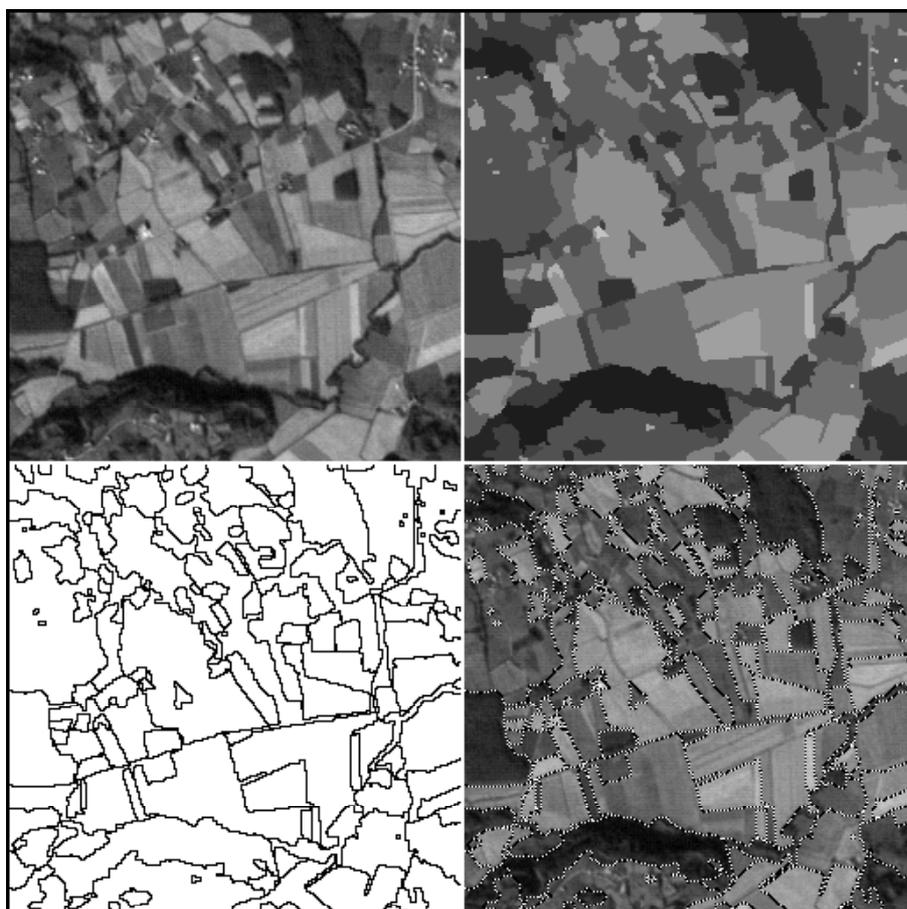
$$\frac{|O_i| \cdot |O_j|}{|O_i| + |O_j|} \cdot \|u_i - u_j\|^2 < \lambda \cdot \ell(\partial(O_i, O_j))$$

En particulier, on peut calculer la valeur de λ à partir de laquelle on fait décroître l'énergie en éliminant le bord entre O_i et O_j .

On obtient l'algorithme de segmentation par croissance de régions :

1. Soit (u_0, K_0) la segmentation triviale (*i.e.* en pixels) de l'image.
2. Pour tout couple de régions voisines, calculer la valeur de λ pour laquelle un merging est possible. Ordonner par valeurs croissantes de λ les couples de régions.
3. Effectuer le merging du couple correspondant à la plus petite valeur de λ et mettre à jour la structure de données.
4. Répéter l'étape précédente, jusqu'à obtenir le nombre de régions demandé.

Cet algorithme est multiéchelle et pyramidal, c'est-à-dire que l'on a $K_{\lambda'} \subset K_{\lambda}$ si $\lambda \leq \lambda'$.



La figure montre une image satellite g en haut à gauche, à droite l'approximation constante par morceaux u , en bas à gauche l'ensemble des bords K et à droite l'image originale avec bords superposés, on a 117 régions et $\lambda = 2048$.

3.1.3 Opérations morphologiques

La théorie de la *morphologie mathématique* a été développée par G. Matheron et J. Serra au Centre de Morphologie Mathématique à l'École des Mines de Paris. Les premiers livres publiés datent de 1975 et 1982.

Dans cette section on va étudier les opérations morphologiques sur les images numériques, c'est-à-dire définies sur \mathbb{Z}^2 , on ne va pas définir les opérateurs morphologiques continus.

En un premier temps on va considérer des images binaires et généraliser ensuite images en niveaux de gris.

Opérations morphologiques booléennes.

Une image booléenne I est une image où chaque pixel ne peut prendre que deux valeurs : vrai/faux, blanc/noir, 0/1. Un objet dans une image binaire est défini par un ensemble de pixels i -connexes ($i = 4, 6, 8$) et son complémentaire sera $(12 - i)$ -connexe. Il faut donc fixer la couleur du fond et des objets.

Dans la suite le fond sera toujours blanc (valeur 255 sur un écran en niveaux de gris) et les objets seront noirs (valeur 0).

Un objet A de l'image I est un ensemble de pixels i -connexes. L'objet A est un sous-ensemble de I , $A \subset I$. Deux objets i -connexes A et B sont disjoints si $A \cap B = \emptyset$.

Un pixel $a = (a_1, a_2)$ appartient à A , $a \in A$ ou n'appartient pas à A , il appartient alors au complémentaire A^c , La théorie des opérateurs morphologiques booléens utilise donc la théorie et le vocabulaire de la théorie des ensembles.

Sur les objets d'une image binaire on définit les opérations de base suivantes :

- La *translation* d'un ensemble, on considère un vecteur $v = (v_1, v_2) \in \mathbb{Z}^2$ et un ensemble A , alors le translaté, $(A)_v$, est défini par

$$(A)_v = \{a + v \mid a \in A\} = \{(a_1 + v_1, a_2 + v_2) \mid (a_1, a_2) \in A\}.$$

On définit le symétrique de A par $\check{A} = \{(-a_1, -a_2) \mid a = (a_1, a_2) \in A\}$.

- L'*addition de Minkowski*, soient deux ensembles A et B , on pose

$$A \oplus B = \bigcup_{b \in B} (A)_b.$$

Note : un élément de B est considéré ici comme étant un vecteur.

- La *soustraction de Minkowski*, soient deux ensembles A et B , on pose

$$A \ominus B = \bigcap_{b \in B} (A)_b.$$

Remarque :

1. Les opérations ainsi définies agissent sur les ensembles de pixels et non pas sur les valeurs de ces pixels. On verra que les opérations morphologiques sur les images en niveaux de gris nécessitent des opérations sur des ensembles et les valeurs de gris de ces pixels.
2. Les images binaires peuvent servir à représenter des scènes «plates», mais posent problème pour des scènes 3D. En effet, on n'a aucun moyen de résoudre les cas d'occultations : si $A \cap B \neq \emptyset$, est-ce que A est devant B ou vice-versa ?

Grâce aux définitions précédentes on va définir les opérations fondamentales de la morphologie mathématique. On introduit :

- La *dilatation* :

$$\mathcal{D}(A, B) = A \oplus B = \bigcup_{b \in B} (A)_b = \{p \in I / p = a + b, a \in A \text{ et } b \in B\}.$$

- L'*érosion* :

$$\mathcal{E}(A, B) = A \ominus \check{B} = \bigcap_{b \in B} (A)_{-b} = \{p \in I / \forall b \in B : p + b \in A\} = \{p \in I / (B)_p \subset A\}.$$

Dans ces définitions A et B sont considérés comme objets contenus dans l'image I , mais en général on distingue les deux et l'on appelle B l'*élément structurant*.

On dit que l'objet A est analysé par l'élément structurant B .

Note : On peut remarquer ici une analogie avec la théorie des filtres linéaires, l'élément structurant prenant la place de la réponse impulsionnelle. De plus la dilatation et l'érosion peuvent s'interpréter comme convolutions dans les algèbres de Boole.

Les opérations de dilatation et d'érosion vérifient un grand nombre de propriétés que nous avons regroupé dans le tableaux suivant.

Commutativité, symétrie	$\mathcal{D}(A, B) = A \oplus B = B \oplus A = \mathcal{D}(B, A)$
Non symétrie	$\mathcal{E}(A, B) \neq \mathcal{E}(B, A)$
Invariance par translation	$\mathcal{D}(A, (B)_v) = \mathcal{D}(A, B)_v = \mathcal{D}((A)_v, B)$ $\mathcal{E}(A, (B)_v) = \mathcal{E}(A, B)_v = \mathcal{E}((A)_v, B)$
Non inverses	$\mathcal{D}(\mathcal{E}(A, B), B) \neq A \neq \mathcal{E}(\mathcal{D}(A, B), B)$
Extensivité	Si $(0, 0) \in B$, alors $A \subset \mathcal{D}(A, B)$
Anti-extensivité	Si $(0, 0) \in B$, alors $\mathcal{E}(A, B) \subset A$
Croissance en "A"	Si $A_1 \subset A_2$: $\mathcal{D}(A_1, B) \subset \mathcal{D}(A_2, B)$
Croissance en "B"	Si $A_1 \subset A_2$: $\mathcal{E}(A_1, B) \subset \mathcal{E}(A_2, B)$
Décroissance en "B"	Si $B_1 \subset B_2$: $\mathcal{D}(A, B_1) \subset \mathcal{D}(A, B_2)$
Décomposition	Si $B_1 \subset B_2$: $\mathcal{E}(A, B_1) \supset \mathcal{E}(A, B_2)$
Décomposition	$\mathcal{D}(A, B \cup C) = \mathcal{D}(A, B) \cup \mathcal{D}(A, C)$ $\mathcal{E}(A, B \cup C) = \mathcal{E}(A, B) \cap \mathcal{E}(A, C)$ $\mathcal{D}(\mathcal{D}(A, B), C) = \mathcal{D}(A, \mathcal{D}(B, C))$ $\mathcal{E}(\mathcal{E}(A, B), C) = \mathcal{E}(A, \mathcal{D}(B, C))$
Dualité	$\mathcal{D}^c(A, B) = \mathcal{E}(A^c, \check{B})$ $\mathcal{E}^c(A, B) = \mathcal{D}(A^c, \check{B})$

Si l'élément structurant B est symétrique, *i.e.* $B = \check{B}$, alors les relations de dualité se simplifient et on peut par exemple obtenir l'érosion de A par B en calculant d'abord la dilatation de A^c par B et en prenant ensuite le complémentaire du résultat.

Les relations de décomposition permettent une implémentation efficace des filtres morphologiques. Un autre résultat de décomposition est dû à L. Vincent (1991). On suppose que l'élément structurant est convexe, borné et contient $(0, 0)$, alors

$$\mathcal{D}(A, B) = A \cup \mathcal{D}(\partial A, B),$$

où ∂A est le i -bord de A , c'est-à-dire l'ensemble des pixels $p \in A$ qui ont au moins un pixel du fond comme i -voisin.

Dans ce cas on n'a pas besoin de considérer tous les éléments de A pour calculer une dilatation et, par dualité, une érosion. La complexité des algorithmes est considérablement réduite, de $O(N^2)$ à $O(N)$ pour une image binaire $N \times N$.

En composant les opérations de dilatation et érosion on obtient deux opérations fréquentes en analyse morphologique classique d'images :

- L'ouverture :

$$\mathcal{O}(A, B) = \mathcal{D}(\mathcal{E}(A, B), B) = \bigcup_{\{v/(B)_v \subset A\}} (B)_v.$$

- La fermeture :

$$\mathcal{F}(A, B) = \mathcal{E}(\mathcal{D}(A, B), B) = \bigcap_{\{v/(\check{B})_v \cap A = \emptyset\}} (\check{B})_v^c = \{p \in I / (B)_p \subset A \oplus B\}.$$

On démontre les propriétés suivantes :

Dualité	$\mathcal{O}^c(A, B) = \mathcal{F}(A^c, \check{B})$ $\mathcal{F}^c(A, B) = \mathcal{O}(A^c, \check{B})$
Invariance par translation	$\mathcal{O}((A)_v, B) = (\mathcal{O}(A, B))_v$ $\mathcal{F}((A)_v, B) = (\mathcal{F}(A, B))_v$
Anti-extensivité	$\mathcal{O}(A, B) \subset A$
Extensivité	$A \subset \mathcal{F}(A, B)$
Croissance en "A"	Si $A_1 \subset A_2 : \mathcal{O}(A_1, B) \subset \mathcal{O}(A_2, B)$ Si $A_1 \subset A_2 : \mathcal{F}(A_1, B) \subset \mathcal{F}(A_2, B)$
Idempotence	$\mathcal{O}(\mathcal{O}(A, B), B) = \mathcal{O}(A, B)$ $\mathcal{F}(\mathcal{F}(A, B), B) = \mathcal{F}(A, B)$

Applications :

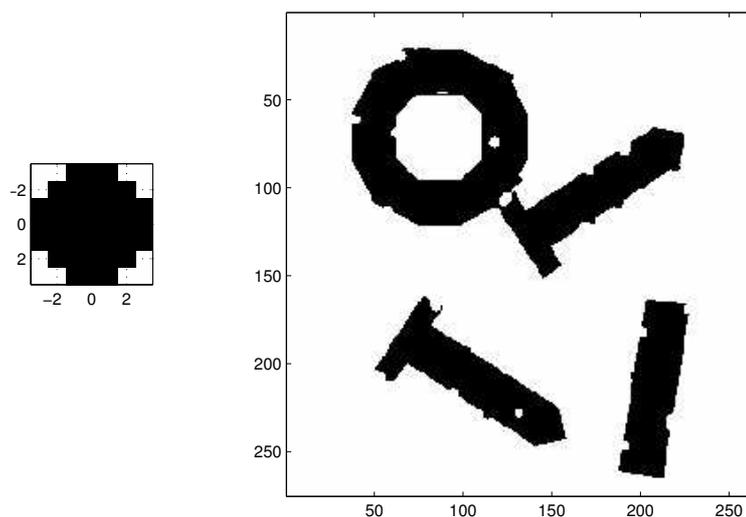
La dilatation agrandit en général un objet, l'érosion le réduit. La façon dont l'objet change, ainsi que l'importance du changement, dépend du choix de l'élément structurant.

L'opération d'ouverture peut séparer des objets qui sont connectés dans une image binaire. La fermeture peut remplir des petits trous à l'intérieur de l'objet. Les deux opérations permettent de rendre les contours plus réguliers, de l'intérieur pour l'ouverture et de l'extérieur pour la fermeture.

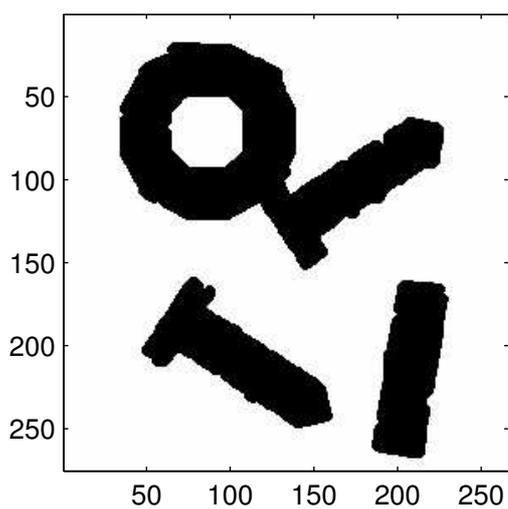
Les exemples suivants illustrent le comportement de ces opérateurs pour un élément structurant B donné (voir figure). On a choisi un ensemble convexe, borné et symétrique de diamètre 7 pixels. L'image originale est binaire et composée de formes simples mais bruitées. Ce bruit peut être dû au capteurs ou à de mauvais paramètres de seuillage lors de la numérisation.

Pour obtenir une ouverture, resp. fermeture, plus régulière on a appliqué l'élément structurant $2B = B \oplus B$.

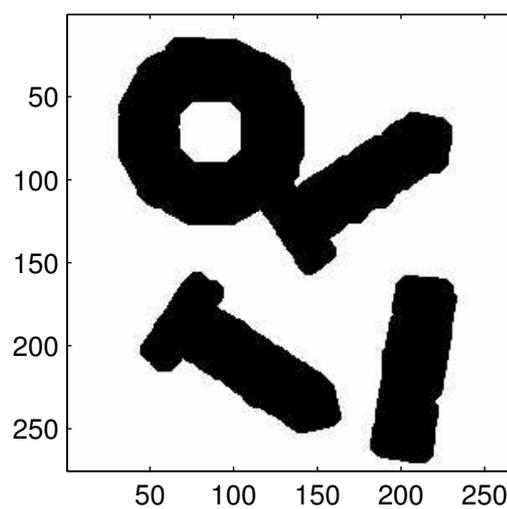
On peut extraire les 8-contours en utilisant $\partial_8(A) = A \setminus \mathcal{E}(A, N_4)$, où N_4 est l'ensemble composé par le pixel $(0, 0)$ et ses quatre 4-voisins. Pour caractériser les objets on applique une érosion sélective, en itérant on obtient le *thinning* de l'objet. On s'arrête sur une structure unidimensionnelle, le *squelette*.



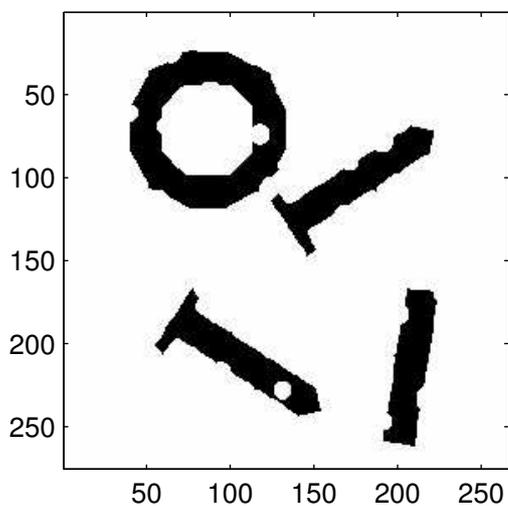
À gauche l'élément structurant B et à droite l'image initiale I .



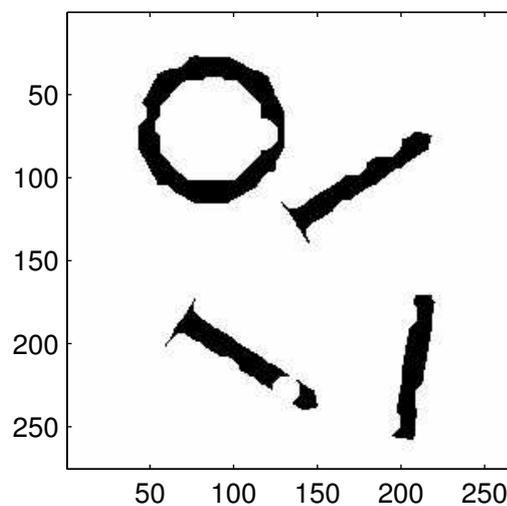
$\mathcal{D}(I, B)$.



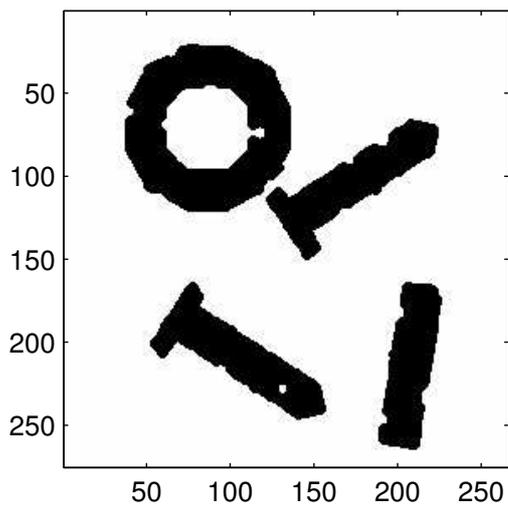
$\mathcal{D}(\mathcal{D}(I, B), B) = \mathcal{D}(I, 2B)$,
avec $2B = B \oplus B$.



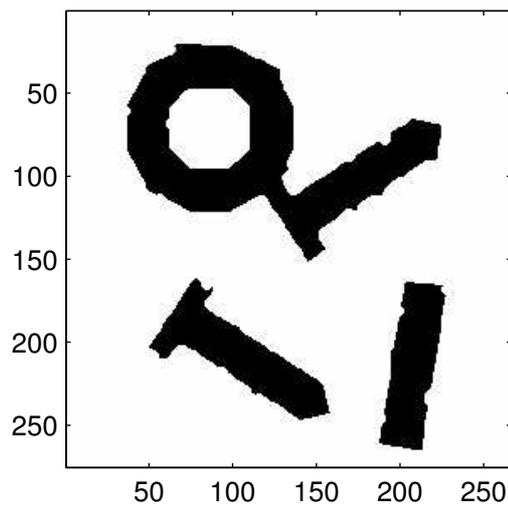
$\mathcal{E}(I, B)$



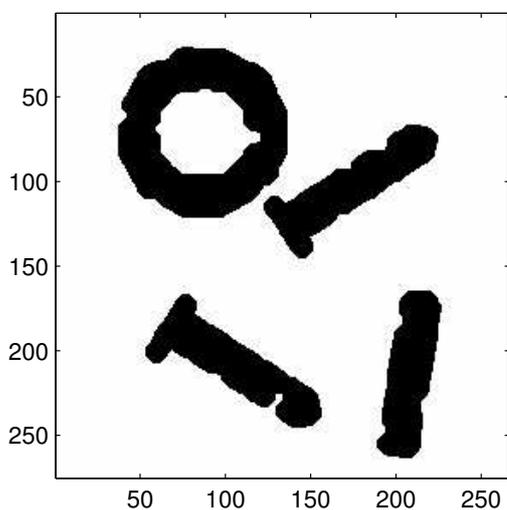
$\mathcal{E}(\mathcal{E}(I, B), B)$



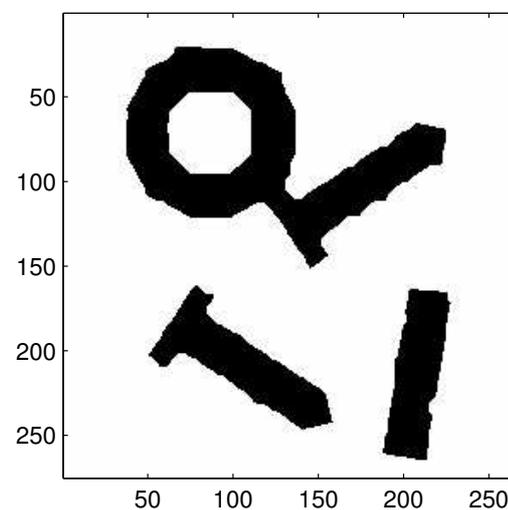
$\mathcal{O}(I, B).$



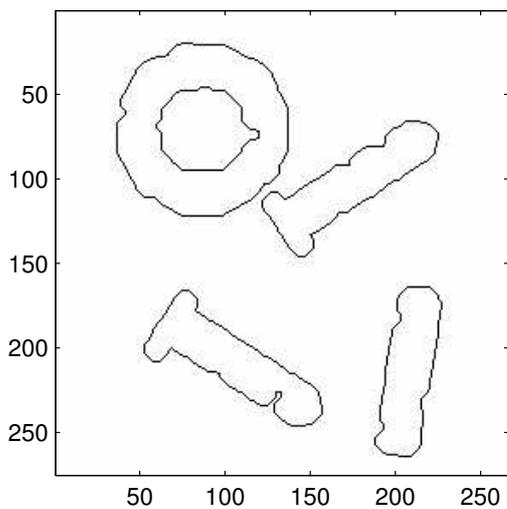
$\mathcal{F}(I, B).$



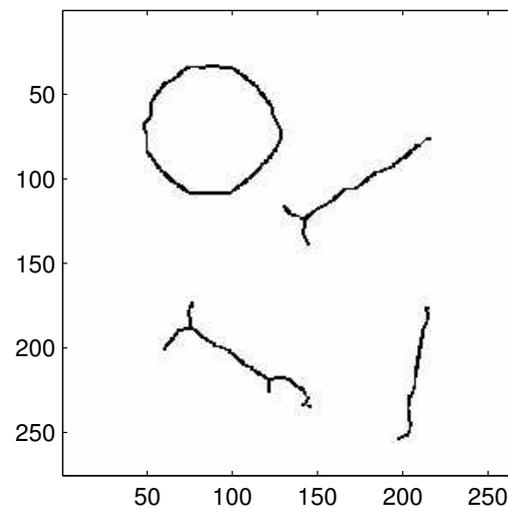
$\mathcal{O}(I, 2B).$



$\mathcal{F}(I, 2B).$



8-contour de $\mathcal{O}(I, 2B).$



Thinning de $\mathcal{O}(I, 2B).$

Opérations morphologiques sur des images à niveaux de gris.

Les filtres morphologiques peuvent être généralisés aux images en niveaux de gris. Il faut prendre en compte les niveaux de gris des pixels qui composent les objets et l'élément structurant. Le cadre choisi sera donc celui des fonctions sur \mathbb{Z}^2 .

Comme pour les signaux bidimensionnels, étudiés dans les premiers chapitres, on va considérer l'image I comme une fonction $I : \Omega \rightarrow G$, avec $\Omega \subset \mathbb{Z}^2$ et G l'ensemble des niveaux de gris. L'élément structurant B est une fonction de $\text{Dom}(B) \subset \mathbb{Z}^2$ dans \mathbb{R} , souvent on identifie le signal B et son domaine de définition $\text{Dom}(B)$. Ainsi y est un pixel de $\text{Dom}(B)$, on va écrire $y \in B$ et la valeur au pixel sera notée $B(y)$.

On définit le translaté de I par $y \in B$ pour tout $x \in \Omega$ par $\tau_y I(x) = I(x - y)$.

Note : Ceci est cohérent avec la translation d'un signal bidimensionnel introduite dans le premier chapitre.

On définit :

- La *dilatation* de I par B au pixel x :

$$\mathcal{D}_g(I, B)(x) = \max_{y \in B} \{I(x - y) + B(y)\}.$$

- L'*érosion* de I par B au pixel x :

$$\mathcal{E}_g(I, B)(x) = \min_{y \in B} \{I(x + y) - B(y)\}.$$

- L'*ouverture* de I par B au pixel x :

$$\mathcal{O}_g(I, B)(x) = \mathcal{D}_g(\mathcal{E}_g(I, B), B)(x).$$

- La *fermeture* de I par B au pixel x :

$$\mathcal{F}_g(I, B)(x) = \mathcal{E}_g(\mathcal{D}_g(I, B), B)(x).$$

La dualité des opérations s'écrit comme suit :

$$\mathcal{E}_g(I, B) = -\mathcal{D}_g(-I, \check{B}), \quad \mathcal{F}_g(I, B) = -\mathcal{O}_g(-I, \check{B}), \quad \text{où } \check{B}(x) = B(-x).$$

Note : Si les translations par les éléments de B dépassent le domaine de définition de l'image I on impose des conditions aux bords (symétrie, périodicité, ...).

Remarque : Les opérations d'intersection (\cap) et de réunion (\cup) sont remplacées par les opérations de minimum (\min) et de maximum (\max) sur les valeurs des pixels. La relation d'ordre inclusion (\subset) est remplacée par la relation d'ordre inférieur ou égal (\leq).

En général on utilise des éléments structurants très particuliers. On va supposer qu'ils sont symétriques ($B(x) = \check{B}(x)$), convexes et bornés. Un cas très important est celui où l'on suppose que B est constant et égal à zéro sur son ensemble de définition. Avec ces hypothèses les définitions des opérateurs de base deviennent :

- *Dilatation*

$$\mathcal{D}_g(I, B)(x) = \max_{y \in B} \{I(x - y)\}.$$

- *Érosion*

$$\mathcal{E}_g(I, B)(x) = \min_{y \in B} \{I(x - y)\}.$$

Les propriétés d'invariance par translation, d'idempotence, de croissance en "A", etc, obtenus dans le cadre booléen, peuvent se démontrer aussi pour les opérations morphologiques en niveaux de gris. On a :

$\mathcal{O}_g(I, B)(x) \leq I(x) \leq \mathcal{F}_g(I, B)(x)$
$\mathcal{O}_g(\mathcal{O}_g(I, B), B)(x) = \mathcal{O}_g(I, B)(x)$
$\mathcal{F}_g(\mathcal{F}_g(I, B), B)(x) = \mathcal{F}_g(I, B)(x)$

Applications :

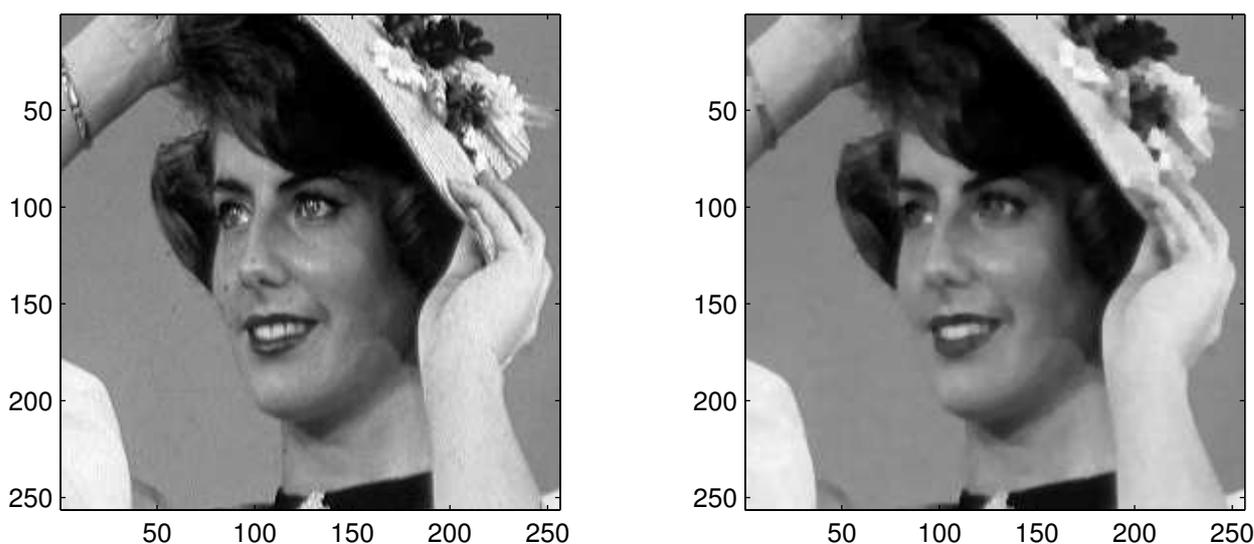
On peut montrer que l'ouverture en niveaux de gris opère sur les maxima locaux de I , c'est-à-dire que l'on régularise par le haut le graphe de $I(x)$, tandis que la fermeture en niveaux de gris opère sur les minima locaux de I , on régularise par le bas le graphe de I .

On peut donc appliquer un filtre de régularisation, que l'on définit par exemple par $\mathcal{F}_g(\mathcal{O}_g(I, B), B)$.

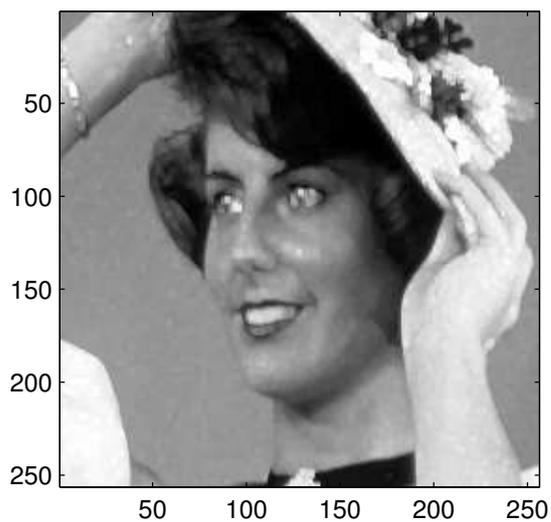
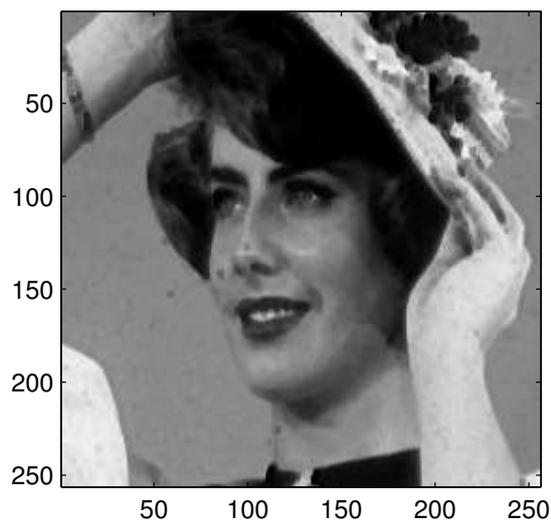
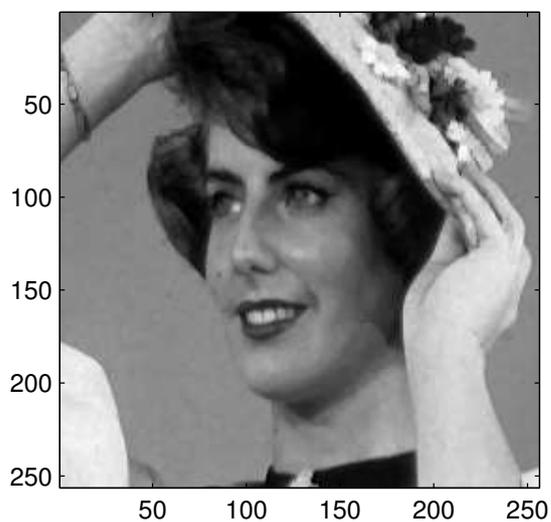
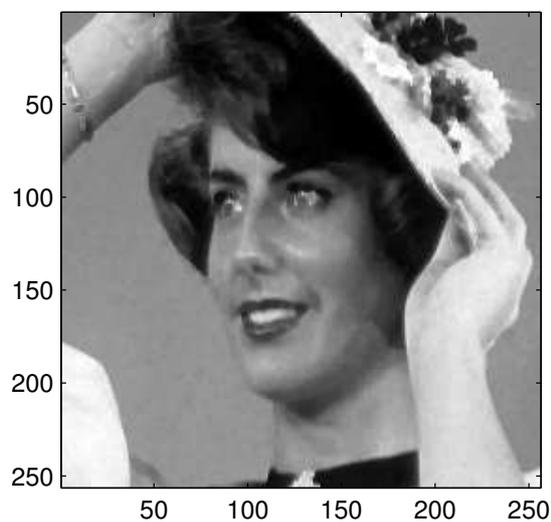
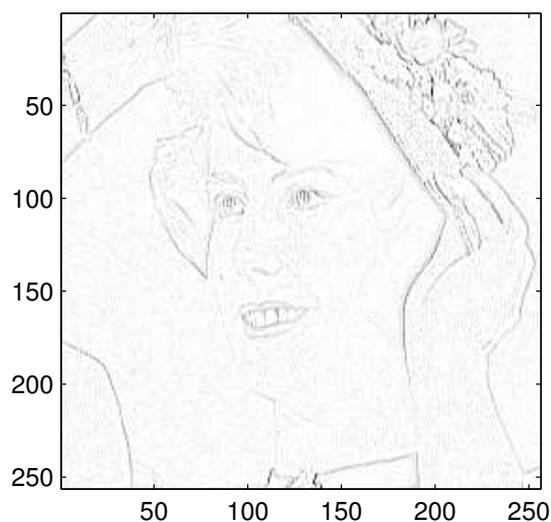
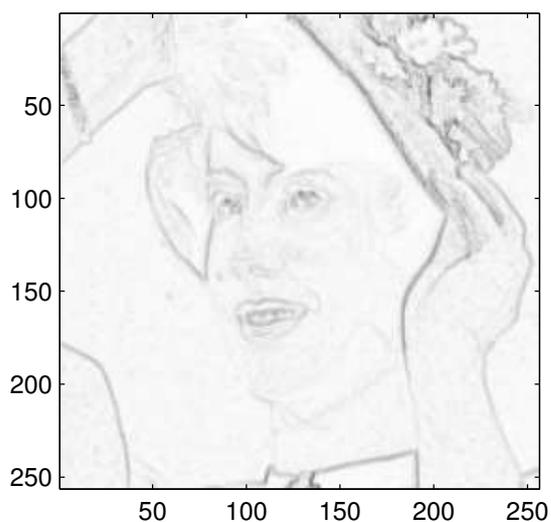
Le gradient morphologique est défini par $\frac{1}{2}(\mathcal{D}_g(I, B) - \mathcal{E}_g(I, B))$.

Le laplacien morphologique est donné par $\frac{1}{2}((\mathcal{D}_g(I, B) - I) - (I - \mathcal{E}_g(I, B)))$.

Dans les exemples suivants on a utilisé comme élément structurant la fonction $B_8(i, j) = 0$ pour $-1 \leq i, j \leq 1$, c'est-à-dire le 8-voisinage de l'origine.



À gauche l'image originale et à droite une régularisation avec $\mathcal{F}_g(\mathcal{O}_g(I, B_8))$.

 $\mathcal{D}_g(I, B_8)$. $\mathcal{E}_g(I, B_8)$. $\mathcal{O}_g(I, B_8)$. $\mathcal{F}_g(I, B_8)$.

À gauche le gradient et à droite le Laplacien morphologiques (avec inversion vidéo).

Le filtre médian.

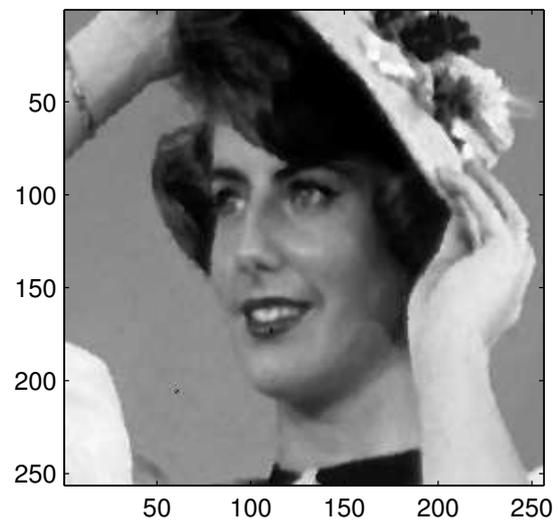
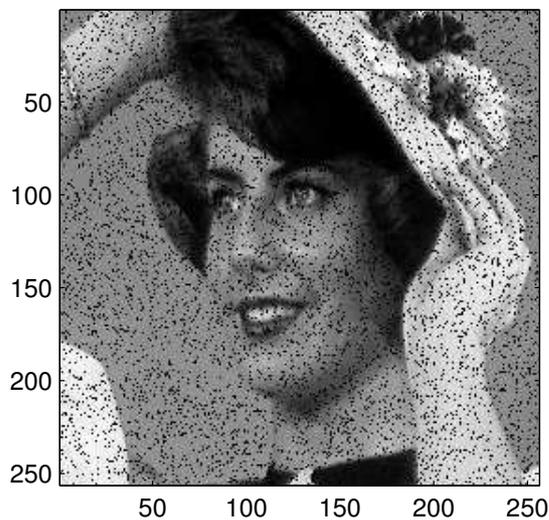
Un exemple important de filtre morphologique est le filtre *médian*, c'est un filtre non linéaire.

Soit $B \subset \mathbb{Z}^2$ symétrique, on définit le médian de I par

$$\text{med}(I, B)(x) = \text{med}_{y \in B} (I(x - y)) ,$$

où la médiane d'un ensemble discret $\{a_1, a_2, \dots, a_n\}$ est l'élément obtenu comme suit : on ordonne les (a_i) en (\tilde{a}_i) tel que $\tilde{a}_1 \leq \tilde{a}_2 \leq \dots \leq \tilde{a}_n$, ensuite on prend $\text{med}_{1 \leq i \leq n} (a_i) = \tilde{a}_{\lfloor \frac{n}{2} \rfloor}$.

Dans l'exemple suivant B est un disque de rayon 2 dans \mathbb{Z}^2 , c'est-à-dire $B = \{(i, j) \in \mathbb{Z}^2 / i^2 + j^2 \leq 2^2\}$.



À gauche l'image bruitée et à droite l'image après application du filtre médian .

3.2 Amélioration de l'image.

Dans cette section, nous allons présenter deux approches différentes pour augmenter la qualité d'une image numérique.

Il peut y avoir des dégradations suite à des problèmes de transmission, de stockage ou d'acquisition d'une image. Pour rétablir la qualité de l'image on fait une *restauration*.

Si l'on ne modifie que l'apparence d'une image, pour pouvoir extraire plus facilement l'information désirée, on fait un *rehaussement* de l'image.

3.2.1 Rehaussement d'images

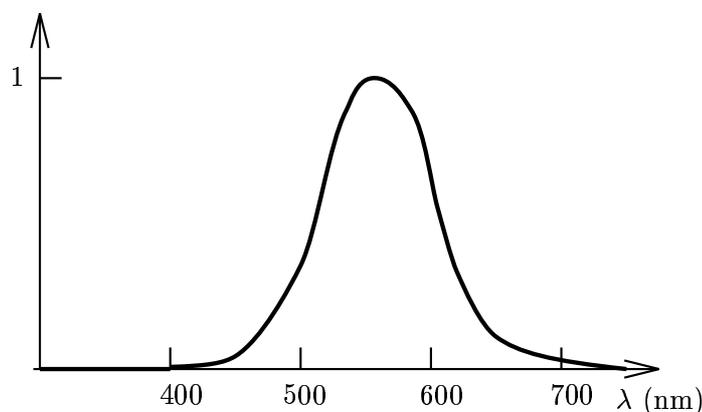
Gamma-correction.

Pour expliquer en quoi consiste la gamma-correction, il faut donner quelques définitions de grandeurs physiques intervenant dans la vision.

L'*intensité lumineuse* est une mesure physique de la puissance d'une source de radiations.

La *luminance* est une pondération de la puissance de radiations par une fonction de sensibilité spectrale, caractéristique de la vision. Cette fonction a été définie par la Commission Internationale de l'Éclairage (CIE). La luminance est noté Y (unité cd/m^2).

La figure suivante représente la fonction de sensibilité de l'oeil humain. Le spectre visible s'étale d'environ 350 nm à 780 nm, la réponse maximale à une lumière monochromatique est obtenue pour une longueur d'onde de $\lambda = 555$ nm.



Souvent on normalise la luminance par rapport à un blanc de référence de luminance Y_n .

La luminance apparente ou *luminosité*, est la réponse perceptuelle, non linéaire, à la luminance (effective) : une source ayant une luminance de seulement 18% de celle d'une source référence, paraît deux fois moins lumineuse. La luminosité a été définie par la CIE comme

$$L^* = 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \quad \text{si } 0,008856 < \frac{Y}{Y_n}.$$

Ainsi, pour $Y/Y_n = 0,18$, on a $L^* = 49,5$ et pour $Y/Y_n = 1$ on trouve bien $L^* = 100$.

Remarque : La luminosité L est grossièrement logarithmique sur un large intervalle et on a la *loi de Weber-Fechner* : $\frac{\Delta L}{L} = c_W$, avec $0,01 \leq c_W \leq 0,02$. On peut détecter une différence d'intensité ΔL si la variation relative est de plus de 0,01.

En intégrant cette relation, certains auteurs proposent une loi logarithmique pour la luminosité, néanmoins la loi puissance 1/3, retenue par la CIE, est plus proche des résultats expérimentaux.

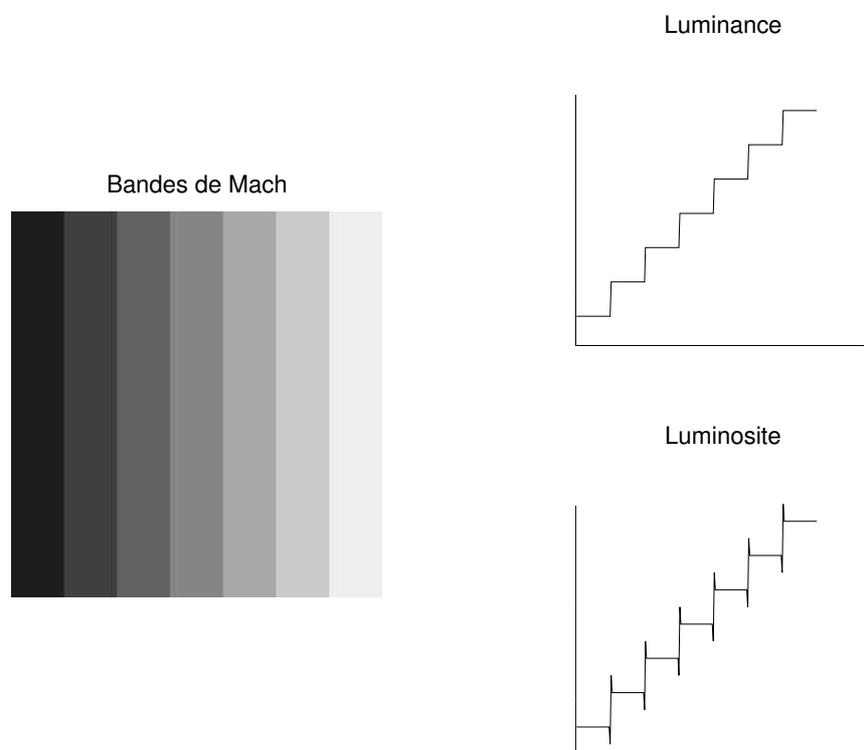
L'intensité émise par un appareil physique n'est en général pas une fonction linéaire du signal en entrée. Pour un écran conventionnel on obtient une intensité de l'affichage qui est proportionnelle au voltage appliqué à la puissance γ , avec $2,35 \leq \gamma \leq 2,55$.

Cette fonction est approximativement l'inverse de la courbe caractéristique de la perception humaine. La *gamma correction* consiste à adapter l'exposant au matériel d'affichage utilisé, ceci de façon à obtenir la meilleure représentation de l'image pour la vision humaine.

Souvent les données sauvegardées dans un format standard (TIFF, GIF, JPEG, PPM, ...) ne correspondent pas nécessairement à l'intensité lumineuse, pour les rendre «visibles» à l'oeil humain, il faut appliquer une correction gamma. Ceci est en particulier le cas si l'image est codée avec les standards vidéo (voir C.A. Poynton).

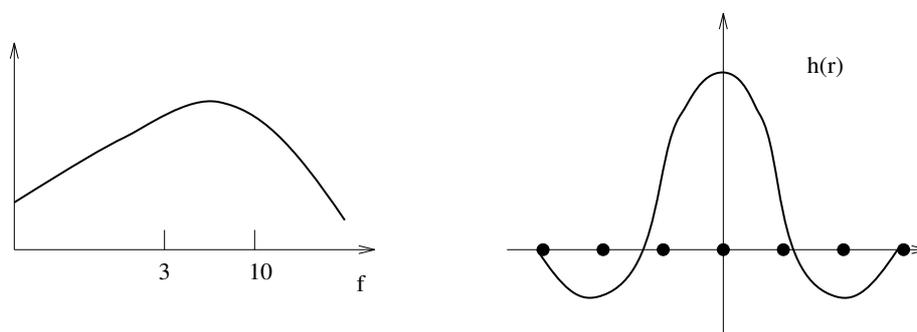
Phénomène de Mach.

L'interaction spatiale de la luminance donne lieu aux phénomènes de *bandes de Mach*, illustré sur la figure suivante.



Dans chaque bande la luminance (effective) est constante, mais la luminosité (luminance apparente) varie en fonction de la distance aux bandes voisines. C'est la réponse fréquentielle (spatiale) de l'oeil qui explique ce phénomène.

Sur la figure suivante on a représenté (à gauche) la forme typique de la sensibilité relative à la luminosité en fonction de la fréquence f (en cycles par degré). Le maximum varie suivant la personne et se trouve entre 3 et 10 cycles/deg.



La courbe à droite représente la réponse impulsionnelle pour le signal post-rétinal. Cette courbe tient compte du fait que le signal neuronal, provenant d'un endroit donné de la rétine, est une superposition de l'information des récepteurs, cônes et bâtonnets, voisins. En abscisse on représente les récepteurs voisins, en ordonnée les poids de leur contribution à la réponse neuronale. Les parties négatives correspondent à l'*inhibition latérale*, c'est-à-dire au fait que le signal post-rétinal, en un endroit donné, a été inhibé par des récepteurs voisins.

La réponse fréquentielle spatiale de l'oeil dépend de ses caractéristiques optiques, de la sensibilité non linéaire aux variations de luminosité ($Y^{1/3}$) et des mécanismes neuronaux (inhibition latérale). Ce modèle est en général admis comme étant une bonne approximation du comportement de l'oeil.

Pour des fréquences inférieures à 3 cycles par degré on a un *comportement différenciateur* de l'oeil. C'est le phénomène de Mach.

On profite de ce phénomène dans la technique du cache filtré pour augmenter le contraste d'une image. À l'image originale on ajoute une version filtrée passe-haut, obtenue par exemple en appliquant le laplacien à l'image initiale. On accentue ainsi les hautes fréquences spatiales, ce qui augmente le contraste.

Si l'on a des fréquences supérieures à 10 cycles par degré on a un *effet d'intégration* : l'oeil n'arrive pas à distinguer des structures dont les éléments sont à une dimension qui correspond à une fréquence spatiale élevée.

Dans l'imprimerie, on utilise ce phénomène pour obtenir des textures différentes à partir de deux couleurs (p.ex. n/b). De même un écran d'ordinateur ne possédant que 256 niveaux de gris, mais ayant une bonne résolution, permet d'afficher des images de grande qualité.

Modification de l'échelle de gris.

Soit $g(m, n)$ une image numérique prenant ses valeurs dans l'intervalle $[l_m, l_M]$, c'est-à-dire $l_m \leq g(m, n) \leq l_M$. L'intervalle $[l_m, l_M]$ est appelé *échelle de gris* de l'image g .

Supposons que l'on peut afficher des niveaux de gris de 0 à L . Si $0 \ll l_m$ et/ou $l_M \ll L$ la *dynamique* de g n'est pas assez grande, on va remplacer les valeurs de $g(m, n)$ de façon à utiliser tout l'intervalle $[0, L]$.

On peut utiliser n'importe quelle fonction croissante f pour transformer un niveau de gris l en $l' = f(l)$. On obtient une nouvelle image $g'(m, n) = f(g(m, n))$.

De cette façon on peut accentuer, ou atténuer, certaines portions dans l'échelle de gris de l'image initiale.

Grâce à la *quantification uniforme* on obtient un *stretching* :

$$l' = f(l) = E \left[(l - l_m) \frac{L}{l_M - l_m} + \frac{1}{2} \right],$$

où $E[\cdot]$ est la partie entière, les valeurs l de l'intervalle $[l_m, l_M]$, sont uniformément réparties sur l'intervalle $[0, L]$.

Pour le spectre d'amplitude de la TFD d'une image on utilise plutôt un changement d'échelle logarithmique $l' = c \cdot \log_{10}(1 + |l|)$. Cette transformation accentue les pixels avec de faible valeur (*e.g.* hautes fréquences), par rapport au pixels (moins nombreux) ayant une très grande valeur (*e.g.* basse fréquence, composante DC).

Si la fonction f est décroissante on obtient une inversion de l'échelle de gris, en particulier pour $l' = f(l) = 255 - l$ on trouve le négatif de l'image initiale.

Modification de l'histogramme.

L'*histogramme* h d'une image est une fonction qui associe à chaque niveau de gris $l \in \{0, \dots, L\}$, son nombre d'occurrences, $h(l)$, dans l'image. Autrement dit, $h(l)$ est le nombre de pixels qui sont au niveau l .

Une modification de l'histogramme permet de changer la distribution des niveaux de gris, par exemple pour obtenir une amélioration du contraste. Des niveaux de gris proches sont écartés et les plages de niveau de gris ayant une faible occupation remplis.

L'*égalisation d'histogramme* consiste à changer la distribution des niveaux de gris de façon à obtenir une répartition uniforme des valeurs.

Pour tout $l \in \{0, \dots, L\}$ on définit

$$l' = f(l) = E \left[\frac{L}{N} \sum_{i=0}^l h(i) \right],$$

avec $N = \sum_{i=0}^L h(i)$, c'est le nombre de pixels de l'image et $h(l)/N$ est la fréquence d'apparition de l dans l'image. La fonction de répartition des niveau de gris dans la nouvelle image, $l' \mapsto \sum_{i=0}^{l'} h(i)$, est alors (approximativement) linéaire ($\approx N/L \cdot l'$).

L'égalisation d'histogramme sur toute l'image donne souvent de mauvais résultats : des zones avec un mauvais contraste influencent des zones où le contraste est bon.

Des approches plus coûteuses en temps de calcul sont les modifications adaptatives de l'histogramme. Pour cela on fixe en chaque pixel un voisinage sur lequel on évalue l'*histogramme local* et le pixel est modifié de façon à obtenir la forme désirée de l'histogramme.

3.2.2 Lissage et restauration d'images

Dans cette section nous allons introduire quelques techniques de débruitage. En un premier temps on va présenter des techniques de lissage, ensuite des exemples de débruitage adapté au bruit.

Techniques de lissage.

Le bruit, s'il est dû à des erreurs de transmission ou à des capteurs bruités, apparaît souvent dans une image numérique sous la forme de pixels dégradés, isolés et sans corrélation spatiale. Ces pixels se démarquent de façon visible de leur voisinage par une différence du niveau de gris.

On parle de *bruit additif* si on ajoute/retranche une valeur aléatoire à des pixels choisis au hasard.

On dit que l'on a un *bruit impulsionnel* si la valeur initiale du pixel est complètement remplacé. La distribution des pixels affectés est uniforme, ainsi que l'amplitude du bruit en un pixel affecté. L'aspect du bruit impulsionnel fait que l'on parle de *salt and pepper noise*.

Filtres linéaires.

La décorrélation spatiale permet de caractériser le bruit par une fréquence spatiale plus élevée que les autres composantes de l'image. Il «suffit» donc d'appliquer des filtres passe-bas pour nettoyer l'image.

Dans le domaine spatial, le filtrage linéaire consiste en une convolution avec la réponse impulsionnelle d'un filtre passe-bas. Des exemples types de filtres passe-bas sont les masques suivants

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{ou} \quad \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{ou} \quad \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} .$$

Ces masques sont normalisés pour éviter des dépassements d'amplitude dans l'image traité.

On peut aussi considérer des masques de taille 5×5 , 7×7 ou plus. Ceci permet un lissage plus important du bruit, mais en même temps on perd des détails de l'image (voir aussi la détection des bords).

Le filtrage linéaire peut se faire aussi dans le domaine spectral. En utilisant une TFR on gagne en temps d'exécution, si le support de la réponse impulsionnelle choisie est grand.

Dans le domaine spectral on multiplie la réponse fréquentielle de l'image bruitée avec la réponse fréquentielle du filtre. Ensuite on utilise la TFR inverse pour obtenir l'image lissée.

Il faut noter qu'il n'est pas intéressant d'appliquer la réponse fréquentielle d'un filtre passe-bas idéal, la transition abrupte entre fréquences passantes et fréquences coupées introduit un effet de *ringing* (tintement, *i.e.* effet de résonnement, *phénomène de Gibbs*). On utilisera plutôt des réponses fréquentielles régulières, comme par exemple la Gaussienne.

Filtre médian.

Les techniques linéaires sont surtout efficaces pour éliminer du bruit additif, en présence de bruit impulsionnel ces filtres ont tendance à lisser trop, et à effacer les détails de l'image. Des méthodes non linéaires sont souvent plus efficaces pour lisser l'image et préserver les détails. Un exemple est le filtre médian.

Considérons un ensemble $\{a_1, a_2, \dots, a_n\}$, avec n impair. Alors la *médiane* de cet ensemble est l'élément a_i , tel que $(n-1)/2$ éléments sont inférieurs ou égal à a_i , et $(n-1)/2$ éléments sont supérieurs ou égal à a_i .

Par exemple la médiane de l'ensemble $\{4, 2, 5, 8, 0, 10, 5\}$ est 5.

En chaque pixel de l'image, on va remplacer la valeur initiale par la médiane prise dans un voisinage V du pixel. En général V sera centrée sur le pixel et de taille 3×3 ou 5×5 , mais tout voisinage connexe peut être utilisé.

La complexité de l'algorithme dépend de la taille du voisinage, c'est-à-dire du nombre de valeurs à trier.

Voir aussi le chapitre sur les opérations morphologiques.

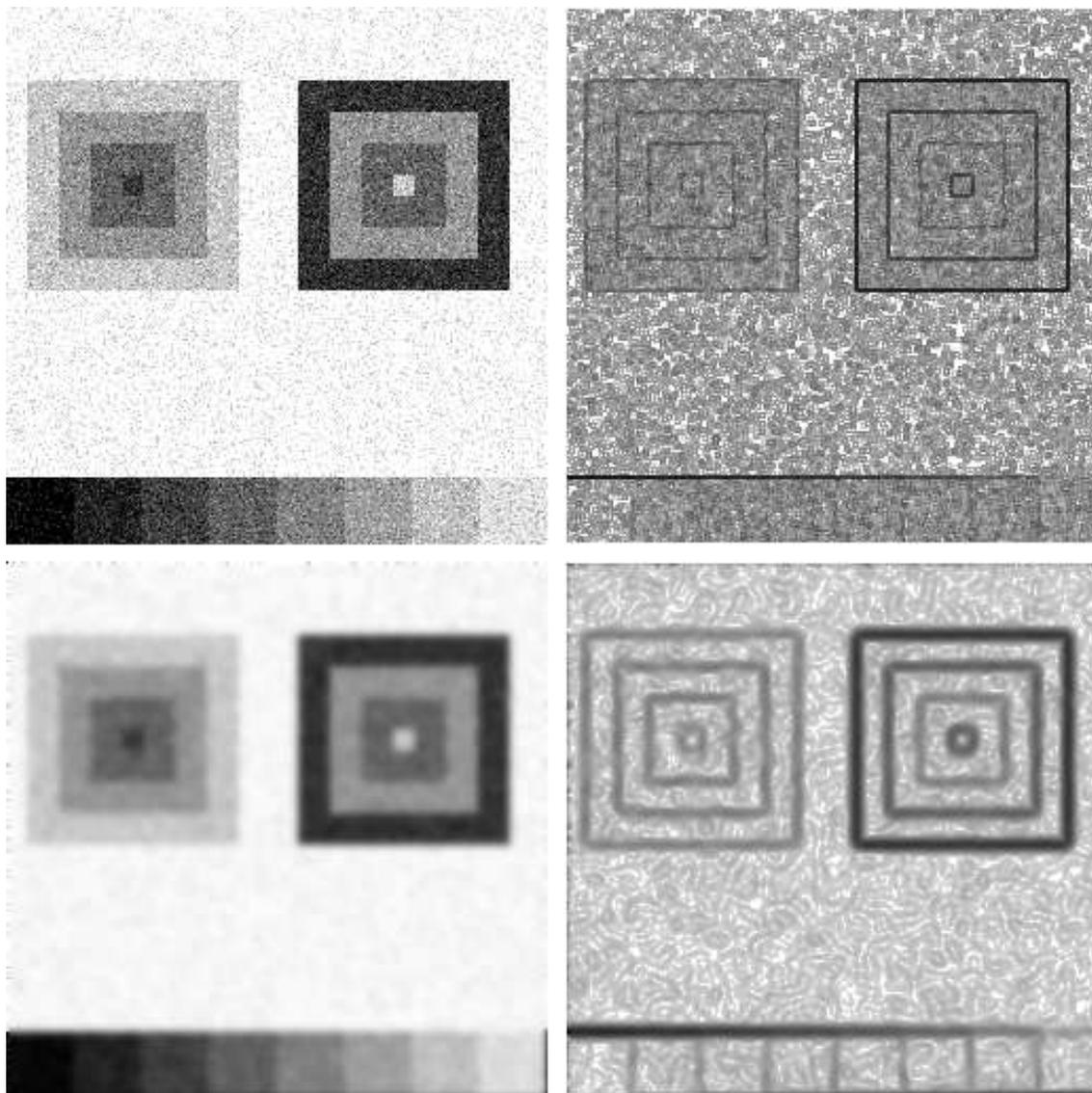
Filtres directionnels.

Le filtre gaussien est un filtre linéaire qui est souvent utilisé, car il a une localisation optimale en espace et en fréquence.

On montre que la convolution avec la Gaussienne consiste à faire évoluer l'image originale suivant l'équation de la chaleur, c'est-à-dire, on a une *diffusion* de l'image bruitée. Or cette diffusion est isotrope, en particulier elle efface les bords.

Pour éviter cela on peut appliquer une *diffusion anisotrope*, il s'agit de lisser l'image partout, sauf au-delà des bords. Cette méthode nécessite donc deux étapes. On doit d'abord détecter les bords et autres structures fines de l'image. Ensuite on va lisser l'image en ne considérant que des voisinages de pixels qui ne traversent pas les bords.

Les deux pages suivantes présentent et comparent des résultats de filtres directionnels. On considère une image u qui sera fonction des lignes et colonnes x et y et d'une variable t . Pour $t = 0$ on a l'image originale, bruitée, plus t est grand plus on a une image lisse. On considère t soit comme étant une échelle, soit comme étant le paramètre de temps.

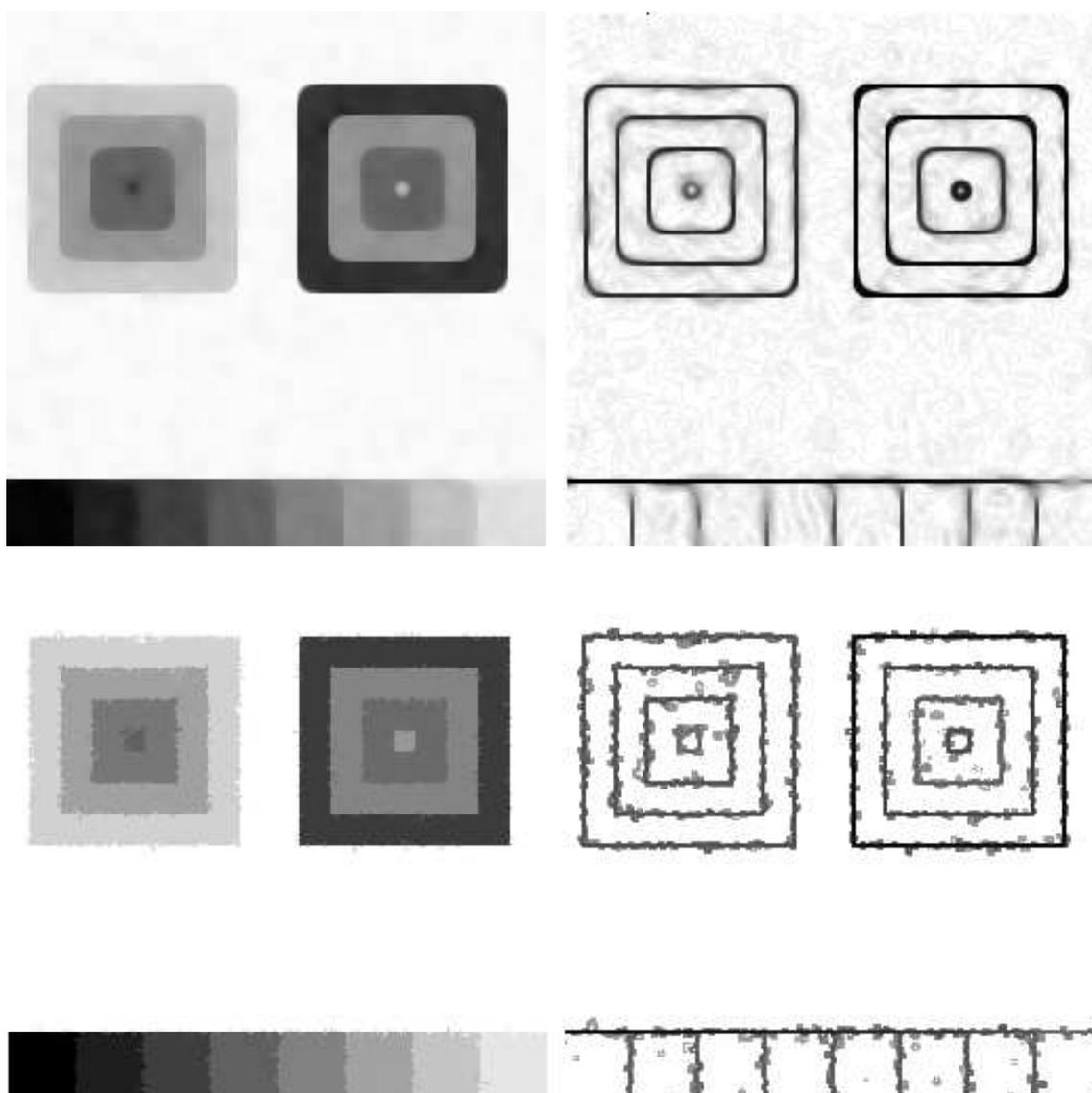


En haut à gauche une image bruitée u et en haut à droite une tentative de détection de bord par gradient discret $|\nabla u|$. Ici un gradient fort est représenté par la couleur noir (blanc=gradient nul).

En bas à gauche on a appliqué l'équation de la chaleur à u : on calcule la solution de l'équation aux dérivées partielles (edp) $\frac{\partial u}{\partial t} = \Delta u$ à l'instant t , c'est-à-dire on diffuse pendant le temps t .

Ceci est équivalent à calculer la convolution de u avec $G_{\sqrt{t}}$.

En appliquant le gradient à cette image lissée on obtient l'image de droite $|\nabla G_{\sqrt{t}} * u|$. On remarquera l'épaisseur des bords obtenus.



En haut à gauche, on a représenté un lissage obtenu par une technique non linéaire modélisée par l'edp $\frac{\partial u}{\partial t} = |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$ et dont la solution est calculé avec un schéma numérique classique.

En appliquant le gradient, on remarque que les bords sont bien localisés. Pourtant, il reste un peu de bruit et les coins sont lissés.

En bas à gauche, le résultat pour le modèle, resp. l'edp, $\frac{\partial u}{\partial t} = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$, en utilisant un algorithme non classique, basé sur les lignes de niveaux de u .

L'image du gradient montre que le bruit dans les zones homogènes a complètement disparu mais les bords sont plus irréguliers.

Sur une image réelle on obtient les résultats suivants;



En haut à gauche, l'image bruitée, en bas à gauche le résultat de la première méthode non linéaire de la page précédente.

La colonne de droite présente deux résultats obtenus par la deuxième méthode non linéaire, mais en utilisant des algorithmes différents (en haut schéma numérique, en bas algorithme «lignes de niveaux»).

Restauration d'images.

La restauration d'images utilise l'image observée et dégradée, et des informations à priori, pour reconstruire l'image originale. Il est important d'avoir des modélisations de la formation d'image et/ou des perturbations, pour pouvoir réduire les possibilités de reconstruction. Dans ce cas seulement, on pourra développer les techniques de débruitage adaptées.

La formation d'images est souvent modélisée par des systèmes linéaires et invariants. On a donc une caractérisation par la réponse impulsionnelle. Nous proposons quelques exemples.

À partir des lois de la physique on montre que la réponse impulsionnelle caractérisant les turbulences atmosphériques (variation de l'indice de réfraction), est du type

$$h_a(x, y) = e^{-\pi c^2(x^2 + y^2)}.$$

Un mouvement relatif de l'objet par rapport à la caméra pendant le temps d'exposition est modélisé par une réponse impulsionnelle h_m . Dans le cas d'un mouvement uniforme dans la direction des x on a

$$h_m(x, y) = \frac{1}{a} \text{rect}_{[0,1]} \left(\frac{x}{a} \right) \delta(y).$$

Des systèmes optiques défectueux (p.ex. une mauvaise mise au point) se comportent comme des filtres passe-bas. Ces dégradations rendent les images floues, en anglais on parle de *blur*, et de *deblurring* pour regrouper les techniques de débruitage associées.

Après le processus de formation d'images, il y a les dégradations dues au récepteurs : non linéarités des caméras ou du papier photographique.

Ces déformations donnent une image $g(x, y)$, à partir de l'image initiale $u(x, y)$, en tenant compte de bruit, on obtient le modèle général suivant pour l'image observée :

$$u_o(x, y) = g(h * u(x, y)) + n(x, y),$$

où le bruit n , peut se décomposer comme suit :

$$n(x, y) = f(g(x, y)) \cdot n_1(x, y) + n_2(x, y).$$

Dans des cas simples, n est indépendant de g et distribué aléatoirement. Mais par exemple le bruit qui est dû au grain d'un film, est modélisé par

$$n(x, y) = \alpha \cdot g(x, y)^\beta \cdot n_1(x, y),$$

avec β proche de 1/3 et n_1 un bruit blanc gaussien.

Les images sonar sont sujet à un autre type de bruit non additif, le *speckle* (*i.e.* moucheté). L'image observée est donnée par

$$u_o(x, y) = g(h * u(x, y)) \cdot s(x, y) + n_2(x, y)$$

où s est un bruit blanc de distribution exponentielle et n_2 est un bruit additif.

Ce bruit apparaît si la rugosité de la surface de l'objet est de l'ordre de grandeur de la longueur d'onde de la radiation incidente. On a à déplorer une perte de résolution dans les images.

Filtre inverse.

Si l'on suppose que $u_o = h * u + n$, et si l'on connaît h , on peut obtenir u par filtrage inverse, on détermine h_i tel que $\hat{h}_i = \hat{h}^{-1}$, pour cela il faut que \hat{h} ne s'annule pas.

En particulier, si n est nul ou négligeable, on obtient $\hat{u} = \hat{h}_i \cdot \hat{u}_o$, d'où l'on peut déduire u .

Si n est non négligeable, cette méthode donnera de mauvais résultats. Le filtrage inverse est une technique très instable.

Filtre de Wiener.

Une autre manière de restauration est de chercher une estimation de l'image u à partir de u_o , de façon à minimiser un critère donné.

Le filtre qui minimise l'erreur quadratique moyenne, au sens statistique, est le filtre de Wiener. Nous ne donnerons pas de détails ici.

3.3 Compression d'images

La compression est utilisée pour diminuer la redondance d'information qu'il y a dans une image. Les applications sont la transmission et le stockage d'images numériques.

Si l'on considère l'image numérique comme étant juste un tableau de nombres on peut lui appliquer des techniques classiques de compression sans perte. Par exemple, le codage Lempel-Ziv, utilisé par les commandes `compress` et `gzip`. Ces techniques permettent en fait de coder n'importe quel fichier, p.ex. ASCII ou binaire, sans perte d'information.

Pour diminuer plus l'information à véhiculer, on peut tirer profit de particularités dues à l'information contenue dans une image. En particulier si les images décompressées sont destinées à l'oeil humain, l'on peut utiliser les caractéristiques de la vision humaine.

Les méthodes de compression peuvent s'appliquer sur la représentation de l'image dans le domaine spatial ou sur une transformée (linéaire) de l'image (p.ex. TFD, TCD, TOD).

3.3.1 Le standard JPEG

C'est une technique de compression d'images (non animées) standardisée, développée par le *Joint Photographic Experts Group* depuis 1989. Ce groupe est formé par l'ISO (International Standards Organisation), le CCITT (Consultative Committee on International Telegraph and Telephone) et l'IEC (International Electrotechnical Commission).

Dans la suite on va donner une description du mode de compression séquentiel de JPEG. Historiquement c'est le premier à faire partie du standard, mais actuellement il y a trois autres modes : le mode sans perte, le mode progressif et le mode hiérarchique.

Le mode sans perte répond à une demande de représentation parfaite de l'image, le mode progressif permet un codage/décodage en plusieurs étapes (avec de plus en plus de précision) et est utilisé pour la transmission d'images (WWW). Dans le mode hiérarchique on sauvegarde une image à basse résolution (rapide à décoder) et une en pleine résolution, ceci permet une consultation accélérée de banques de données d'images.

L'outil principal de JPEG est la transformation en cosinus discrète (TCD) sur des blocs 8×8 , l'algorithme de la compression dans le mode séquentiel se présente pour un tel bloc de la façon suivante :

1. Préparation des données.
2. Transformée en cosinus discrète.
3. Quantification.
4. Compression (méthode de Huffman).

Le fichier résultat contient toutes les données nécessaires pour reconstruire l'image.

Préparation des données : L'image originale sera en général en couleur, avec les composantes RGB, en vue d'un meilleur taux de compression on va changer de représentation. Grâce à une transformation linéaire on va représenter l'image dans les coordonnées $YCbCr$, où Y est la luminance et Cb et Cr contiennent l'information sur la couleur.

L'oeil humain étant plus sensible aux changements de luminance, on pourra utiliser moins de précision dans la représentation des canaux Cb et Cr . On va sous-échantillonner les composantes Cb et Cr en remplaçant deux pixels voisins par leur moyenne, ce qui fait un gain de 1/3 sur le volume initial.

Pour améliorer les performances des étapes suivantes, on centre les coefficients des composantes autour de zéro, l'intervalle $[0, 255]$ sera représenté sur $[-128, 127]$.

Transformée en cosinus discrète : A chaque bloc 8×8 on va appliquer une TCD, c'est une transformation linéaire séparable, proche de la TFD (on symétrise avant de périodiser) et comme pour la TFD il existe des algorithmes rapides.

La TCD d'un bloc $u(x, y)$ ($0 \leq x, y \leq 7$) est donnée par

$$\text{TCD}(u)(k, l) = \frac{1}{4} \alpha(k) \alpha(l) \sum_{0 \leq x, y \leq 7} u(x, y) \cos\left(\frac{(2x+1)k\pi}{16}\right) \cos\left(\frac{(2y+1)l\pi}{16}\right)$$

et la transformation réciproque

$$u(x, y) = \frac{1}{4} \sum_{0 \leq k, l \leq 7} \alpha(k) \alpha(l) \text{TCD}(u)(k, l) \cos\left(\frac{(2x+1)k\pi}{16}\right) \cos\left(\frac{(2y+1)l\pi}{16}\right).$$

où $\alpha(0) = 1/\sqrt{2}$ et $\alpha(t) = 1$ si $t \neq 0$.

En théorie cette transformation est exacte, en pratique on devra compter avec une perte d'information due aux erreurs d'arrondi dans les calculs.

Le résultat de la TCD est un bloc 8×8 , contenant en haut à gauche les basses fréquences et en bas à droite les hautes fréquences. En particulier $\text{TCD}(u)(0, 0)$ est la valeur moyenne des 64 pixels, les autres coefficients sont de plus en plus petit si on s'éloigne de $(0, 0)$: la TCD concentre l'énergie sur quelques coefficients.

Quantification : Dans cette étape, chaque coefficient de la TCD est divisé par un coefficient $Q(k, l)$ et le résultat est arrondi au prochain entier. On obtient un tableau de nombres entiers donné pour chaque élément (k, l) du bloc par :

$$q(\text{TCD}(u))(k, l) = E \left[\frac{\text{TCD}(u)(k, l)}{Q(k, l)} + \frac{1}{2} \right].$$

Le tableau $Q(k, l)$ est prédéfini, en général on utilise des valeurs plus petites pour les basses fréquences, ceci afin d'obtenir une quantification plus fine. Ce tableau est associé à chaque image, il peut être choisi par l'utilisateur suivant la nature de l'image.

Pour la reconstruction il faut multiplier $q(\text{TCD}(u))(k, l)$ par $Q(k, l)$, pour obtenir une valeur approchée ($\pm 1/2$) de $\text{TCD}(u)(k, l)$.

C'est cette étape qui est responsable de la perte d'information dans JPEG, mais c'est aussi le choix judicieux du tableau $Q(k, l)$ qui permet d'obtenir de bonnes performances de compression.

Compression : Après la quantification il reste donc un bloc 8×8 qui contient des entiers. Le codage de Huffman est une technique qui permet d'optimiser la représentation des données en utilisant des codes binaires.

Le code d'un symbole (*i.e.* entier) dépend de sa fréquence d'apparition. La valeur en $(0, 0)$ est remplacée par sa différence avec la valeur du bloc précédent. Ensuite tout le bloc est parcouru en zigzag à partir de $(0, 0)$, afin de profiter des zéros pour les hautes fréquences.

Le tableau de correspondance doit être attaché au fichier final. Comme pour le tableau de quantification, le groupe JPEG a fourni quelques tables de codage standard, mais pas nécessairement optimales.

Cette brève description est censée illustrer comment un algorithme de compression utilise de nombreuses techniques différentes pour obtenir les performances souhaitées.

On peut noter que le taux de compression dépend de la qualité souhaitée pour la reconstruction. Cette qualité est mesurée dans JPEG par un nombre Q , allant de 100 (parfait) à 0 (perte totale de l'image). Des valeurs plus grandes que 75 sont en général pas nécessaires, en dessous de 50 par contre on aura une dégradation de l'image reconstruite.

L'algorithme et le standard JPEG évoluent. En particulier, dans les nouvelles versions (JPEG 2000) la TCD est remplacée par une *transformée en ondelettes discrète*.

3.3.2 Le format GIF

C'est principalement un format de stockage d'images auquel est associé un algorithme de compression sans perte (Lempel-Ziv Welch).

Le format GIF est propriété de CompuServe, l'algorithme de compression adopté est basé sur un travail de T.A. Welch, datant de 1984.

L'algorithme de Lempel-Ziv Welch utilise la répétition de chaînes de caractères. Au fur et à mesure que l'algorithme parcourt les données, il crée une table de symboles à partir d'une table initiale comportant les symboles à représenter (p.ex. les couleurs/gris). En même temps les données initiales sont codées grâce à cette table et le résultat est un fichier codé.

Pour reconstruire on n'a besoin que de ce fichier et du nombre de symboles à représenter, le décodeur reconstruit la table des symboles au fur et à mesure.

Comme le codage, resp. le décodage, se font en un seul passage, cet algorithme est bien adapté pour un décodage avec affichage immédiat.

Cette technique est moins performante que l'algorithme JPEG, mais on n'aura pas de dégradations.

Bibliographie

On donne une liste non exhaustive des livres et documents ayant servi à rédiger ce cours et utiles pour un traitement approfondi du sujet.

- A. K. Jain : *Fundamentals of Digital Image Processing*.
Prentice Hall, 1989.
- M. Kunt : *Traitement numérique des signaux*.
Coll. "Traité d'électricité, d'électronique et d'électrotechnique", Dunod, 1981.
- D. Marr : *Vision*.
Freeman 1982.
- A. V. Oppenheim, R. W. Schaffer : *Discrete-Time Signal Processing*.
Prentice-Hall Int., Inc. , 1989.
- C. A. Poynton : FAQs sur la couleur et la vidéo,
adresse WWW : <http://www.inforamp.net/~poynton>
- W. K. Pratt : *Digital Image Processing*.
Second Edition. Wiley-Interscience Publication 1991.
- W. H. Press, et al. : *Numerical Recipes in C*.
Cambridge University Press, 1991.
- A. Rosenfeld et A.C. Kak : *Digital picture processing*.
Comp. Sc. and Applied Maths., Academic Press 1982.