

# Représentation des images et ondelettes

François Malgouyres  
Département d'informatique de l'université Paris 13  
<http://www.zeus.math.univ-paris13.fr/~malgouy>  
[malgouy@math.univ-paris13.fr](mailto:malgouy@math.univ-paris13.fr)



# Table des matières

<b>Table des matières</b>	<b>2</b>
<b>1 Notions mathématiques pour le traitement d'images</b>	<b>5</b>
1.1 La création de l'image numérique . . . . .	5
1.1.1 Images numériques . . . . .	5
1.1.2 La convolution . . . . .	6
1.1.3 Le fenêtrage . . . . .	7
1.1.4 L'échantillonnage . . . . .	7
1.1.5 Le bruit . . . . .	8
1.1.6 La quantification . . . . .	8
1.1.7 Autres dégradations . . . . .	9
1.1.8 Exemple . . . . .	9
1.2 La transformée de Fourier . . . . .	13
1.2.1 D'une image analogique . . . . .	13
1.2.2 D'une suite finie . . . . .	15
1.2.3 Convolution et transformée de Fourier . . . . .	16
1.2.4 Échantillonnage et transformée de Fourier . . . . .	18
1.2.5 Bruit et transformée de Fourier . . . . .	20
1.2.6 Fourier : une base parmi d'autres . . . . .	21
1.3 Bases de cosinus et cosinus locaux . . . . .	22
1.3.1 Les cosinus . . . . .	22
1.3.2 Les cosinus locaux . . . . .	24
1.4 Bases d'ondelettes discrètes . . . . .	24
1.4.1 Objectif . . . . .	24
1.4.2 Des banques de filtres aux ondelettes, en dimension 1 . . . . .	25
1.4.3 Les ondelettes discrètes en dimension 1 . . . . .	27
1.4.4 Les ondelettes discrètes en dimension 2 . . . . .	31
1.4.5 Choisir une ondelette . . . . .	32
1.5 Les paquets d'ondelettes . . . . .	34
1.5.1 Principe des paquets d'ondelettes en dimension 1 . . . . .	34
1.5.2 Les arbres de paquets d'ondelettes . . . . .	35
1.5.3 Localisation des paquets d'ondelettes . . . . .	37
1.5.4 Les dictionnaires de paquets d'ondelettes . . . . .	39
1.5.5 Le passage aux dimensions 2 et plus . . . . .	39
1.6 D'autres bases . . . . .	39
1.7 La représentation dans un dictionnaire . . . . .	40
1.7.1 Introduction . . . . .	40
1.7.2 Préliminaires mathématiques et notations . . . . .	42
1.7.3 La minimisation $l^1$ . . . . .	45
1.7.4 Les algorithmes de type "Orthogonal Matching Pursuit" . . . . .	46

<b>2</b>	<b>La restauration d'images</b>	<b>49</b>
2.1	Le débruitage par convolution . . . . .	49
2.2	La déconvolution . . . . .	52
2.2.1	Le filtre inverse . . . . .	52
2.2.2	Le filtre de Wiener . . . . .	53
2.3	L'interpolation . . . . .	56
2.3.1	Application : Le zoom . . . . .	58
2.3.2	Application : La translation d'une image . . . . .	58
<b>3</b>	<b>Survol de problèmes et solutions du traitement des images</b>	<b>61</b>
3.1	Le recalage . . . . .	61
3.1.1	La méthode de Horn et Schunck . . . . .	61
3.1.2	Applications . . . . .	62
3.2	L'indexation par le contenu . . . . .	64
3.2.1	Les signatures d'images . . . . .	65
3.2.2	Comparaison de signatures . . . . .	66
	<b>Bibliographie</b>	<b>69</b>

# Chapitre 1

## Notions mathématiques pour le traitement d'images

Nous verrons dans ce chapitre les notions mathématiques de base en traitement des images. Il y a en fait beaucoup de notions importantes en traitement des images que nous ne verrons pas, par manque de temps. Il faut savoir que le traitement des images est aujourd'hui un domaine dans lequel les grandes avancées sont basées sur des résultats mathématiques (ex : JPEG2000 est basé sur la décomposition en coefficients d'ondelettes).

### 1.1 La création de l'image numérique

La création d'une image numérique est faite par un appareil de mesure (scanner, appareil photo numérique, webcam, barrette CCD, ...). Malgré la diversité des appareils de mesure, elle s'écrit (à quelques approximations près) sous la forme d'une unique équation mathématique. Ce sont les éléments mathématiques utiles à l'écriture et à la compréhension de cette équation que nous allons introduire ici. Commençons par définir ce qu'est une image numérique.

#### 1.1.1 Images numériques

Une image numérique est définie sur une grille à deux dimensions. Les éléments de cette grille sont appelés des *pixels*. Ainsi, une image est définie sur un ensemble

$$\{1, \dots, M\} \times \{1, \dots, N\},$$

où  $M$  et  $N$  sont des entiers strictement positifs.

De plus, à chaque pixel, l'image attribue une couleur. Il existe plusieurs façons de représenter une couleur (système RVB, HSV, niveaux de gris, ...). Le point commun entre ces méthodes est de représenter une couleur par un ou plusieurs nombres (généralement trois). Chacun d'entre eux représentant la composante de notre couleur dans la direction d'une couleur primaire de référence.

Afin de simplifier les notations dans la suite du cours, nous ne travaillerons que sur des images noir et blanc. Ceci induit une simplification notable, puisqu'une couleur, que l'on appellera maintenant *niveau de gris*, n'est plus représentée que par un seul nombre. La convention habituelle veut que la valeur 0 corresponde au **noir** et que la valeur 255 corresponde au **blanc**. Les valeurs intermédiaires donnent les différentes teintes de gris. Il nous faut par ailleurs coder ces niveaux de gris. Pour ce faire, on doit utiliser un nombre fini de niveaux de gris. On a ainsi un ensemble fini  $Col \subset \mathbb{R}$  représentant nos couleurs. Par exemple, pour des niveaux de gris codés sur 8 bits, on a  $Col = \{0, 1, \dots, 255\}$ .

Ainsi, une image numérique est donnée par une suite

$$\begin{aligned} \{1, \dots, M\} \times \{1, \dots, N\} &\longrightarrow Col \\ (m, n) &\longrightarrow u_{m,n} \end{aligned}$$

Dans la suite, nous utiliserons indifféremment les termes *image*, *fonction* et *suite* pour désigner une image numérique. Nous précisons qu'une image ou une fonction n'est pas numérique en la qualifiant d'*analogique*. De plus, nous supposons que nos images sont carrées. On a ainsi  $M = N$ .

### 1.1.2 La convolution

Tous les appareils de mesure commencent par faire un “moyennage”, sur un voisinage d'un pixel, avant d'attribuer cette valeur au pixel. Ce moyennage ne dépend pas (on supposera en tout cas que cette dépendance, si elle existe, est négligeable) du pixel considéré. Mathématiquement, cette opération est connue sous le nom de convolution. C'est l'objet de ce chapitre.

#### De fonctions analogiques

Pour simplifier, nous ne considérerons que des fonctions, dont le module est *intégrable*<sup>1</sup>, définies sur  $\mathbb{R}^2$ . Ces fonctions représentent des images analogiques. On notera  $L^1(\mathbb{R}^2)$  l'ensemble des fonctions de module intégrable sur  $\mathbb{R}^2$ .

Pour deux fonctions  $h$  et  $v$  dans  $L^1(\mathbb{R}^2)$ , on note  $h * v$  le *produit de convolution de  $h$  par  $v$*  et on le définit par

$$\forall (x, y) \in \mathbb{R}^2, h * v(x, y) = \int_{\mathbb{R}} \int_{\mathbb{R}} h(x - x', y - y') v(x', y') dx' dy'.$$

Il s'agit donc d'une fonction (il n'est pas très difficile de voir qu'elle est aussi dans  $L^1(\mathbb{R}^2)$ ).

L'intuition qu'il faut avoir de  $h * v(x, y)$  est bien celle d'un “moyennage” de  $v$  au voisinage de  $(x, y)$ . Par exemple, si l'on prend  $h(x, y) = \mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]^2}(x, y)$ , on a, pour tout  $(x, y) \in \mathbb{R}^2$ ,

$$\begin{aligned} h * v(x, y) &= \int_{\mathbb{R}} \int_{\mathbb{R}} h(x - x', y - y') v(x', y') dx' dy' \\ &= \int_{y-\frac{1}{2}}^{y+\frac{1}{2}} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} v(x', y') dx' dy'. \end{aligned} \quad (1.1)$$

qui est bien la moyenne de  $v$  sur un carré de côté 1, centré en  $(x, y)$ . Modifier le support de  $h$  revient à changer le support sur lequel on fait la moyenne et modifier les valeurs de  $h$  revient à ajouter une pondération (certains points du voisinage de  $(x, y)$  comptant plus que d'autres). Il est important de noter que le moyennage ne dépend que de  $h$  et reste le même quel que soit  $(x, y) \in \mathbb{R}^2$ . On dit que l'opérateur de convolution est *invariant par translation*. On appelle  $h$  le *noyau de convolution*. En général, pour que la convolution soit vraiment un moyennage, on prend  $h$  tel que

$$\int_{\mathbb{R}^2} h(x, y) dx dy = 1$$

La convolution est la première dégradation subie par l'image analogique. Ainsi, à un appareil de mesure correspond un noyau de convolution  $h$  (celui-ci peut être du à l'optique, un compteur de photons, ...).

#### De suites finies

Nous profitons de l'introduction de la convolution pour les fonctions analogiques pour la définir pour des suites finies. Soit  $(h_{m,n})_{1 \leq m \leq N, 1 \leq n \leq N}$  et  $(v_{m,n})_{1 \leq m \leq N, 1 \leq n \leq N}$ , deux suites finies. On note  $h * v$ , le *produit de convolution de  $h$  par  $v$* , la suite

$$(h * v)_{m,n} = \sum_{m'=1}^N \sum_{n'=1}^N h_{m-m', n-n'} v_{m', n'}.$$

Ici, on suppose que  $h$  est périodisé en dehors de  $\{1, \dots, N\}^2$ . Ceci veut dire que l'on définit

$$\forall (m, n) \in \{1, \dots, N\}^2, \forall (k, l) \in \mathbb{Z}, h_{m+kN, n+lN} = h_{m,n}.$$

<sup>1</sup>Par intégrable, on veut dire que l'intégrale, sur son domaine de définition, du module de la fonction existe.

Il est à noter que, même si  $h$  et  $v$  sont des images numériques,  $h * v$  n'est pas forcément une image numérique car elle prend, à priori, des valeurs hors de  $\text{Col}$ . Nous verrons par la suite comment remédier à ce problème.

L'intuition est la même pour le produit de convolution entre des fonctions numériques et des fonctions analogiques. On a un effet de moyennage de  $v$  autour des points  $(m, n)$ . Vous pouvez à titre d'exercice retrouver une formule de type de (1.1) pour  $h_{m,n} = \frac{1}{9} \mathbf{1}_{\{-1,0,1\}^2}(m, n)$  et  $h_{m,n} = \frac{1}{25} \mathbf{1}_{\{-2,-1,0,1,2\}^2}(m, n)$ .

### 1.1.3 Le fenêtrage

Dans le chapitre précédent, pour la convolution entre fonctions analogiques, le domaine est  $\mathbb{R}^2$ . Or, en pratique, une image numérique ne représente qu'une partie finie de l'ensemble de la scène observable. Il y a donc un *fenêtrage* de cette image analogique avant la numérisation. Le choix de la fenêtre correspond à ce que les photographes appellent le cadrage.

Ainsi, après la convolution, on a une image analogique  $h * v$  dont on ne gardera que la partie intérieure à une fenêtre  $[0, N]^2$ . Ceci revient mathématiquement à la multiplier par  $\mathbf{1}_{[0, N]^2}$ .

Cette perte d'informations joue un rôle important près des bords de l'image. En effet, lors de la convolution (voir Figure 1.1), si  $(x, y)$  est tel qu'il existe  $(x', y')$  hors de  $[0, N]^2$  tel que  $h(x - x', y - y') \neq 0$ ,  $h * v(x, y)$  dépend de  $v$  en un endroit où on la connaît peu (en pratique pas).

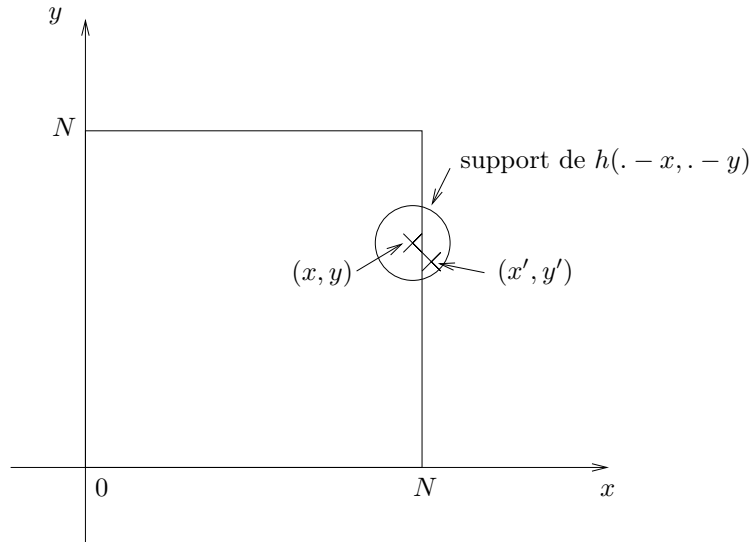


FIG. 1.1 – La valeur de  $h * v(x, y)$  dépend de la valeur de  $v(x', y')$ .

De même, l'information sur la valeur de  $v(x, y)$  est répartie sur les valeurs de  $h * v(x', y')$ , avec  $(x', y')$  tel que  $h(x' - x, y' - y) \neq 0$ . Si un tel  $(x', y')$  est en dehors de  $[0, N]^2$ , cette information est perdue.

Pour remédier à ces problèmes, nous prolongerons  $(h * v) \mathbf{1}_{[0, N]^2}$  par périodisation. Ceci revient à poser

$$h * v(x, y) = h * v(x + t_x N, y + t_y N)$$

avec  $t_x$  et  $t_y$  tels que  $(x + t_x N, y + t_y N) \in [0, N]^2$ .

Il y a bien sûr beaucoup d'autres possibilités pour traiter ces *problèmes de bord*. Nous choisissons la périodisation parce que grâce à elle toutes les formules utilisant la transformée de Fourier (voir Chapitre 1.2) seront exactes (ce ne seront pas des approximations).

### 1.1.4 L'échantillonnage

L'échantillonnage est bien souvent la partie du processus de création d'une image digitale durant laquelle la plus d'informations est perdue. Elle consiste à ne garder que les valeurs de  $(h * v) \mathbf{1}_{[0, N]^2}$  aux points entiers. Mathématiquement, on obtient une fonction définie sur  $\{1, \dots, N\}^2$  définie par

$$h * v(m, n) \mathbf{1}_{[0, N]^2}(m, n),$$

pour  $(m, n) \in \{1, \dots, N\}^2$ .

Pour sous-échantillonner, d'un rapport  $K$  (pour  $K$  un entier strictement positif), une image digitale  $u$  définie sur  $\{1, \dots, KN\}^2$ , pour obtenir une image digitale  $u'$  définie sur  $\{1, \dots, N\}^2$ , on effectue l'opération

$$u'_{m,n} = u_{Km,Kn}.$$

On considère en général que le sous-échantillonnage d'une image digitale approxime bien les effets de l'échantillonnage permettant de créer une image à partir de  $K = 3$ .

### 1.1.5 Le bruit

Un appareil de mesure créant une image digitale génère toujours un bruit. Les causes peuvent venir de plusieurs sources (caractère probabiliste du nombre de photons issus d'une région d'intensité donnée, imperfections électroniques de l'appareil, imperfections des capteurs, ...). Nous ne considérerons ici que les bruits additifs. Il s'agira d'une suite, définie sur  $\{1, \dots, N\}^2$ , dont la valeur est aléatoire. Bien que cela soit rarement le cas, il est souvent raisonnable de supposer que le bruit  $b$  est *blanc* (les valeurs  $b_{m,n}$  sont indépendantes les unes des autres). On supposera de plus qu'il est *Gaussien*. Ceci veut dire que chaque  $b_{m,n}$  est une réalisation de la loi de probabilité continue, de densité

$$p(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right), \quad (1.2)$$

pour  $\sigma > 0$ . On a introduit  $\sigma > 0$ , qui représente l'importance du bruit. On obtient alors une fonction définie sur  $\{1, \dots, N\}^2$ , valant

$$h * v(m, n) \mathbf{1}_{|[0,N]^2}(m, n) + b_{m,n}.$$

Pour un bruit  $b$ , comme pour toute variable aléatoire, on peut parler de *l'espérance* d'une fonction  $f(b)$ . On la définit mathématiquement par

$$\mathbb{E}(f(b)) = \int_{\mathbb{R}} f(t)p(t)dt$$

où  $p(t)$  est la densité de la loi de  $b$  (ici la loi Gaussienne définie par (1.2)). L'espérance représente la valeur moyenne de  $f(b)$  pour un nombre infini de réalisations indépendantes de  $b$ . C'est même en utilisant cette propriété que l'on calcule numériquement l'espérance, lorsque l'on sait produire des réalisations (indépendantes) de  $b$ .

Par exemple, il n'est pas dur de voir (ce sont de simples changements de variable) que

$$\mathbb{E}(b) = \frac{1}{\sqrt{2\pi}\sigma} \int_{\mathbb{R}} t \exp\left(-\frac{t^2}{2\sigma^2}\right) dt = 0.$$

Ceci représente la *valeur moyenne* de  $b$ .

On définit aussi la *variance* de  $b$ , par

$$\begin{aligned} \mathbb{E}\left((b - \mathbb{E}(b))^2\right) &= \frac{1}{\sqrt{2\pi}\sigma} \int_{\mathbb{R}} t^2 \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\ &= \frac{\sigma^2}{\sqrt{2\pi}} \int_{\mathbb{R}} t^2 \exp\left(-\frac{t^2}{2}\right) dt \\ &= \sigma^2 \end{aligned}$$

La variance (dans le cas d'un bruit Gaussien, simplement,  $\sigma^2$ ) nous donne l'écart quadratique moyen entre une réalisation de  $b$  et sa valeur moyenne.

### 1.1.6 La quantification

La quantification est l'opération qui consiste à traduire les valeurs de

$$h * v(m, n) \mathbf{1}_{|[0,N]^2}(m, n) + b_{m,n}.$$



sous la forme d'une couleur. Pour cela, il faut approximer cette valeur (qui est dans  $\mathbb{R}^3$ , pour des images couleurs et  $\mathbb{R}$  pour des images noir et blanc) de façon à ce qu'elle soit codable dans un ordinateur (on dispose d'un nombre fini de bits).

Si l'on considère le cas d'images noir et blanc avec des niveaux de gris appartenant à un ensemble  $\{0, 1, \dots, 255\}$ , on *quantifie* une valeur  $t \in \mathbb{R}$  en l'approximant par la valeur  $Ar(t)$ , la plus proche de  $t$  dans  $\{0, 1, \dots, 255\}$ .

On obtient ainsi enfin notre image digitale sous la forme

$$u_{m,n} = Ar(h * v(m, n) \mathbf{1}_{[0,N]^2}(m, n) + b_{m,n})$$

**Remarque :** Il est parfois nécessaire de modifier la dynamique de l'image avant la quantification (par exemple : si l'image est trop sombre, une quantification brutale engendrerait trop de perte d'informations ; si l'image est trop claire, beaucoup de points satureraient à la valeur 255). Nous négligerons dans la suite ce *changement de contraste*.

### 1.1.7 Autres dégradations

Il y a bien sûr beaucoup d'autres sources possibles de dégradation d'une image. Nous n'avons abordé ici que celles concernant l'appareil de mesure fonctionnant normalement. On peut mentionner par exemple :

- **Le changement de contraste :** Comme nous l'avons dit au chapitre précédent, on a parfois intérêt à modifier le contraste. Les caméras numériques et appareils photo numériques font presque toujours un changement de contraste pour s'adapter aux conditions d'éclairage.
- **La perte d'une partie de l'image :** Il peut arriver aussi qu'une partie de l'image soit perdue (par exemple : sur une photo abîmée, durant la transmission d'une image satellite, vieux films). Dans ce cas, on a un masque  $M$ , défini sur  $\{1, \dots, N\}^2$ , à valeur dans  $\{0, 1\}$  et la nouvelle image est donnée par

$$\tilde{u}_{m,n} = M_{m,n} u_{m,n}.$$

- **Des distorsions géométriques :** Certains appareils de mesure ne font pas un échantillonnage sur une grille parfaite. Cela peut être du à des imperfections du capteur, des vibrations d'un satellite, ... On a alors une fonction de déformation  $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  et l'image prend la forme

$$u_{m,n} = Ar(h * v(\varphi(m, n)) \mathbf{1}_{[0,N]^2}(\varphi(m, n)) + b_{m,n})$$

- **Les pertes dues à la compression :** Pour stocker ou transmettre une image, on la compresse souvent. Cette compression peut générer des défauts sur l'image reconstruite. Ces défauts dépendent évidemment de la méthode et du niveau de compression.

Il existe évidemment d'autres sources de dégradations possibles. Nous n'aborderons pas (ou peu) les méthodes visant à réduire les effets de ces dégradations.

### 1.1.8 Exemple

Nous montrons dans les images suivantes la dégradation due à une *chaîne image* (l'ensemble du processus permettant de créer une image) définie par un noyau de convolution  $h(x, y) = \mathbf{1}_{[-1,1]^2}$  et un bruit de variance  $\sigma = 4$ . (Ce noyau de convolution est réaliste, on rencontre des noyaux ayant le même genre d'effets en imagerie satellite. Le bruit dans un tel cas serait plus faible. Nous l'avons volontairement augmenté afin qu'il soit bien visible.)



FIG. 1.2 – Image analogique de départ.



FIG. 1.3 – Image analogique après la convolution.



FIG. 1.4 – Image analogique après la convolution et le fenêtrage.



FIG. 1.5 – Image analogique après la convolution, le fenêtrage et l'échantillonnage.



FIG. 1.6 – Image analogique après la convolution, le fenêtrage, l'échantillonnage et l'ajout d'un bruit.



FIG. 1.7 – Image analogique après la convolution, le fenêtrage, l'échantillonnage, l'ajout d'un bruit et la quantification. C'est une image digitale.

## 1.2 La transformée de Fourier

La transformée de Fourier est un outil mathématique important en traitement des images pour deux raisons :

- La plupart des dégradations rencontrées lors de la création de l'image s'expriment simplement en terme de transformée de Fourier. Cette dernière permet donc de comprendre le comportement d'une chaîne image.
- C'est un exemple (le seul que nous verrons) de traitement d'une image à partir de la représentation de l'image dans une base (autre que la base canonique). C'est un des grands domaines de recherche actuels.

### 1.2.1 D'une image analogique

Nous nous contenterons ici de définir la transformée de Fourier d'une fonction définie sur une fenêtre. Elle est en fait aussi définie pour une fonction définie sur  $\mathbb{R}^2$ . Il y a même un lien entre la transformée de Fourier d'une fonction définie sur  $\mathbb{R}^2$  et celle de cette même fonction après le fenêtrage. Les détails de ce passage n'apportent cependant pas grand chose à la compréhension du fenêtrage. Nous le laisserons donc de côté.

**Définition 1** Soit  $v \in L^1([0, N]^2)$ , sa transformée de Fourier est définie, pour  $(k, l) \in \mathbb{Z}^2$ , par

$$\hat{v}_{\frac{k}{N}, \frac{l}{N}} = \int_0^N \int_0^N v(x, y) e^{-2i\pi \frac{kx+ly}{N}} dx dy,$$

où  $i$  représente le nombre complexe habituel.

Vous pourrez rencontrer d'autres définitions équivalentes (notamment sans le  $2\pi$ ) de la transformée de Fourier. L'intérêt de celle-ci est de simplifier plusieurs formules que nous verrons dans la suite. La normalisation du domaine de Fourier (on parle de  $\hat{v}_{\frac{k}{N}, \frac{l}{N}}$  et pas de  $\hat{v}_{k,l}$ ) vient du lien entre la transformée de Fourier d'une fonction définie sur  $\mathbb{R}^2$  et d'un fenêtrage de cette dernière. Dans la suite, on appellera *fréquences* les points du domaine de Fourier.

**Remarque 1 :** Dans toutes les manipulations de la transformée de Fourier de fonctions de  $L^1([0, N]^2)$ , on supposera que  $v$  est périodisée en dehors de  $[0, N]^2$ .

**Remarque 2 :** La transformée de Fourier est, en fait, définie pour des fonctions à valeur dans  $\mathbb{C}$ . Les coefficients de Fourier sont d'ailleurs dans  $\mathbb{C}$ . Par contre, comme la fonction est à valeur dans  $\mathbb{R}$ , on a (on note  $z^*$  le nombre complexe conjugué de  $z$ )

$$\begin{aligned} \hat{v}_{\frac{k}{N}, \frac{l}{N}}^* &= \int_0^N \int_0^N v(x, y) \left( e^{-2i\pi \frac{kx+ly}{N}} \right)^* dx dy \\ &= \int_0^N \int_0^N v(x, y) e^{-2i\pi \frac{-kx-ly}{N}} dx dy \\ &= \hat{v}_{-\frac{k}{N}, -\frac{l}{N}}. \end{aligned}$$

**Remarque 3 :** Une des intuitions importantes qu'il faut avoir (nous ne l'aborderons pas de façon formelle) est que *plus la fonction  $v$  est régulière, plus ses coefficients de Fourier décroissent rapidement*.

**Remarque 4 :** Un autre aspect, très important en traitement des images, est que la transformée de Fourier est une transformation globale (l'intégrale porte sur tout le domaine  $[0, N]^2$ ). Ainsi changer  $v$ , même sur une petite région de  $[0, N]^2$ , a un impact sur tous les coefficients de Fourier.

**Exemple**

L'exemple que nous allons traiter est le calcul de la transformée de Fourier de la fonction  $v = \frac{1}{a^2} \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]^2}$ , pour  $a > 0$ . Il est important car on le rencontre souvent. On a, pour tout  $(k, l) \in \mathbb{Z}^2$ ,

$$\begin{aligned} \hat{v}_{\frac{k}{N}, \frac{l}{N}} &= \int_0^N \int_0^N \frac{1}{a^2} \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]^2}(x, y) e^{-2i\pi \frac{kx+ly}{N}} dx dy \\ &= \int_0^N \frac{1}{a} \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]}(x) e^{-2i\pi \frac{kx}{N}} dx \int_0^N \frac{1}{a} \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]}(y) e^{-2i\pi \frac{ly}{N}} dy. \end{aligned}$$

Car  $e^{-2i\pi \frac{kx+ly}{N}} = e^{-2i\pi \frac{kx}{N}} e^{-2i\pi \frac{ly}{N}}$  et  $\mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]^2}(x, y) = \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]}(x) \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]}(y)$ . On a alors, comme  $e^{-2i\pi \frac{kx}{N}}$  est périodique (de période  $N$ ),

$$\begin{aligned} \int_0^N \frac{1}{a} \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]}(x) e^{-2i\pi \frac{kx}{N}} dx &= \frac{1}{a} \int_{-\frac{a}{2}}^{\frac{a}{2}} e^{-2i\pi \frac{kx}{N}} dx \\ &= \frac{N}{-2ik\pi a} \left[ e^{-2i\pi \frac{kx}{N}} \right]_{-\frac{a}{2}}^{\frac{a}{2}} \\ &= \frac{N}{-2ik\pi a} \left( e^{-i\pi \frac{ka}{N}} - e^{i\pi \frac{ka}{N}} \right) \\ &= \frac{N}{k\pi a} \sin\left(\frac{k\pi a}{N}\right) \\ &= \text{sinc}\left(\frac{k\pi a}{N}\right), \end{aligned}$$

avec  $\text{sinc}(t) = \frac{\sin(t)}{t}$ , si  $t \neq 0$ , et  $\text{sinc}(0) = 1$ . On a donc

$$\hat{v}_{\frac{k}{N}, \frac{l}{N}} = \text{sinc}\left(\frac{k\pi a}{N}\right) \text{sinc}\left(\frac{l\pi a}{N}\right). \quad (1.3)$$

On appelle *sinus cardinal* la fonction *sinc* apparaissant dans (1.3). Elle donne la transformée de Fourier de la fonction de fenêtrage telle que nous l'avons vue au chapitre 1.1.3. C'est aussi la transformée de Fourier du noyau de convolution que nous avons vu en TP.

On voit que  $\frac{1}{a^2} \mathbf{1}_{[-\frac{a}{2}, \frac{a}{2}]^2}$  est discontinue et que sa transformée de Fourier décroît comme  $\frac{1}{kl}$ .  $\square$

Une propriété, qui rend la transformée de Fourier utile, est qu'elle est inversible. C'est à dire qu'à partir de coefficients de Fourier, on peut reconstruire une fonction correspondante. Plus précisément :

**Proposition 1** Soit  $v \in L^1([0, N]^2)$ . On note  $\hat{v}_{\frac{k}{N}, \frac{l}{N}}$ , pour  $k$  et  $l$  dans  $\mathbb{Z}$  ses coefficients de Fourier, on a alors pour tout  $(x, y) \in [0, N]^2$

$$v(x, y) = \frac{1}{N^2} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \hat{v}_{\frac{k}{N}, \frac{l}{N}} e^{2i\pi \frac{kx+ly}{N}}.$$

La preuve de cette propriété nous entraînerait dans des détails mathématiques dépassant la portée de ce cours, nous la verrons dans le cas de la transformée de Fourier de suites finies.

Les intérêts de cette proposition sont multiples. Tout d'abord, en pratique, elle permet de reconstruire une fonction à partir de ses coefficients de Fourier. Ce qui veut dire que l'on peut calculer la transformée de Fourier d'une fonction, manipuler ses coefficients (de façon intelligente, si possible) et reconstruire un résultat.

Par ailleurs, elle met en évidence le sens de la transformée de Fourier, qui est de calculer les coordonnées d'une image dans une base constituée des fonctions  $\left( e^{2i\pi \frac{kx+ly}{N}} \right)_{k, l \in \mathbb{Z}}$ .

**Les coefficients de Fourier ne sont ainsi qu'une autre façon de décrire une fonction analogique.**

### 1.2.2 D'une suite finie

Dans cette partie, nous allons définir les notions analogues à celles que nous avons déjà vues pour des fonctions analogiques, mais pour des suites finies. Tout cela s'applique évidemment aux images numériques (la quantification n'induit pas de difficulté supplémentaire).

**Définition 2** Soit  $(w_{m,n})_{1 \leq m \leq N, 1 \leq n \leq N}$  une suite finie de nombres réels. Sa transformée de Fourier discrète est définie pour  $(k, l) \in \{1, \dots, N\}^2$  (ou, de façon équivalente, sur  $(k, l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2}\}^2$ ) par

$$\hat{w}_{k,l} = \sum_{m,n=1}^N w_{m,n} e^{-2i\pi \frac{km+ln}{N}}$$

L'intuition est la même pour la transformée de Fourier discrète que pour la transformée de Fourier d'une fonction analogique. Un élément nouveau, cependant, est que la transformée de Fourier discrète peut en fait être définie sur  $\mathbb{Z}^2$  auquel cas elle est périodique. (Il n'est pas difficile de voir que  $\hat{w}_{k+t_1N, l+t_2N} = \hat{w}_{k,l}$ . C'est dû au fait que, pour tout  $(m, n) \in \mathbb{Z}^2$ ,  $e^{-2i\pi \frac{km+ln}{N}} = e^{-2i\pi \frac{(k+t_1N)m+(l+t_2N)n}{N}}$ .)

Comme dans le cas de la transformée de Fourier, le fait que la suite  $w$  soit à valeur dans  $\mathbb{R}$  implique que

$$\hat{w}_{k,l}^* = \hat{w}_{-k,-l}. \quad (1.4)$$

L'inversion de la transformée de Fourier discrète est donnée par

**Proposition 2 (inversion de la transformée de Fourier)**

Soit  $(w_{m,n})_{1 \leq m \leq N, 1 \leq n \leq N}$  une suite finie de nombres réels. Si on note  $(\hat{w}_{k,l})_{-\frac{N}{2}-1 \leq k, l \leq \frac{N}{2}}$ , ses coefficients de Fourier, on a alors, pour tout  $(m, n) \in \{1, \dots, N\}^2$

$$w_{m,n} = \frac{1}{N^2} \sum_{k,l=1}^N \hat{w}_{k,l} e^{2i\pi \frac{km+ln}{N}}.$$

*Preuve.* On a en effet

$$\begin{aligned} \frac{1}{N^2} \sum_{k,l=1}^N \hat{w}_{k,l} e^{2i\pi \frac{km+ln}{N}} &= \frac{1}{N^2} \sum_{k,l=1}^N \left( \sum_{m',n'=1}^N w_{m',n'} e^{-2i\pi \frac{km'+ln'}{N}} \right) e^{2i\pi \frac{km+ln}{N}} \\ &= \frac{1}{N^2} \sum_{m',n'=1}^N w_{m',n'} \sum_{k=1}^N e^{-2i\pi \frac{k(m'-m)}{N}} \sum_{l=1}^N e^{-2i\pi \frac{l(n'-n)}{N}} \end{aligned}$$

Étudions  $\sum_{k=1}^N e^{-2i\pi \frac{k(m'-m)}{N}}$  (on aura le même résultat pour  $\sum_{l=1}^N e^{-2i\pi \frac{l(n'-n)}{N}}$ ). Si  $m' = m$ , on a

$$\sum_{k=1}^N e^{-2i\pi \frac{k(m'-m)}{N}} = \sum_{k=1}^N 1 = N.$$

De plus, si  $m' \neq m$ , on a

$$\sum_{k=1}^N e^{-2i\pi \frac{k(m'-m)}{N}} = \sum_{k=1}^N a^k,$$

avec  $a = e^{-2i\pi \frac{(m'-m)}{N}}$ . Donc

$$\begin{aligned} \sum_{k=1}^N e^{-2i\pi \frac{k(m'-m)}{N}} &= \frac{a^{N+1} - a}{a - 1} \\ &= 0, \end{aligned}$$

car  $a^{N+1} = e^{-2i\pi \frac{(N+1)(m'-m)}{N}} = e^{-2i\pi(m'-m)} e^{-2i\pi \frac{m'-m}{N}} = a$ .

On a ainsi,

$$\frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N \hat{w}_{k,l} e^{2i\pi \frac{km+ln}{N}} = w_{m,n}.$$

□

Un intérêt supplémentaire de la transformée de Fourier discrète est qu'il existe un algorithme rapide pour la calculer (ainsi que son inverse). Il s'agit de la *Fast Fourier Transform* ou *FFT*. C'est un algorithme dont la complexité est de l'ordre de  $N^2 \log(N)$ , pour une image de taille  $N \times N$ .

**Exemple :** Un calcul similaire à celui que nous avons fait au chapitre précédent nous assure que la transformée de Fourier discrète d'une fonction  $w = \frac{1}{(2M+1)^2} \mathbf{1}_{\{-M, \dots, M\}^2}$  vaut

$$\hat{w}_{k,l} = \text{sinc}_d^M \left( \frac{k\pi}{N} \right) \text{sinc}_d^M \left( \frac{l\pi}{N} \right)$$

avec  $\text{sinc}_d^M(t) = \frac{\sin(t(2M+1))}{(2M+1)\sin(t)}$ , si  $t \neq 0$ , et  $\text{sinc}_d^M(0) = 1$ . C'est la version discrète du sinus cardinal.

### 1.2.3 Convolution et transformée de Fourier

L'un des intérêts majeurs de la transformée de Fourier est le lien qui la lie à la convolution. En effet, on a

**Proposition 3** Soit  $h$  et  $w$  deux suites finies, définies sur  $\{1, \dots, N\}^2$ . On a alors, pour tout  $(k, l) \in \{1, \dots, N\}^2$ ,

$$(\widehat{h * w})_{k,l} = \hat{h}_{k,l} \hat{w}_{k,l}.$$

*Preuve.* En effet,

$$\begin{aligned} (\widehat{h * w})_{k,l} &= \sum_{m,n=1}^N (h * w)_{m,n} e^{-2i\pi \frac{km+ln}{N}} \\ &= \sum_{m,n=1}^N \left( \sum_{m',n'=1}^N h_{m-m',n-n'} w_{m',n'} \right) e^{-2i\pi \frac{k(m-m')+l(n-n')}{N}} e^{-2i\pi \frac{km'+ln'}{N}} \\ &= \sum_{m',n'=1}^N w_{m',n'} e^{-2i\pi \frac{km'+ln'}{N}} \sum_{m,n=1}^N h_{m-m',n-n'} e^{-2i\pi \frac{k(m-m')+l(n-n')}{N}}. \end{aligned}$$

Pour la dernière somme, on a, en changeant de variable,

$$\sum_{m,n=1}^N h_{m-m',n-n'} e^{-2i\pi \frac{k(m-m')+l(n-n')}{N}} = \sum_{m=1-m'}^{N-m'} \sum_{n=1-n'}^{N-n'} h_{m,n} e^{-2i\pi \frac{km+ln}{N}}.$$

Or, du fait de la périodicité de  $h$  et de l'exponentielle, on a, pour  $m \in \{1-m', \dots, 0\}$  (idem pour  $n \in \{1-n', \dots, 0\}$ ),

$$h_{m,n} e^{-2i\pi \frac{km+ln}{N}} = h_{m+N,n} e^{-2i\pi \frac{k(m+N)+ln}{N}}$$

et  $m+N \in \{N-m'+1, \dots, N\}$ . Ainsi, on peut réarranger la somme et l'on a

$$\begin{aligned} \sum_{m,n=1}^N h_{m-m',n-n'} e^{-2i\pi \frac{k(m-m')+l(n-n')}{N}} &= \sum_{m,n=1}^N h_{m,n} e^{-2i\pi \frac{km+ln}{N}} \\ &= \hat{h}_{k,l}. \end{aligned}$$



On trouve donc bien

$$\begin{aligned}\widehat{h * w}_{k,l} &= \sum_{m',n'=1}^N w_{m',n'} e^{-2i\pi \frac{km'+ln'}{N}} \hat{h}_{k,l} \\ &= \hat{h}_{k,l} \hat{w}_{k,l}.\end{aligned}$$

□

Une propriété analogue peut, bien entendu, être énoncée pour la convolution entre fonctions de  $L^1([0, N]^2)$  et pour la transformée de Fourier dans  $L^1([0, N]^2)$ . Elle est aussi vraie pour la transformée de Fourier inverse.

Cette propriété est d'un grand intérêt pratique. On peut en effet l'utiliser pour faire une convolution quand le noyau de convolution  $h$  a un support étendu en espace. En effet, la complexité d'une convolution effectuée directement est  $N^2T$ , où  $T$  représente la taille du support du noyau. Il vaut donc mieux faire la convolution dans le domaine de Fourier, dès que  $T > \log(N)$ .

A ce stade, il est facile de montrer une propriété de la transformée de Fourier que nous utiliserons plus tard.

**Proposition 4** *Soit  $w$  une suite finie de réels, définie sur  $\{1, \dots, N\}^2$ . On a alors*

$$\sum_{k,l=1}^N |\hat{w}_{k,l}|^2 = N^2 \sum_{m,n=1}^N w_{m,n}^2.$$

*Preuve.* On a en effet,

$$\begin{aligned}\sum_{k,l=1}^N |\hat{w}_{k,l}|^2 &= \sum_{k,l=1}^N \hat{w}_{k,l}^* \hat{w}_{k,l} \\ &= \sum_{k,l=1}^N (\widehat{\tilde{w} * w})_{k,l}\end{aligned}$$

avec  $\tilde{w}_{m,n} = w_{-m,-n}$ . (Il est facile de voir que l'on a bien  $\hat{\tilde{w}}_{k,l} = \hat{w}_{k,l}^*$ .) On peut maintenant appliquer le résultat sur l'inversion de la transformation de Fourier et on obtient

$$\begin{aligned}\sum_{k,l=1}^N |\hat{w}_{k,l}|^2 &= \sum_{k,l=1}^N (\widehat{\tilde{w} * w})_{k,l} \\ &= \sum_{k,l=1}^N (\widehat{\tilde{w} * w})_{k,l} e^{2i\pi \frac{k \cdot 0 + l \cdot 0}{N}} \\ &= N^2 (\tilde{w} * w)_{0,0} \\ &= N^2 \sum_{m,n=1}^N \tilde{w}_{0-m,0-n} w_{m,n} \\ &= N^2 \sum_{m,n=1}^N w_{m,n}^2.\end{aligned}$$

□

Cette proposition implique notamment que si, pour une raison ou pour une autre (quantification, bruit, ...), nous commettons une erreur sur les coefficients de Fourier, on a une erreur "globalement" du même ordre sur la suite correspondante (il suffit d'appliquer cette proposition à l'erreur en question). Cette proposition nous garantit la stabilité des manipulations que nous ferons sur les coefficients de Fourier.

### 1.2.4 Échantillonnage et transformée de Fourier

Un autre intérêt de la transformée de Fourier vient du lien qui existe entre la transformée de Fourier d'une suite finie définie  $\{1, \dots, KN\}^2$  (un résultat analogue existe pour les fonctions de  $L^1([0, N]^2)$ ) et la transformée de Fourier du sous-échantillonnage d'un rapport  $K$  de cette suite. On a en effet,

**Proposition 5 (Formule de Poisson)**

Soit  $w$  une suite finie de nombres réels, définie sur  $\{1, \dots, KN\}^2$ . On note  $u$  le sous-échantillonnage de  $w$  défini, pour  $(m, n) \in \{1, \dots, N\}^2$ , par

$$u_{m,n} = w_{Km,Kn}.$$

On a alors, pour tout  $(k, l) \in \{1, \dots, N\}$

$$\hat{u}_{k,l} = \frac{1}{K^2} \sum_{t_1=0}^{K-1} \sum_{t_2=0}^{K-1} \hat{w}_{k+t_1N, l+t_2N}.$$

*Preuve.*

En effet, on a

$$\begin{aligned} \hat{u}_{k,l} &= \sum_{m,n=1}^N u_{m,n} e^{-2i\pi \frac{km+ln}{N}} \\ &= \sum_{m,n=1}^N w_{Km,Kn} e^{-2i\pi \frac{km+ln}{N}} \\ &= \sum_{m,n=1}^N \left( \frac{1}{K^2 N^2} \sum_{k',l'=1}^{KN} \hat{w}_{k',l'} e^{2i\pi \frac{k'Km+l'Kn}{KN}} \right) e^{-2i\pi \frac{km+ln}{N}} \\ &= \frac{1}{K^2 N^2} \sum_{k',l'=1}^{KN} \hat{w}_{k',l'} \sum_{m,n=1}^N e^{-2i\pi \frac{(k-k')m+(l-l')n}{N}} \\ &= \frac{1}{K^2 N^2} \sum_{k',l'=1}^{KN} \hat{w}_{k',l'} \sum_{m=1}^N e^{-2i\pi \frac{(k-k')m}{N}} \sum_{n=1}^N e^{-2i\pi \frac{(l-l')n}{N}}. \end{aligned} \quad (1.5)$$

Mais, on voit alors (le même calcul s'applique aussi à  $\sum_{n=1}^N e^{-2i\pi \frac{(l-l')n}{N}}$ ) que

– si  $k'=k+tN$ , avec  $t \in \mathbb{Z}$ ,

$$\sum_{m=1}^N e^{-2i\pi \frac{(k-k')m}{N}} = \sum_{m=1}^N 1 = N.$$

– sinon

$$\begin{aligned} \sum_{m=1}^N e^{-2i\pi \frac{(k-k')m}{N}} &= \frac{e^{-2i\pi \frac{(k-k')(N+1)}{N}} - e^{-2i\pi \frac{k-k'}{N}}}{e^{-2i\pi \frac{k-k'}{N}} - 1} \\ &= 0. \end{aligned}$$

On peut donc se contenter de faire la première somme de (1.5) pour des indices de la forme  $k' = k + t_1N$  avec  $t_1 \in \{0, \dots, K-1\}$  (même chose pour  $l'$ ). On obtient alors

$$\hat{u}_{k,l} = \frac{1}{K^2} \sum_{t_1=0}^{K-1} \sum_{t_2=0}^{K-1} \hat{w}_{k+t_1N, l+t_2N},$$

qui est le résultat annoncé. □

Ainsi, lors de l'échantillonnage, le coefficient de Fourier que l'on mesure est en fait la somme de plusieurs coefficients de Fourier de la fonction de départ. Ceci est illustré par la Figure 1.8. On appelle les fréquences  $(k + t_1N, l + t_2N)$ , qui modifient la vraie valeur de  $\hat{w}_{k,l}$ , les *alias*. Il faut faire attention ici au sens des fréquences. En effet, les basses fréquences de  $w$  (celles correspondant aux fréquences présentes dans  $u$ ) sont celles vérifiant

$$k \in \{1, \dots, \frac{N}{2} - 1\} \cup \{KN - \frac{N}{2}, \dots, KN\}$$

et

$$l \in \{1, \dots, \frac{N}{2} - 1\} \cup \{KN - \frac{N}{2}, \dots, KN\}$$

Comme les images sont généralement basses fréquences, les alias seront les fréquences qui n'appartiennent pas à cet ensemble.

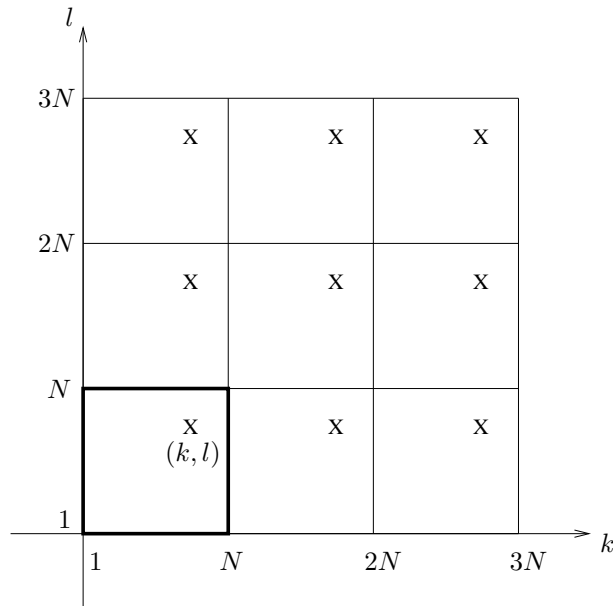


FIG. 1.8 – La valeur mesurée en  $(k, l)$  est la somme des valeurs de la transformée de Fourier, avant le sous-échantillonnage, aux autres points désignés par les croix.

Cette formule permet d'expliquer certains phénomènes que nous observons lors de l'échantillonnage d'une image. On appelle ce genre de phénomène de *aliasing*. On peut le voir notamment lorsque l'image contient une texture qui est bien localisée en fréquence (elle est représentée par un petit nombre de coefficients, tous localisés dans un voisinage d'une fréquence). Cette texture est alors "recopiée" dans une autre zone du plan fréquentiel. Ceci a pour effet de complètement modifier la texture en question. Ceci est illustré par la figure 1.9.

Il est important de noter que l'aliasing est un phénomène qu'un bon appareil de mesure doit éviter et ce pour au moins deux raisons. Tout d'abord, il n'est à priori pas possible de distinguer l'aliasing de la véritable information contenue initialement à la fréquence  $(k, l)$ . Ceci peut donc conduire à de mauvaises interprétations de l'image. Il vaut donc mieux effacer cette "fausse" information. Dans le cas de film, l'aliasing ne sera pas identique d'une image du film à une autre. Ainsi, on verra une texture bouger au fur et à mesure que le film se déroule.

On peut éviter (ou tout du moins réduire énormément) l'aliasing à l'aide de la convolution qui précède l'échantillonnage. En effet, si on exprime en fréquence la dégradation subie par l'image à la fin de l'échantillonnage (on néglige ici le fenêtrage en supposant l'image de départ périodique), on a, en posant

$$u_{m,n} = h * v(m, n),$$

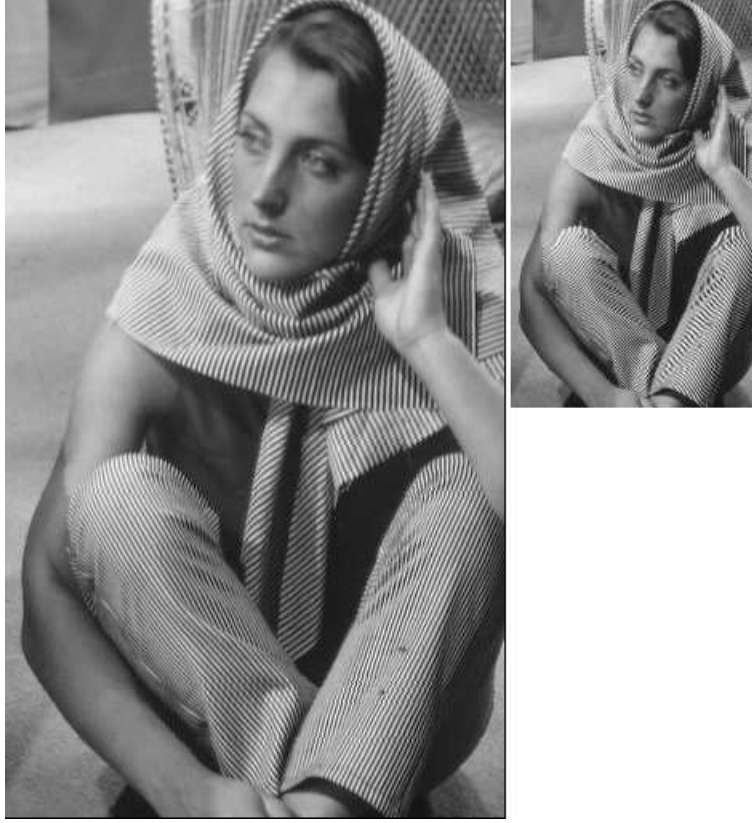


FIG. 1.9 – Gauche: Image de départ. Droite: Image sous-échantillonnée. La texture (qui est bien localisée en fréquence) a changé d'orientation durant le sous-échantillonnage.

on obtient, pour  $(k, l) \in \{1, \dots, N\}^2$ ,

$$\hat{u}_{k,l} = \sum_{t_1=0}^{K-1} \sum_{t_2=0}^{K-1} \hat{h}_{k+t_1N, l+t_2N} \hat{v}_{k+t_1N, l+t_2N}$$

(Rappel : faire une convolution revient à faire une multiplication des transformées de Fourier). Il suffit de prendre  $\hat{h}_{k,l} = 0$ , dès que  $(k, l)$  est un alias, pour éviter l'aliasing. Cela n'est en fait pas si simple. On se contente, en fait, souvent de noyaux  $h$  dont la transformée de Fourier est faible aux alias.

### 1.2.5 Bruit et transformée de Fourier

L'intérêt de l'hypothèse que nous avons faite sur le bruit (il est Gaussien) est que sa transformée de Fourier reste un bruit Gaussien. (En effet, la loi d'une somme de bruits est la convolution des lois des bruits. De plus, une convolution de fonctions Gaussiennes reste une Gaussienne.)

De plus, pour tout  $(k, l) \in \{1, \dots, N\}^2$ , son espérance vaut

$$\begin{aligned} \mathbb{E}(\hat{b}_{k,l}) &= \mathbb{E} \left( \sum_{m,n=1}^N b_{m,n} e^{-2i\pi \frac{km+ln}{N}} \right) \\ &= \sum_{m,n=1}^N \mathbb{E}(b_{m,n}) e^{-2i\pi \frac{km+ln}{N}} \\ &= 0. \end{aligned}$$

On peut aussi calculer sa variance qui vaut, quel que soit  $(k, l) \in \{1, \dots, N\}^2$ ,

$$\begin{aligned} \mathbb{E}(|\hat{b}_{k,l}|^2) &= \mathbb{E}\left(\left|\sum_{m,n=1}^N b_{m,n} e^{-2i\pi \frac{km+ln}{N}}\right|^2\right) \\ &= \mathbb{E}\left(\left(\sum_{m,n=1}^N b_{m,n} e^{-2i\pi \frac{km+ln}{N}}\right)\left(\sum_{m',n'=1}^N b_{m',n'} e^{2i\pi \frac{km'+ln'}{N}}\right)\right) \\ &= \sum_{m,n=1}^N \sum_{m',n'=1}^N \mathbb{E}(b_{m,n} b_{m',n'}) e^{-2i\pi \frac{k(m-m')+l(n-n')}{N}}. \end{aligned}$$

Or, si  $m \neq m'$  ou  $n \neq n'$ , comme les différentes valeurs de  $b$  sont indépendantes, on a

$$\mathbb{E}(b_{m,n} b_{m',n'}) = \mathbb{E}(b_{m,n}) \mathbb{E}(b_{m',n'}) = 0.$$

Si, au contraire,  $m = m'$  et  $n = n'$ ,

$$\mathbb{E}(b_{m,n} b_{m',n'}) = \mathbb{E}((b_{m,n})^2) = \sigma^2.$$

On a donc

$$\begin{aligned} \mathbb{E}(|\hat{b}_{k,l}|^2) &= \sum_{m,n=1}^N \sigma^2 \\ &= N^2 \sigma^2. \end{aligned}$$

Par contre, les  $\hat{b}_{k,l}$  ne sont pas des variables aléatoires indépendantes. On a en effet toujours

$$\hat{b}_{k,l}^* = \hat{b}_{-k,-l}$$

### 1.2.6 Fourier : une base parmi d'autres

La notion de base vient de l'algèbre linéaire. Nous resterons ici en dimension finie. Plus précisément, une base de  $\mathbb{R}^{N^2}$  (ou  $\mathbb{C}^{N^2}$ ) est un ensemble  $(e^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$ , tel que pour tout  $(k, l) \in \{0, \dots, N-1\}^2$ ,  $e^{k,l} \in \mathbb{R}^{N^2}$  (ou  $\mathbb{C}^{N^2}$ ) tel que

$$\forall v \in \mathbb{R}^{N^2}, \exists! (\lambda_{k,l})_{(k,l) \in \{0, \dots, N-1\}^2} \in \mathbb{R}^{N^2}, \text{ tel que } v = \sum_{k,l=0}^{N-1} \lambda_{k,l} e^{k,l}.$$

(Ci-dessus, il faut remplacer tous les  $\mathbb{R}^{N^2}$  par des  $\mathbb{C}^{N^2}$ , dans le cas de l'espace  $\mathbb{C}^{N^2}$ .)

**Exemple :** Il est facile de voir que  $(\delta^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$  est une base de  $\mathbb{R}^{N^2}$ , pour

$$\delta_{m,n}^{k,l} = \begin{cases} 1 & , \text{ si } (m, n) = (k, l), \\ 0 & , \text{ si } (m, n) \neq (k, l). \end{cases}$$

Dans ce cas, l'élément  $\delta^{k,l}$  est une image ayant tous ses pixels à 0, et seul le pixel  $(k, l)$  est à 1. Les coordonnées d'une image  $v$  dans cette base sont simplement les valeurs de ses niveaux de gris. On l'appelle la base canonique.

Le cas de Fourier correspond à une base de  $\mathbb{C}^{N^2}$ , constituée des éléments  $(e^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$ , avec pour tout  $(k, l)$ ,

$$e_{m,n}^{k,l} = e^{2i\pi \frac{km+ln}{N}}, \text{ pour } (m, n) \in \{0, \dots, N-1\}^2. \quad (1.6)$$

Cela n'est en effet pas très difficile à voir (voir Proposition 2 et remarquer qu'il y a  $N^2$  éléments  $e^{k,l}$ ). Comme nous l'avons vu, le fait que les images soient en fait dans  $\mathbb{R}^{N^2}$  ne pose pas de difficulté. Cela se traduit juste par une équation qui est satisfaite par nos images (voir (1.4)).

On peut interpréter les résultats que nous avons déjà vus, dans ce contexte. Ainsi, la Proposition 2 nous dit que les coefficients de Fourier sont les coordonnées d'une image dans la base de Fourier. La FFT est donc simplement un changement de base. (On passe de la base  $(\delta^{k,l})_{(k,l) \in \{0,\dots,N-1\}^2}$  vue en exemple à la base de Fourier, ou l'inverse.)

De même, la Proposition 3 nous dit qu'une convolution est un opérateur diagonal dans la base de Fourier. Ses valeurs propres sont les coefficients de Fourier du noyau de convolution.

La Proposition 4 nous garantit que la base de Fourier est une base orthogonale. La proposition 5 pourrait s'exprimer en terme de matrice...

En résumé, toutes les propriétés que nous avons vues (et qui nous permettront d'obtenir les algorithmes de traitement d'images du Chapitre 2) se traduisent en terme de propriétés sur les bases.

En fait, quelque soit l'application sur laquelle on travaille, en traitement d'images, on peut toujours choisir une base particulière. Dans certains cas, on préférera la base canonique, Fourier, ou on peut créer une base qui aurait de bonnes propriétés vis-à-vis de l'application visée.

Nous allons voir dans la suite d'autres bases qui sont très utilisées pour la compression et la restauration d'images.

## 1.3 Bases de cosinus et cosinus locaux

### 1.3.1 Les cosinus

Comme nous l'avons vu au chapitre précédent, la base de Fourier est périodique. On a, pour  $e^{k,l}$  défini par (1.6) et quelque soit  $(m,n) \in \{0,\dots,N-1\}^2$

$$\begin{aligned} e_{m,n}^{k,l} &= e^{2i\pi \frac{km+ln}{N}} \\ &= e^{2i\pi \frac{k(m+N)+l(n+N)}{N}} \\ &= e_{m+N,n+N}^{k,l}. \end{aligned}$$

Ceci crée souvent un brusque changement près des bords de l'image. Ce n'est pas souhaitable pour (au moins) deux raisons :

- Dans les traitements utilisant la transformée de Fourier (par exemple la convolution), on aura souvent des problèmes près des bords puisque l'on utilise de l'information provenant de deux côtés opposés de l'image.
- On a vu que la régularité d'une image est liée à la décroissance des coefficients de Fourier (pour les hautes fréquences). La discontinuité que l'on crée près des bords de l'image se traduit donc par des "grands" coefficients de Fourier. (Ce qui n'est pas souhaitable pour des application comme la compression où l'on cherche à avoir le moins de grands coefficients possible.)

La base de cosinus a donc été introduite pour éviter ce défaut. Elle ressemble beaucoup à la base de Fourier. En fait, c'est, à peu de choses près, la transformée de Fourier de l'image périodisée. Plus précisément, on prolonge une suite finie  $(w_{m,n})_{(m,n) \in \{0,\dots,N-1\}^2}$ , en la suite finie  $(\tilde{w}_{m,n})_{(m,n) \in \{0,\dots,2N-1\}^2}$ , en posant

$$\tilde{w}_{m,n} = \tilde{w}_{m,2N-1-n} = \tilde{w}_{2N-1-m,n} = \tilde{w}_{2N-1-m,2N-1-n} = w_{m,n} \quad (1.7)$$

pour  $(m,n) \in \{0,\dots,N-1\}^2$ .

Si l'on note  $(\hat{w}_{k,l})_{(k,l) \in \{0, \dots, 2N-1\}^2}$ , la transformée de Fourier de  $\tilde{w}$ , on a (voir Proposition 2 et (1.7))

$$\begin{aligned} w_{m,n} &= \frac{1}{(2N)^2} \sum_{k,l=0}^{2N-1} \hat{w}_{k,l} e^{2i\pi \frac{km+ln}{2N}} \\ &= \frac{1}{(2N)^2} \sum_{k,l=0}^{2N-1} \hat{w}_{k,l} e^{2i\pi \frac{k(2N-1-m)+ln}{2N}} \\ &= \frac{1}{(2N)^2} \sum_{k,l=0}^{2N-1} \hat{w}_{k,l} e^{2i\pi \frac{km+l(2N-1-n)}{2N}} \\ &= \frac{1}{(2N)^2} \sum_{k,l=0}^{2N-1} \hat{w}_{k,l} e^{2i\pi \frac{k(2N-1-m)+l(2N-1-n)}{2N}}. \end{aligned}$$

En sommant ces égalités, on obtient

$$w_{m,n} = \frac{1}{(2N)^2} \sum_{k,l=0}^{2N-1} \hat{w}_{k,l} e^{-2i\pi \frac{k}{4N}} \cos\left(2\pi \frac{k(m+\frac{1}{2})}{2N}\right) e^{-2i\pi \frac{l}{4N}} \cos\left(2\pi \frac{l(n+\frac{1}{2})}{2N}\right), \quad (1.8)$$

car

$$\begin{aligned} A &= e^{2i\pi \frac{km+ln}{2N}} + e^{2i\pi \frac{k(2N-1-m)+ln}{2N}} + e^{2i\pi \frac{km+l(2N-1-n)}{2N}} + e^{2i\pi \frac{k(2N-1-m)+l(2N-1-n)}{2N}} \\ &= \left( e^{2i\pi \frac{km}{2N}} + e^{2i\pi \frac{k(2N-1-m)}{2N}} \right) \left( e^{2i\pi \frac{ln}{2N}} + e^{2i\pi \frac{l(2N-1-n)}{2N}} \right) \\ &= e^{-2i\pi \frac{k}{4N}} 2 \cos\left(2\pi \frac{k(m+\frac{1}{2})}{2N}\right) e^{-2i\pi \frac{l}{4N}} 2 \cos\left(2\pi \frac{l(n+\frac{1}{2})}{2N}\right). \end{aligned}$$

Enfin, on peut simplifier (1.8) en remarquant que, pour  $k \in \{0, \dots, N-1\}$  et tout  $m \in \{0, \dots, N-1\}$

$$\begin{aligned} \cos\left(2\pi \frac{(2N-k)(m+\frac{1}{2})}{2N}\right) &= -\cos\left(2\pi \frac{k(m+\frac{1}{2})}{2N}\right), \\ e^{-2i\pi \frac{(2N-k)}{4N}} \hat{w}_{2N-k,l} &= -e^{-2i\pi \frac{k}{4N}} \hat{w}_{k,l} \end{aligned}$$

et que

$$\cos\left(2\pi \frac{N(m+\frac{1}{2})}{2N}\right) = \cos(m\pi + \frac{\pi}{2}) = 0.$$

On a donc

$$\sum_{k=0}^{2N-1} \hat{w}_{k,l} e^{-2i\pi \frac{k}{4N}} \cos\left(2\pi \frac{k(m+\frac{1}{2})}{2N}\right) = \hat{w}_{0,l} \cos\left(2\pi \frac{0(m+\frac{1}{2})}{2N}\right) + \sum_{k=1}^{N-1} 2\hat{w}_{k,l} e^{-2i\pi \frac{k}{4N}} \cos\left(2\pi \frac{k(m+\frac{1}{2})}{2N}\right).$$

En faisant la même simplification pour la somme en  $l$ , on voit finalement que l'on peut exprimer  $w$  comme une somme de la forme

$$w_{m,n} = \frac{1}{(2N)^2} \sum_{k,l=0}^{N-1} c_{k,l} \cos\left(\pi \frac{k(m+\frac{1}{2})}{N}\right) \cos\left(\pi \frac{l(n+\frac{1}{2})}{N}\right), \quad (1.9)$$

pour des valeurs appropriées  $(c_{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$ .

Ceci conduit à considérer la base constituée des éléments  $(c^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$ , avec

$$c_{m,n}^{k,l} = \cos\left(\pi \frac{k(m+\frac{1}{2})}{N}\right) \cos\left(\pi \frac{l(n+\frac{1}{2})}{N}\right), \quad (1.10)$$

pour  $(m, n) \in \{0, \dots, N-1\}^2$ .

Le fait que pour tout  $w \in \mathbb{R}^{N^2}$  il existe  $(c_{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$  vérifiant (1.9) et que  $(c^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$  contienne  $N^2$  éléments nous garantit que  $(c^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$  est bien une base. Les coordonnées de  $w$  dans cette base pourraient être explicités en détaillant plus que nous ne l'avons fait les calculs menant à (1.9). (Il suffirait d'exprimer complètement les valeurs  $(c_{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$ .) Il n'est pas très difficile de voir que la base de cosinus ainsi obtenue est orthogonale (on pourrait facilement la normaliser).

On ne donnera pas plus de détails sur la base de cosinus. Il faut tout de même savoir que, comme la base de Fourier, il existe un algorithme de complexité  $N^2 \log N$  (pour une image  $N \times N$ ) permettant de calculer les coordonnées d'une image dans la base de cosinus. Cet algorithme s'appelle la "Discrete Cosine Transform" (DCT).

### 1.3.2 Les cosinus locaux

La base de cosinus que nous avons vue au chapitre précédent présente encore un gros défaut : Elle est globale. En effet, les fonctions (1.10) sont supportées par  $\{0, \dots, N-1\}^2$  tout entier. Ainsi, si on modifie une image sur zone (même très petite), toutes ses coordonnées, dans la base de cosinus, sont modifiées...

(On rappelle que pour une base orthogonale  $(e^{k,l})_{(k,l) \in \{0, \dots, N-1\}^2}$  de  $\mathbb{R}^{N^2}$ , la coordonnée  $\lambda_{k,l}$  d'une image  $v$  est

$$\lambda_{k,l} = \frac{1}{\alpha_{k,l}} \sum_{m,n=0}^{N-1} v_{m,n} e_{m,n}^{k,l},$$

avec

$$\alpha_{k,l} = \sum_{m,n=0}^{N-1} e_{m,n}^{k,l} e_{m,n}^{k,l}.$$

Une première approche (qui est utilisée par le standard de compression JPEG) consiste à découper notre image en sous-images de taille  $8 \times 8$  et à utiliser une base de cosinus sur chacune des sous-images. On a ainsi "localisé" la base de cosinus.

Il est clair que les fonctions ainsi obtenues forment une base orthogonale, car les fenêtres  $8 \times 8$  sont disjointes et il n'y a pas de trou entre les sous-images. (On ne considère pas ici le cas où la taille de l'image de départ n'est pas un multiple de 8.)

On remarque aussi l'importance de localiser une base de cosinus et non une base de Fourier. En effet, comme on utilise beaucoup de sous-images, les défauts que l'on observe sur les bords de l'image, lorsque l'on utilise la base de Fourier, auraient eu lieu aux bords des sous-images. C'est-à-dire partout... De même, les grands coefficients créés par la périodisation, lorsque l'on utilise la base de Fourier, auraient été présents pour chaque sous-image. Ils auraient ainsi été très nombreux.

## 1.4 Bases d'ondelettes discrètes

### 1.4.1 Objectif

Nous avons vu jusque-là des bases dont les éléments étaient soit très bien localisés en espace (la base canonique) soit très bien localisés en fréquence (la base de Fourier et la base de cosinus). La seule base "intermédiaire" est la base de cosinus locaux. Elle est à la fois localisée en espace (un coefficient ne dépend que des valeurs de l'image dans une sous-image de taille  $8 \times 8$ ) et en fréquence (un coefficient représente l'importance d'une fréquence dans une fenêtre  $8 \times 8$ ).

Les ondelettes (et paquets d'ondelettes) nous fourniront toute une palette pour les localisations en espace et en fréquence. La décomposition en ondelettes présente aussi l'avantage de représenter l'information contenue dans l'image de façon "multi-résolution". Intuitivement, cela reviendrait à choisir automatiquement la meilleure taille possible pour les sous-images pour la base de cosinus locaux. ( $8 \times 8$  semble en effet un peu arbitraire.)

Nous recommandons la lecture de [2, 3, 1], pour tout ce qui touche aux ondelettes.



### 1.4.2 Des banques de filtres aux ondelettes, en dimension 1

Les ondelettes sont en fait un cas particulier de banques de filtres ("filter banks", en anglais). Afin d'introduire ces notions, on va considérer le cas d'un signal.

Soit  $u \in \mathbb{R}^N$  (on suppose que  $N$  est pair),  $h, g, \tilde{h}, \tilde{g}$  des noyaux de convolution, définis sur  $\{0, \dots, N-1\}$ .

On note

$$u_m^1 = (\bar{h} * u)_{2m} \text{ et } u_m^2 = (\bar{g} * u)_{2m}, \text{ pour } m \in \{0, \dots, \frac{N}{2}\}.$$

On note ici  $\bar{h}_m = h_{-m}$  (on a dans le domaine de Fourier,  $\hat{h}_k = \hat{h}_k^*$ , où  $*$  désigne le nombre complexe conjugué.)

Ceci correspond à une décomposition.

Pour la recomposition, on note

$$\tilde{u}_m^1 = \begin{cases} u_{\frac{m}{2}}^1 & , \text{ si } m \text{ est pair,} \\ 0 & , \text{ si } m \text{ est impair,} \end{cases} \quad \text{et } \tilde{u}_m^2 = \begin{cases} u_{\frac{m}{2}}^2 & , \text{ si } m \text{ est pair,} \\ 0 & , \text{ si } m \text{ est impair,} \end{cases}$$

On note

$$v = \tilde{h} * \tilde{u}^1 + \tilde{g} * \tilde{u}^2 \quad (1.11)$$

qui est le résultat de la recomposition.

En résumé, on a défini deux opérations :

- Une décomposition : on a décomposé un signal de taille  $N$  en deux signaux de tailles  $\frac{N}{2}$ . Remarquez que cette décomposition est un opérateur linéaire.
- Une recomposition : on a construit un signal de taille  $N$  à partir de deux signaux de tailles  $\frac{N}{2}$ .

Remarquez que cette recomposition est un opérateur linéaire.

Nous allons maintenant voir sous quelles conditions ces opérateurs sont inverses l'un de l'autre.

Pour cela, nous allons voir comment ces opérateurs agissent sur les coefficients de Fourier de  $u$ . On a, en effet, en utilisant les analogues des Propositions 3 et 5 pour des signaux

$$\widehat{u^1}_k = \frac{1}{2} \left( \widehat{h}_k^* \widehat{u}_k + \widehat{h}_{k+\frac{N}{2}}^* \widehat{u}_{k+\frac{N}{2}} \right) \quad (1.12)$$

et

$$\widehat{u^2}_k = \frac{1}{2} \left( \widehat{g}_k^* \widehat{u}_k + \widehat{g}_{k+\frac{N}{2}}^* \widehat{u}_{k+\frac{N}{2}} \right), \quad (1.13)$$

pour  $k \in \{0, \dots, \frac{N}{2} - 1\}$ .

Pour la reconstruction, on a, pour  $k \in \{0, \dots, \frac{N}{2} - 1\}$ ,

$$\begin{aligned} \widehat{\tilde{u}^1}_k &= \sum_{m=0}^{N-1} \tilde{u}_m^1 e^{-2i\pi \frac{km}{N}} \\ &= \sum_{m=0}^{\frac{N}{2}-1} u_m^1 e^{-2i\pi \frac{2km}{N}} \\ &= \widehat{u^1}_k \end{aligned}$$

et

$$\begin{aligned} \widehat{\tilde{u}^1}_{k+\frac{N}{2}} &= \sum_{m=0}^{N-1} \tilde{u}_m^1 e^{-2i\pi \frac{(k+\frac{N}{2})m}{N}} \\ &= \sum_{m=0}^{\frac{N}{2}-1} u_m^1 e^{-2i\pi \frac{(2k+N)m}{N}} \\ &= \widehat{u^1}_k. \end{aligned}$$

On récrit ce résultat sous la forme

$$\widehat{\tilde{u}^1}_k = \widehat{u^1}_{k[\frac{N}{2}]},$$

où  $k \in \{0, \dots, N-1\}$  et  $k[\frac{N}{2}]$  signifie " $k$  modulo  $\frac{N}{2}$ ".

De même, on a, pour  $k \in \{0, \dots, N-1\}$ ,

$$\widehat{u^2}_k = \widehat{u^2}_{k[\frac{N}{2}]}.$$

On obtient donc finalement, pour tout  $k \in \{0, \dots, N-1\}$ ,

$$\begin{aligned} \hat{v}_k &= \widehat{\tilde{h}}_k \widehat{u^1}_k + \widehat{\tilde{g}}_k \widehat{u^2}_k \\ &= \frac{1}{2} \widehat{\tilde{h}}_k \left( \widehat{h}^*_k \widehat{u}_k + \widehat{h}^*_{k+\frac{N}{2}} \widehat{u}_{k+\frac{N}{2}} \right) + \frac{1}{2} \widehat{\tilde{g}}_k \left( \widehat{g}^*_k \widehat{u}_k + \widehat{g}^*_{k+\frac{N}{2}} \widehat{u}_{k+\frac{N}{2}} \right). \end{aligned}$$

Notez que ci-dessus, on étend la transformée de Fourier par périodisation. On utilise aussi le fait que : soit  $k[\frac{N}{2}] = k$ , soit  $k = k[\frac{N}{2}] + \frac{N}{2}$ .

On a enfin, pour tout  $k \in \{0, \dots, N-1\}$ ,

$$\hat{v}_k = \frac{1}{2} \left( \widehat{\tilde{h}}_k \widehat{h}^*_k + \widehat{\tilde{g}}_k \widehat{g}^*_k \right) \widehat{u}_k + \frac{1}{2} \left( \widehat{\tilde{h}}_k \widehat{h}^*_{k+\frac{N}{2}} + \widehat{\tilde{g}}_k \widehat{g}^*_{k+\frac{N}{2}} \right) \widehat{u}_{k+\frac{N}{2}}$$

Ceci nous assure que  $v = u$ , quelque soit le signal  $u$  de départ, si et seulement si, pour tout  $k \in \{0, \dots, N-1\}$ ,

$$\widehat{\tilde{h}}_k \widehat{h}^*_k + \widehat{\tilde{g}}_k \widehat{g}^*_k = 2 \quad (1.14)$$

et

$$\widehat{\tilde{h}}_k \widehat{h}^*_{k+\frac{N}{2}} + \widehat{\tilde{g}}_k \widehat{g}^*_{k+\frac{N}{2}} = 0 \quad (1.15)$$

Ces deux équations nous donnent les conditions reliant  $h$ ,  $g$ ,  $\tilde{h}$  et  $\tilde{g}$  sous lesquelles les décompositions et recompositions que nous avons décrites ci-dessus sont inverses l'une de l'autre. On appelle les filtres qui satisfont ces deux équations des "banques de filtres à reconstruction exacte". Pour de tels filtres, comme les décompositions/recompositions sont linéaires, ce sont des changements de base. En d'autres termes, les signaux  $u^1$  et  $u^2$  représentent les coordonnées du signal de départ dans une base.

La question qui se pose maintenant est de savoir comment choisir les filtres  $h$ ,  $g$ ,  $\tilde{h}$  et  $\tilde{g}$ , pour que la base soit intéressante.

Nous allons dans un premier temps voir comment simplifier (1.14) et (1.15) de manière à simplifier la construction des filtres  $h$ ,  $g$ ,  $\tilde{h}$  et  $\tilde{g}$ .

La dernière équation nous garantit que le vecteur  $(\widehat{\tilde{h}}_k, \widehat{\tilde{g}}_k)$  est orthogonal à  $(\widehat{h}_{k+\frac{N}{2}}, \widehat{g}_{k+\frac{N}{2}})$ . Il existe donc un nombre  $\lambda_k \in \mathbb{C}$ , tel que

$$\widehat{\tilde{h}}_k = \lambda_k^* \widehat{g}^*_{k+\frac{N}{2}} \text{ et } \widehat{\tilde{g}}_k = -\lambda_k^* \widehat{h}^*_{k+\frac{N}{2}}. \quad (1.16)$$

En utilisant (1.14), on obtient que

$$\widehat{g}_{k+\frac{N}{2}} \widehat{h}_k - \widehat{h}_{k+\frac{N}{2}} \widehat{g}_k \neq 0$$

et

$$\lambda_k = \frac{2}{\widehat{g}_{k+\frac{N}{2}} \widehat{h}_k - \widehat{h}_{k+\frac{N}{2}} \widehat{g}_k}. \quad (1.17)$$

On peut en déduire que

$$\lambda_{k+\frac{N}{2}} = -\lambda_k. \quad (1.18)$$

**Remarque :** Les équations (1.16) et (1.17) nous disent comment construire les filtres pour la reconstruction ( $\tilde{h}$  et  $\tilde{g}$ ) une fois que les filtres pour la décomposition ( $h$  et  $g$ ) sont fixés. Nous ne les utiliserons cependant pas en l'état. Nous allons plutôt utiliser (1.14) et (1.15) pour déduire  $g$  et  $\tilde{g}$  en fonction de  $h$  et  $\tilde{h}$ .

En utilisant la première égalité de (1.16) (à la fréquence  $k + \frac{N}{2}$ ) et (1.18), on trouve que

$$\begin{aligned} \widehat{g}^*_k &= \frac{1}{\lambda^*_{k+\frac{N}{2}}} \widehat{h}_{k+\frac{N}{2}} \\ &= -\frac{1}{\lambda^*_k} \widehat{h}_{k+\frac{N}{2}}. \end{aligned}$$

L'équation (1.14) devient alors

$$\begin{aligned} 2 &= \widehat{h}_k \widehat{h}_k^* + \left( -\lambda_k^* \widehat{h}_{k+\frac{N}{2}}^* \right) \left( -\frac{1}{\lambda_k} \widehat{h}_{k+\frac{N}{2}} \right) \\ &= \widehat{h}_k \widehat{h}_k^* + \widehat{h}_{k+\frac{N}{2}}^* \widehat{h}_{k+\frac{N}{2}}. \end{aligned}$$

Cette équation nous donne une condition que doivent satisfaire  $h$  et  $\tilde{h}$ . Si  $h$  et  $\tilde{h}$  sont fixés, on peut trouver  $g$  et  $\tilde{g}$  en utilisant

$$\widehat{g}_k = -\frac{1}{\lambda_k} \widehat{h}_{k+\frac{N}{2}}^* \text{ et } \widehat{\tilde{g}}_k = -\lambda_k^* \widehat{h}_{k+\frac{N}{2}}. \quad (1.19)$$

En pratique, on se contente de prendre

$$\lambda_k = -e^{2i\pi \frac{k}{N}}. \quad (1.20)$$

Ce choix peut sembler arbitraire mais n'est, en fait, pas une grosse restriction (voir le Théorème 7.8, dans [2]).

Ceci nous donne, pour tout  $m \in \{0, \dots, N-1\}$

$$\begin{aligned} g_m &= \frac{1}{N} \sum_{k=0}^{N-1} \widehat{g}_k e^{2i\pi \frac{km}{N}} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} e^{-2i\pi \frac{k}{N} \widehat{h}_{k+\frac{N}{2}}^*} e^{2i\pi \frac{km}{N}} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \widehat{h}_k^* e^{2i\pi \frac{(k-\frac{N}{2})(m-1)}{N}} \\ &= e^{-i\pi(m-1)} \left( \frac{1}{N} \sum_{k=0}^{N-1} \widehat{h}_k^* e^{2i\pi \frac{k(m-1)}{N}} \right) \\ &= (-1)^{(1-m)} \left( \frac{1}{N} \sum_{k=0}^{N-1} \widehat{h}_k e^{2i\pi \frac{k(1-m)}{N}} \right)^* \\ &= (-1)^{(1-m)} \tilde{h}_{1-m}. \end{aligned}$$

Car  $\tilde{h}$  est à valeur dans  $\mathbb{R}$ . On a de même

$$\tilde{g}_m = (-1)^{(1-m)} h_{1-m}.$$

### 1.4.3 Les ondelettes discrètes en dimension 1

#### Ondelettes biorthogonales

**Théorème 1** Soit  $h$  et  $\tilde{h} \in \mathbb{R}^N$  (on les suppose périodisés en dehors de  $\{0, \dots, N-1\}$ ), tels que

$$\widehat{h}_k^* \widehat{h}_k + \widehat{h}_{k+\frac{N}{2}}^* \widehat{h}_{k+\frac{N}{2}} = 2. \quad (1.21)$$

On note

$$g_m = (-1)^{(1-m)} \tilde{h}_{1-m} \text{ et } \tilde{g}_m = (-1)^{(1-m)} h_{1-m}, \quad (1.22)$$

pour  $m \in \{0, \dots, N-1\}$ .

L'ensemble  $\{\tilde{h}_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{\tilde{g}_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}}$  est une base de  $\mathbb{R}^N$ .

De même, l'ensemble  $\{h_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{g_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}}$  est une base de  $\mathbb{R}^N$ . On appelle ces bases "bases d'ondelettes biorthogonales".

*Preuve.* En fait, ce résultat découle de ce que, sous les hypothèses du théorème,  $h$ ,  $\tilde{h}$ ,  $g$  et  $\tilde{g}$  forment une banque de filtres à reconstruction exacte. En effet, étant donné (1.22), on peut combiner les équations (1.19) et (1.20) pour obtenir que, pour tout  $k \in \{0, \dots, N-1\}$ ,

$$\hat{g}_k = e^{-2i\pi \frac{k}{N}} \hat{h}_{k+\frac{N}{2}}^* \text{ et } \hat{g}_k = e^{-2i\pi \frac{k}{N}} \hat{h}_{k+\frac{N}{2}}^*.$$

L'hypothèse (1.21) nous garantit alors que

$$\begin{aligned} 2 &= \hat{h}_k^* \hat{h}_k + \hat{h}_{k+\frac{N}{2}}^* \hat{h}_{k+\frac{N}{2}} \\ &= \hat{h}_k^* \hat{h}_k + \hat{g}_k^* \hat{g}_k. \end{aligned}$$

Ce qui correspond à la condition (1.14) des banques de filtres à reconstruction exacte. On a aussi, pour tout  $k \in \{0, \dots, N-1\}$ ,

$$\begin{aligned} \hat{h}_k \hat{h}_{k+\frac{N}{2}}^* + \hat{g}_k \hat{g}_{k+\frac{N}{2}}^* &= \hat{h}_k \hat{h}_{k+\frac{N}{2}}^* + \left( e^{-2i\pi \frac{k}{N}} \hat{h}_{k+\frac{N}{2}}^* \right) \left( e^{2i\pi \frac{k+\frac{N}{2}}{N}} \hat{h}_k \right) \\ &= \hat{h}_k \hat{h}_{k+\frac{N}{2}}^* - \hat{h}_k \hat{h}_{k+\frac{N}{2}}^* \\ &= 0, \end{aligned}$$

qui correspond à la condition (1.15) des banques de filtres à reconstruction exacte.

Enfin,  $h$ ,  $\tilde{h}$ ,  $g$  et  $\tilde{g}$  forment une banque de filtres à reconstruction exacte et donc, quelque soit  $u \in \mathbb{R}^N$ , (1.11) peut s'écrire

$$u_m = \sum_{l=0}^{\frac{N}{2}-1} u_l^1 \tilde{h}_{m-2l} + \sum_{l=0}^{\frac{N}{2}-1} u_l^2 \tilde{g}_{m-2l},$$

pour tout  $m \in \{0, \dots, N-1\}$ . Ceci signifie que l'ensemble  $\{\tilde{h}_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{\tilde{g}_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}}$  est une base de  $\mathbb{R}^N$ .

Il est aussi important de noter que l'on peut interchanger, dans (1.21), les rôles de  $h$  et  $\tilde{h}$ . Ceci nous assure que le deuxième ensemble est une base.  $\square$

Dans la preuve ci-dessus, nous ne justifions pas le terme "biorthogonal". Celui-ci a en fait un sens précis est lie les deux bases : celle avec le  $\tilde{\cdot}$  et celle sans le  $\tilde{\cdot}$ .

En fait, on précise en général que la base  $\{\tilde{h}_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{\tilde{g}_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}}$  (même chose pour l'autre base) est une base d'ondelettes biorthogonale "de niveau 1".

En effet, en général, si  $N = 2^K$ , on peut redécomposer  $u^1$  pour obtenir  $(u^1)^1$ , que l'on redécompose à son tour et ainsi de suite. On appelle "niveau de décomposition" le nombre de fois que l'on a décomposé avec  $h$ . On peut décomposer tant que la taille du signal est plus grande que la taille des filtres  $\tilde{h}$  et  $\tilde{g}$ .

La base de niveau  $J$ , prend alors la forme

$$\{\tilde{h}_{m-2^J l}^J\}_{l \in \{0, \dots, \frac{N}{2^J}-1\}} \cup \{\tilde{g}_{m-2^J l}^J\}_{l \in \{0, \dots, \frac{N}{2^J}-1\}} \cup \dots \cup \{\tilde{g}_{m-2l}^1\}_{l \in \{0, \dots, \frac{N}{2}-1\}}. \quad (1.23)$$

La localisation en espace (on dit aussi en temps, pour des signaux) des éléments de notre base est illustrée sur la figure 1.10.

On utilise généralement deux notations pour désigner les coefficients d'ondelettes (i.e. les coordonnées dans une base d'ondelettes). Le but est de distinguer s'il s'agit d'un coefficient correspondant à un élément  $\tilde{h}^J$ , ou s'il s'agit d'une coordonnée correspondant à un élément  $\tilde{g}^J$ . On note

$$\{a_l^J\}_{l \in \{0, \dots, \frac{N}{2^J}-1\}}$$

les coordonnées suivant les  $\{\tilde{h}_{m-2^J l}^J\}_{l \in \{0, \dots, \frac{N}{2^J}-1\}}$ , et

$$\{d_l^J\}_{l \in \{0, \dots, \frac{N}{2^J}-1\}},$$

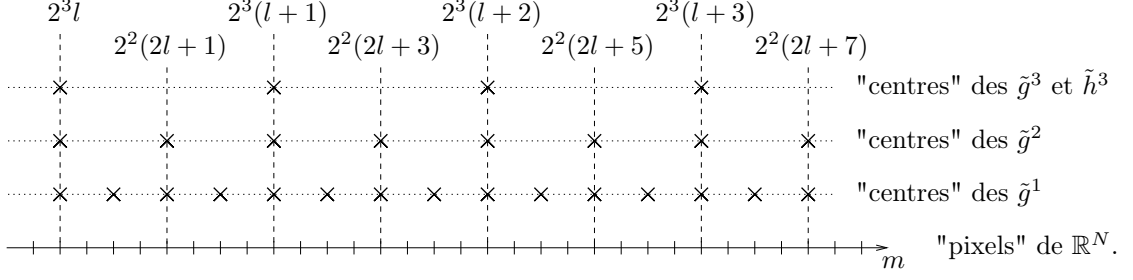


FIG. 1.10 – Exemple de localisation en espace d'une base d'ondelettes de niveau 3.

celles selon  $\{\tilde{g}_{m-2^j l}^j\}_{l \in \{0, \dots, \frac{N}{2^j}-1\}}$ , pour  $j \in \{1, \dots, J\}$ . (Cette notation vient de *average*, pour  $a_l^j$ , et *details*, pour  $d_l^j$ .)

Les relations entre  $\tilde{h}^j$ , les  $\{\tilde{g}^j\}_{1 \leq j \leq J}$ ,  $\tilde{h}$  et  $\tilde{g}$  peuvent ne pas être simple. Dans le cas où  $h$  et  $g$  ont un support de petite taille, on a par exemple

$$\begin{aligned} (\widehat{u^1})_k^1 &= \frac{1}{2} \left( \widehat{h_{2k}^*} \widehat{u_k^1} + \widehat{h_{2k+\frac{N}{2}}^*} \widehat{u_{k+\frac{N}{4}}^1} \right) \\ &= \frac{1}{4} \left( \widehat{h_{2k}^*} \left( \widehat{h_k^*} \widehat{u_k} + \widehat{h_{k+\frac{N}{2}}^*} \widehat{u_{k+\frac{N}{2}}} \right) + \widehat{h_{2k+\frac{N}{2}}^*} \left( \widehat{h_{k+\frac{N}{4}}^*} \widehat{u_{k+\frac{N}{4}}} + \widehat{h_{k+\frac{3N}{4}}^*} \widehat{u_{k+\frac{3N}{4}}} \right) \right) \\ &= \frac{1}{4} \left( \widehat{h_{2k}^*} \widehat{h_k^*} \widehat{u_k} + \widehat{h_{2k+\frac{N}{2}}^*} \widehat{h_{k+\frac{N}{4}}^*} \widehat{u_{k+\frac{N}{4}}} + \widehat{h_{2k}^*} \widehat{h_{k+\frac{N}{2}}^*} \widehat{u_{k+\frac{N}{2}}} + \widehat{h_{2k+\frac{N}{2}}^*} \widehat{h_{k+\frac{3N}{4}}^*} \widehat{u_{k+\frac{3N}{4}}} \right), \end{aligned}$$

pour  $k \in \{0, \frac{N}{4} - 1\}$ . Ceci nous assure que

$$\begin{aligned} \widehat{h_k^2}^* &= \widehat{h_{2k}^*} \widehat{h_k^*} \\ \widehat{h_{k+\frac{N}{4}}^2}^* &= \widehat{h_{2k+\frac{N}{2}}^*} \widehat{h_{k+\frac{N}{4}}^*} \\ \widehat{h_{k+\frac{N}{2}}^2}^* &= \widehat{h_{2k}^*} \widehat{h_{k+\frac{N}{2}}^*} \\ \widehat{h_{k+\frac{3N}{4}}^2}^* &= \widehat{h_{2k+\frac{N}{2}}^*} \widehat{h_{k+\frac{3N}{4}}^*} \end{aligned}$$

pour  $k \in \{0, \dots, \frac{N}{4} - 1\}$ , où  $h^2$  est tel que

$$(u^1)_m^1 = (\overline{h^2} * u)_{4m}.$$

En résumé, on sait que, pour  $k \in \{0, \dots, N-1\}$ ,

$$\widehat{h_k^2} = \widehat{h_{2k}} \widehat{h_k}. \quad (1.24)$$

On a, de même, pour  $k \in \{0, \dots, N-1\}$ ,

$$\widehat{g_k^2} = \widehat{g_{2k}} \widehat{g_k}, \quad (1.25)$$

où  $g^2$  est tel que, pour tout  $m \in \{0, \dots, \frac{N}{4} - 1\}$ ,

$$(u^1)_m^2 = (\overline{g^2} * u)_{4m}.$$

On voit bien sur (1.24) et (1.25), que les termes  $\widehat{h_{2k}}$  et  $\widehat{g_{2k}}$  permettent d'avoir une localisation en fréquence de la base d'ondelettes de plus en plus fine pour les basses fréquences. Par exemple, si  $h$  est un filtre passe bas (ce qui est généralement le cas),  $h^2$  "laisse passer" deux fois moins de fréquences que  $h$ .

Plus généralement, si on note  $h^J$  et  $\{g^j\}_{1 \leq j \leq J}$  les filtres permettant d'obtenir les coordonnées d'un élément de  $\mathbb{R}^N$  dans la base d'ondelettes biorthogonale (voir (1.23)), on a, pour  $k \in \{0, \dots, N-1\}$ ,

$$\widehat{h_k^J} = \widehat{h_{2^J k}} \widehat{h_{2^{J-1} k}} \dots \widehat{h_{2k}} \widehat{h_k},$$

et, pour  $j \in \{1, \dots, J\}$ ,

$$\widehat{g}_k^j = \widehat{g}_{2^j k} \widehat{h}_{2^{j-1} k} \dots \widehat{h}_{2k} \widehat{h}_k.$$

Dans le cas où  $h$  est un filtre passe bas, localisé en fréquence sur  $[-\frac{N}{4}, \frac{N}{4}]$ , on obtient le découpage fréquentiel illustré sur la Figure 1.11. On parle d'une base "multi-échelle" (*multiscale*, en anglais).

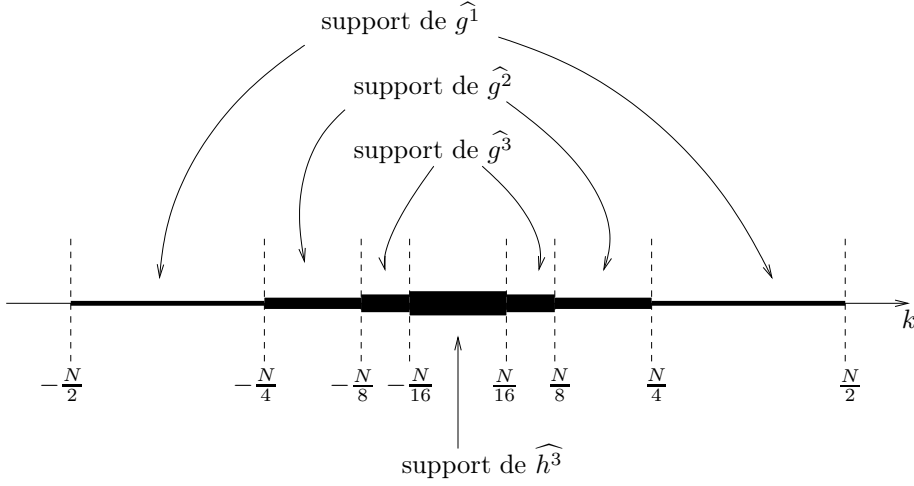


FIG. 1.11 – Exemple de localisation fréquentielle d'une base d'ondelettes de niveau 3.

### Ondelettes orthogonales

**Théorème 2** Soit  $h \in \mathbb{R}^N$  (on suppose  $h$  périodisé en dehors de  $\{0, \dots, N-1\}$ ) tel que

$$|\widehat{h}_k|^2 + |\widehat{h}_{k+\frac{N}{2}}|^2 = 2. \quad (1.26)$$

On note

$$g_m = (-1)^{(1-m)} h_{1-m},$$

pour  $m \in \{0, \dots, N-1\}$ .

L'ensemble  $\{h_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{g_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}}$  est une base orthogonale de  $\mathbb{R}^N$ . On appelle cette base "base d'ondelettes orthogonale".

*Preuve.* Le fait que  $\{h_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{g_{m-2l}\}_{l \in \{0, \dots, \frac{N}{2}-1\}}$  soit une base est une conséquence du Théorème 1. Il suffit de l'appliquer avec  $\tilde{h} = h$ .

Soit  $l \neq l'$ , dans  $\{0, \dots, \frac{N}{2}-1\}$ , pour montrer que deux termes  $h_{m-2l}$  et  $h_{m-2l'}$  sont orthogonaux, on remarque d'abord que

$$\widehat{\tau_{2l} h}_k = e^{-2i\pi \frac{2kl}{N}} \widehat{h}_k,$$

pour  $k \in \{0, \dots, N-1\}$  et

$$(\tau_{2l} h)_m = h_{m-2l},$$

la translaté de  $2l$  de  $h$ .

On a alors (en adaptant la Proposition 4), que

$$\begin{aligned}
\sum_{m=0}^{N-1} h_{m-2l} h_{m-2l'} &= \frac{1}{N} \sum_{k=0}^{N-1} e^{-2i\pi \frac{2kl}{N}} \hat{h}_k e^{2i\pi \frac{2kl'}{N}} \hat{h}_k^* \\
&= \frac{1}{N} \sum_{m=0}^{\frac{N}{2}-1} |\hat{h}_k|^2 e^{-2i\pi \frac{2k(l-l')}{N}} + |\hat{h}_{k+\frac{N}{2}}|^2 e^{-2i\pi \frac{(2k+N)(l-l')}{N}} \\
&= \frac{2}{N} \sum_{m=0}^{\frac{N}{2}-1} e^{-2i\pi \frac{2k(l-l')}{N}} \\
&= 0,
\end{aligned}$$

car  $l \neq l'$ .

On montre de même les autres relations d'orthogonalités. (Pour mémoire,  $\hat{g}_k = e^{-2i\pi \frac{k}{N}} \hat{h}_{k+\frac{N}{2}}^*$ )  $\square$

On fait la même construction que pour les bases biorthogonales, pour obtenir des bases d'ondelettes orthogonales de niveau supérieur à 1.

Les localisations en espace et fréquence d'une base d'ondelettes orthogonale sont similaires à celles d'une base d'ondelettes biorthogonale.

#### 1.4.4 Les ondelettes discrètes en dimension 2

Le passage à la dimension deux ne pose pas de réelles difficultés. On le fait par un produit tensoriel. On obtient alors des bases de  $\mathbb{R}^{N^2}$  constituées d'éléments du type

$$\begin{aligned}
&\left\{ \{\tilde{h}_{m-2l}^J\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \{\tilde{g}_{m-2l}^J\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \cup \dots \cup \{\tilde{g}_{m-2l}^1\}_{l \in \{0, \dots, \frac{N}{2}-1\}} \right\} \\
&\quad \times \left\{ \{\tilde{h}_{n-2l'}^J\}_{l' \in \{0, \dots, \frac{N}{2}-1\}} \cup \{\tilde{g}_{n-2l'}^J\}_{l' \in \{0, \dots, \frac{N}{2}-1\}} \cup \dots \cup \{\tilde{g}_{n-2l'}^1\}_{l' \in \{0, \dots, \frac{N}{2}-1\}} \right\},
\end{aligned}$$

pour  $(m, n) \in \mathbb{R}^{N^2}$ .

Par exemple, pour la base de niveau 1, on a

$$\begin{aligned}
&\left\{ \tilde{h}_{m-2l} \tilde{h}_{n-2l'} \right\}_{(l, l') \in \{0, \dots, \frac{N}{2}-1\}^2} \cup \left\{ \tilde{h}_{m-2l} \tilde{g}_{n-2l'} \right\}_{(l, l') \in \{0, \dots, \frac{N}{2}-1\}^2} \\
&\quad \cup \left\{ \tilde{g}_{m-2l} \tilde{h}_{n-2l'} \right\}_{(l, l') \in \{0, \dots, \frac{N}{2}-1\}^2} \cup \left\{ \tilde{g}_{m-2l} \tilde{g}_{n-2l'} \right\}_{(l, l') \in \{0, \dots, \frac{N}{2}-1\}^2}.
\end{aligned}$$

On obtient donc quatre types d'éléments. Dans le cas où  $h$  est un filtre passe bas, on a un élément bas-bas, un bas-haut, un haut-bas, un haut-haut. Les localisations fréquentielles respectives de ces éléments sont représentées sur la Figure 1.12. Pour des bases de niveaux supérieurs, on redécoupe le carré correspondant à la partie bas-bas en suivant le même motif.

Algorithmiquement, on calcule la transformée en ondelettes 2 D en calculant toutes les transformées en ondelettes 1 D des lignes de l'image, puis toutes les transformées en ondelettes 1D des colonnes des résultats (les parties basses et hautes fréquences). On recommence éventuellement sur la partie correspondant aux basses fréquences pour obtenir une base de niveau supérieur.

Les coefficients d'ondelettes de l'image "barbara", dans une base d'ondelettes de niveau 1, sont représentés sur la Figure 1.13.

Pour obtenir la décomposition dans la base de niveau 2 (voir Figure 1.14), on a simplement décomposé la partie correspondant aux basses fréquences (appelée "résumé" en Français et *remainder* en anglais).

On utilise souvent des bases d'ondelettes de niveau important. Un exemple de décomposition au niveau 5 est illustré sur la Figure 1.15.

**Remarque :** Le contraste des coefficients d'ondelettes correspondant aux détails a été augmenté sur Figures 1.13, 1.14 et 1.15. Ceci permet de faire ressortir les coefficients correspondant aux détails. La partie correspondant aux détails aurait autrement été presque uniformément noire.

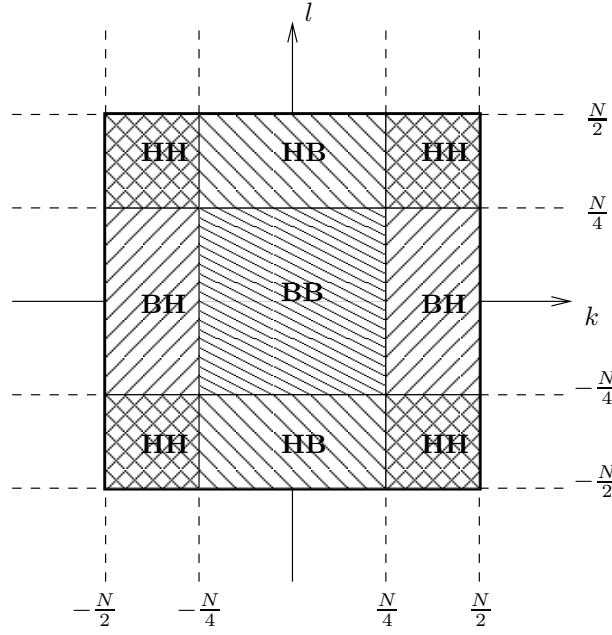


FIG. 1.12 – Localisation fréquentielle d'une base d'ondelettes de niveau 1. La localisation de bas-bas correspond à la partie **BB**, celle de haut-bas à **HB**, celle de bas-haut à **BH**, celle de haut-haut à **HH**.

### 1.4.5 Choisir une ondelette

Une question que nous n'avons pas abordée pour le moment est la question du choix des filtres  $h$  et (éventuellement)  $\tilde{h}$ . Ce choix dépend évidemment de l'application pour laquelle on utilise la base d'ondelettes. En général (notamment pour la compression et le débruitage d'images), le but est de choisir ce(s) filtre(s) de manière à représenter les images avec le moins de coefficients possible.

Le choix du ou des filtres est généralement un compromis entre le nombre de coefficients nécessaire pour représenter une "zone régulière" de l'image et les coefficients intervenant dans la représentation des bords séparant deux zones régulières.

#### Le nombre de moments nuls

Pour comprendre combien d'éléments de la base d'ondelettes vont intervenir dans la représentation d'une zone régulière, il faut être capable de caractériser la régularité de cette zone. Malheureusement, cela n'est pas possible dans le cadre de bases d'ondelettes discrètes. L'analyse est donc faite dans le cas de bases d'ondelettes de fonctions définies sur  $\mathbb{R}$  et nous ne rentrerons pas dans ces détails (voir [2]).

L'intuition derrière cette analyse est liée au fait que, pour une fonction régulière en un point  $x$ , on peut l'approximer, au voisinage de  $x$ , par un polynôme. La propriété d'avoir " $p$  moments nuls", où  $p$  est un entier positif, fait en sorte que les éléments de la base représentant les détails fins de l'image soient orthogonaux aux polynômes de degré inférieur à  $p$ .

Ainsi, dans le cas (simpliste) où une image contient une large zone  $Z$  dans laquelle elle vaut la même chose qu'un polynôme de degré inférieur à  $p$ , pour toutes les échelles (les indices  $j$ ) et les localisations (les indices  $l$ ), pour lesquelles le support de  $g_{\cdot, -2^j l}^j$  est inclus dans  $Z$ , la coordonnée  $d_l^j$  est nulle. Intuitivement, les indices  $(j, l)$  concernés forment un cône : Si le support de  $g_{\cdot, -2^j l}^j$  est inclus dans  $Z$ , les supports de  $g_{\cdot, -2^{j-1} 2l}^{j-1}$  et  $g_{\cdot, -2^{j-1} (2l+1)}^{j-1}$  sont inclus dans  $Z$ , et ainsi de suite. (C'est évidemment approximatif.)

Par ailleurs, on peut montrer mathématiquement que la propriété d'avoir plusieurs moments nuls est équivalente au fait que le filtre  $h$  soit tel que  $\hat{h}$  "aille rapidement à 0" en  $\pi$ . (Dans le cas des bases d'ondelettes de  $\mathbb{R}^N$  telles que nous les avons vues, il s'agit de la fréquence  $\frac{N}{2}$ .) Cette propriété n'est en fait pas très difficile à obtenir. Il faut d'abord remarquer que, pour des ondelettes, la décroissance vers 0 de  $\hat{h}$  en  $\pi$  est la même que celle de  $\hat{g}$  en 0. Le résultat vient alors simplement du fait que la transformée





FIG. 1.13 – Coefficients d'ondelettes pour une base d'ondelettes de niveau 1.

de Fourier d'un polynôme, défini sur  $\mathbb{R}$ , est "concentrée sur la fréquence 0". Ce résultat fait cependant intervenir des outils mathématiques (les distributions) que nous n'aborderons pas dans ce cours.

Ainsi, le fait d'avoir beaucoup de moments nuls (et donc que notre base représente efficacement les zones où l'image est régulière) correspond au fait que  $h$  soit bien localisé sur les basses fréquences.

Une ondelette est souvent associée à un nombre de moments nuls : il représente l'aptitude de la base à bien représenter des images très régulières (avec peu de bords nets).

### La taille du support de l'ondelette

En suivant l'analyse faite ci-dessus, on se rend compte que les zones où l'image est régulière sont essentiellement représentées par le résumé de la décomposition en ondelettes. En première analyse, les coefficients d'ondelettes correspondant aux détails interviennent donc pour représenter les bords séparant les zones régulières. Si on note  $B$  les pixels du bord entre deux zones régulières, les  $g_{-2^{j-1}2l}^{j-1}$  intersectant  $B$  vont produire des "forts" coefficients. À une échelle donnée ( $j$  est fixé), il y aura d'autant plus d'indices  $l$ , pour lesquels  $g_{-2^{j-1}2l}^j$  intersecte  $B$ , que le support de  $g^j$  est grand. Or la taille du support de  $g^j$  croît avec les tailles des supports de  $g$  et de  $h$ . (Il va de soit que la taille du support de  $g$  est la même que celle du support de  $\tilde{h}$  ou  $h$  (voir les Théorèmes 1 et 2.)

Il semble donc naturel de chercher des filtres  $h$  et  $\tilde{h}$  (ou  $h$ , pour des ondelettes orthogonales) ayant un support de petite taille. Ceci permet d'avoir une représentation efficace des bords présents dans l'image.

Cette propriété est (explicitement) une condition sur la localisation en espace des éléments de la base d'ondelettes.

### Conclusion

Le fait d'avoir un support de petite taille est en fait contradictoire avec le fait d'avoir beaucoup de moments nuls. Il faudra ainsi choisir un compromis entre ces deux propriétés. Ce compromis dépend du type d'images que l'on souhaite traiter.

- Si elles contiennent des grandes zones très régulières, on préférera des ondelettes ayant beaucoup de moments nuls et un support un peu grand.

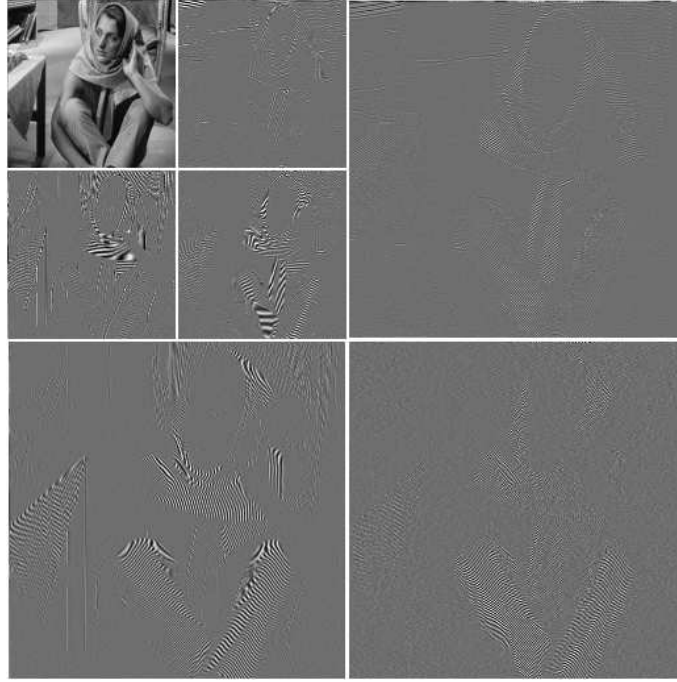


FIG. 1.14 – Coefficients d'ondelettes pour une base d'ondelettes de niveau 2.

- Si elles contiennent beaucoup de petites zones régulières, on préférera des ondelettes avec un petit support et peu de moments nuls.

Souvent un bon compromis est obtenu pour une ondelette ayant 3 ou 4 moments nuls.

Il existe dans la littérature beaucoup d'ondelettes (Daubechies, splines, Meyer, Haar,...). Les ondelettes de Daubechies connaissent un grand succès, car elles ont la propriété d'avoir la plus petite taille possible, étant donné un nombre de moments nuls. Les bases d'ondelettes sont données explicitement par les coefficients des filtres  $h$  et, si besoin,  $\tilde{h}$  (voir [2]).

## 1.5 Les paquets d'ondelettes

### 1.5.1 Principe des paquets d'ondelettes en dimension 1

À la base de la décomposition en ondelettes, il y a :

- un algorithme permettant de décomposer une suite finie de  $\mathbb{R}^N$  en deux suites finies de  $\mathbb{R}^{\frac{N}{2}}$  ;
- un algorithme permettant de recomposer deux suites finies de  $\mathbb{R}^{\frac{N}{2}}$  en une suite finie de  $\mathbb{R}^N$ .

Ces algorithmes sont inverses l'un de l'autre. Ils sont décrits au début de la section 1.4.2. Pour la décomposition en ondelette, on fait le choix de toujours décomposer le “résumé”. C'est à dire que, parmi les deux signaux obtenus après une décomposition, on décompose le signal représentant les basses fréquences. Ceci est itéré de manière récursive jusqu'à atteindre un niveau de décomposition donné.

Avec cette construction, une décomposition en ondelettes est caractérisée par :

- un filtres  $h$  (resp<sup>t</sup> deux filtres  $h$  et  $\tilde{h}$ ) dans le cas d'une base d'ondelettes orthogonale (resp<sup>t</sup> biorthogonale) ;
- un niveau de décomposition.

Dans le cadre des paquets d'ondelettes, on utilise la même décomposition/recomposition de signaux mais on s'autorise, à chaque niveau, à décomposer le signal représentant les basses fréquence et/ou le signal représentant les hautes fréquences.

La décomposition en paquets d'ondelettes est donc caractérisée par :

- un filtres  $h$  (resp<sup>t</sup> deux filtres  $h$  et  $\tilde{h}$ ) dans le cas d'une base de paquets d'ondelettes orthogonale (resp<sup>t</sup> biorthogonale) ;

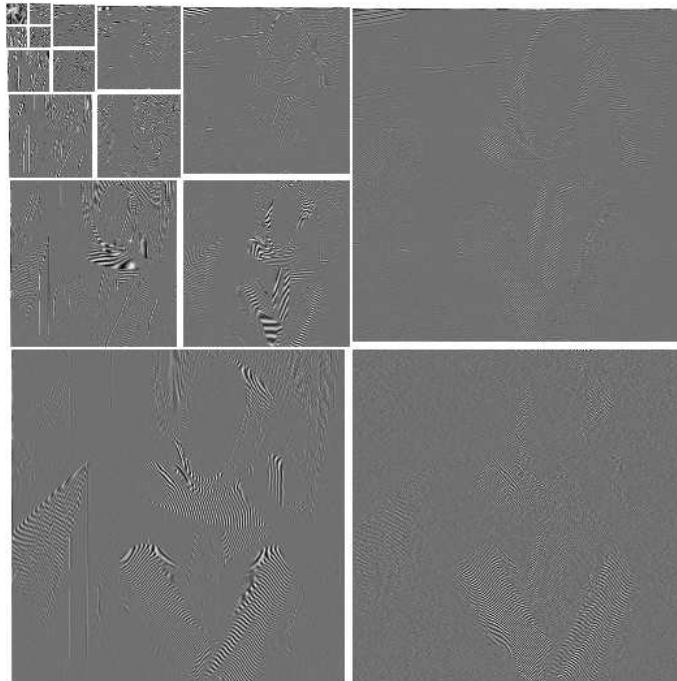


FIG. 1.15 – Coefficients d'ondelettes pour une base d'ondelettes de niveau 5.

- un arbre décrivant les décompositions à effectuer (voir la section 1.5.2, pour les détails).

Bien entendu, comme chaque décomposition est linéaire et inversible, son résultat correspond à des coordonnées du signal de départ dans une base. Faire successivement plusieurs décomposition revient à composer plusieurs changement de bases. C'est donc toujours un changement de base.

Par ailleurs, si tous les changements de bases correspondent à des bases orthogonales, la composition de ces changements de bases orthogonales est toujours un changement de base orthogonale.

Ainsi, on pourra toujours parler de bases orthogonales ou biorthogonales de paquets d'ondelettes. De plus, les conditions sur les filtres  $h$  et  $\tilde{h}$  pour obtenir des bases orthogonales et biorthogonales sont les mêmes que pour les bases d'ondelettes (voir (1.21) et (1.26)).

### 1.5.2 Les arbres de paquets d'ondelettes

En dimension 1, à chaque fois que l'on décompose un signal on obtient deux signaux. Pour décrire une décomposition un noeud à donc soit

- 0 fils : Il n'est pas décomposé
- 2 fils : Il est décomposé. Dans ce cas, chacun des fils représente un signal décomposé. On peut par exemple adopter la convention suivante :
  - le fils de gauche correspond au signal obtenu en décomposant à l'aide du filtre  $h$  (i.e. basses fréquences) ;
  - le fils de droite correspond au signal obtenu en décomposant à l'aide du filtre  $g$  (i.e. hautes fréquences).

Ces deux sortes de noeuds sont représentées sur la Figure 1.16.

La racine de l'arbre correspond au signal avant toute décomposition.

Ainsi, avec cette façon de définir les arbres, on peut décrire toutes les bases de paquets d'ondelettes. Par exemple, une base d'ondelettes de niveau 3 est une base de paquets d'ondelettes utilisant l'arbre représenté sur la Figure 1.17.

Nous avons évidemment beaucoup de possibilités pour le choix de l'arbre. Ceux-ci peuvent être adaptés à la nature du signal (ou de l'image) et à une application. Comme pour les ondelettes on comprend et interprète une décomposition en paquets d'ondelettes en analysant sa localisation fréquentielle et spatial.

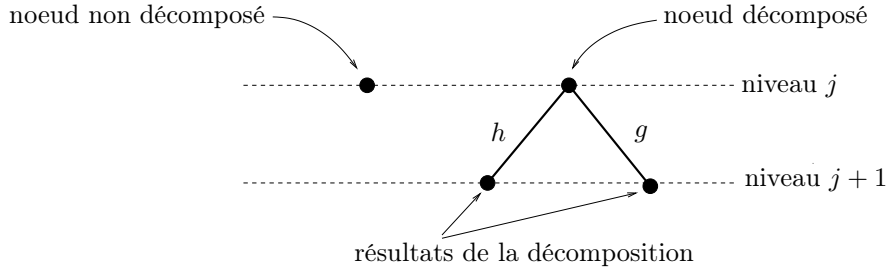


FIG. 1.16 – Les deux sortes de noeuds possibles dans un arbres de paquets d'ondelettes en dimension 1.

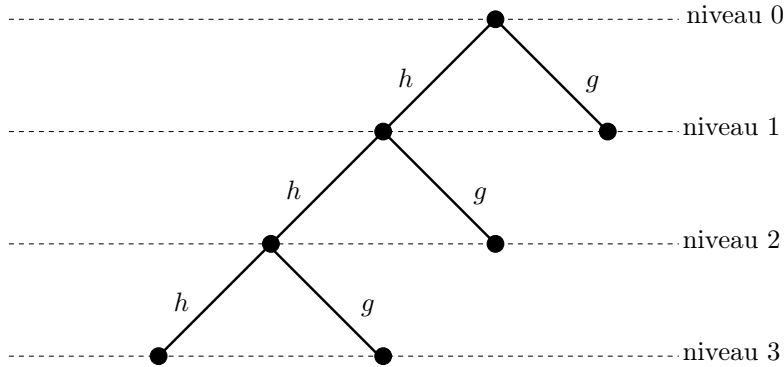


FIG. 1.17 – Arbre décrivant une décomposition en paquet d'ondelette. La décomposition décrite par cet arbre correspond à une décomposition en ondelette de niveau 3.

Nous verrons dans la section suivante comment faire cette analyse.

On parle d'un arbre de paquets d'ondelettes complètement décomposé jusqu'au niveau  $k$  pour un arbre dont tous les noeuds sont décomposés jusqu'au niveau  $k$ . Un arbre complètement décomposé jusqu'au niveau 2 est représenté sur la Figure 1.18.

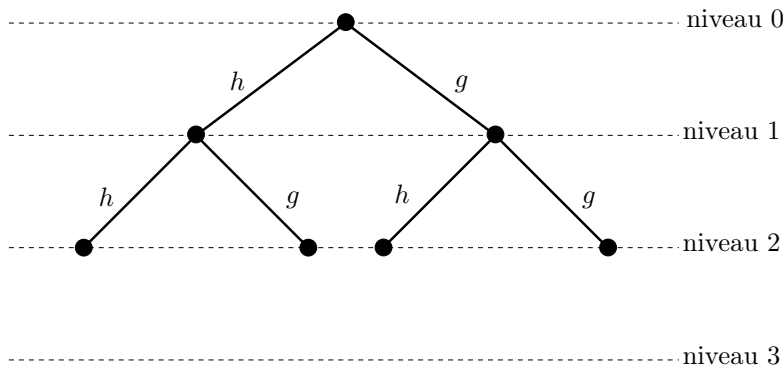


FIG. 1.18 – Arbre décrivant une base de paquets d'ondelettes complètement décomposée jusqu'au niveau 2.

On sera amené à considérer un arbre miroir d'ondelettes. C'est un arbre dans lequel on décompose successivement le signal correspondant aux hautes fréquences. Cet arbre permet de décomposer le plus possible les hautes fréquences et le moins possible les basses fréquences. il est utile pour certaines déconvolution. Un arbre miroir d'ondelette de niveau 4 est représenté sur la Figure 1.19. La raison pour laquelle cet arbre a cette forme devrait être claire après la lecture de la Section 1.5.3.

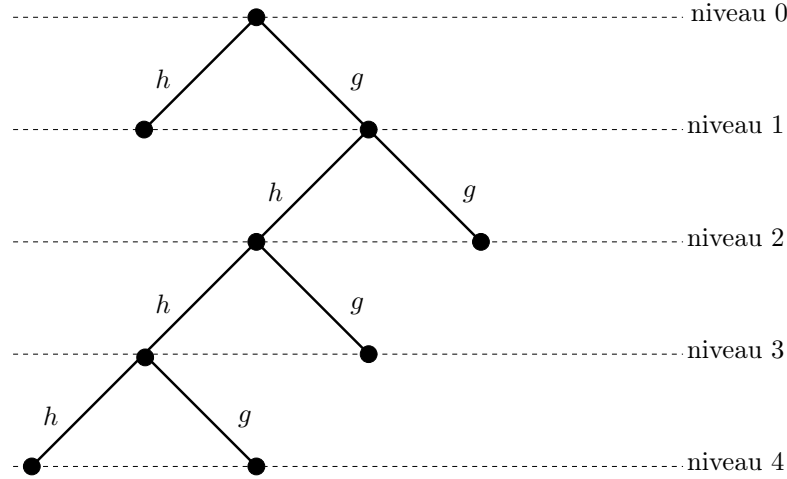


FIG. 1.19 – Arbre décrivant une base de paquets d'ondelettes: arbre miroir d'ondelette de niveau 4.

Dans le cas d'images, on a un “quadtrees”<sup>2</sup> dont chaque noeud a exactement 0 ou 4 fils. En dimension 3, on a un “octree”<sup>3</sup> dont chaque noeud a 0 ou 8 noeuds. Ainsi de suite, en dimension  $n$ , on caractérise une base de paquets d'ondelettes par des arbres dont chaque noeud a exactement 0 ou  $2^n$  noeuds.

On peut faire, en dimension quelconque, les mêmes constructions d'arbre d'ondelettes, d'arbres complètement décomposés, d'arbre miroir d'ondelettes qu'en dimension 1.

### 1.5.3 Localisation des paquets d'ondelettes

#### La localisation spatiale

Les principes expliquant la localisation spatiale des éléments de la base de paquets d'ondelettes sont similaire à ceux expliquant la localisation des ondelettes. Un élément de la base de paquets d'ondelettes correspondant à un niveau de décomposition  $j$  est translaté tous les  $2^j$  pixels. Le calcul de ses coefficients s'est fait en sous-échantillonnant  $j$  fois par un facteur 2. La taille du support d'un élément de la base de paquets d'ondelettes du niveau  $j$  est comprise entre  $2^j \min(|h|, |g|)$  et  $2^j \max(|h|, |g|)$ .

On considère donc généralement que plus le niveau de décomposition du coefficient d'ondelette est élevé, plus sa localisation spatiale est mauvaise.

#### La localisation fréquentielle

Pour suivre ce qui se passe dans le domaine fréquentiel, lorsque l'on décompose un signal  $u \in \mathbb{R}^N$ , rappelons (1.12) et (1.13) :

$$\widehat{u^1}_k = \frac{1}{2} \left( \widehat{h}_k^* \widehat{u}_k + \widehat{h}_{k+\frac{N}{2}}^* \widehat{u}_{k+\frac{N}{2}} \right)$$

et

$$\widehat{u^2}_k = \frac{1}{2} \left( \widehat{g}_k^* \widehat{u}_k + \widehat{g}_{k+\frac{N}{2}}^* \widehat{u}_{k+\frac{N}{2}} \right).$$

Ces formules nous donnent la transformée de Fourier des signaux obtenus après la décomposition en fonction de la transformée de Fourier du signal  $u$ .

Rappelons aussi que dans le cadre des ondelettes,  $h$  est typiquement passe bas et  $g$  passe haut. Pour simplifier l'analyse, on suppose

$$\widehat{h}_k \sim 0, \text{ pour } k \in \left\{ -\frac{N}{4}, \dots, \frac{N}{4} \right\}$$

et

$$\widehat{g}_k \sim 0, \text{ pour } k \notin \left\{ -\frac{N}{4}, \dots, \frac{N}{4} \right\}.$$

<sup>2</sup>Un “quadtrees” est un arbre dans lequel chaque noeud moins de 4 noeuds.

<sup>3</sup>Un “octree” est un arbre dans lequel chaque noeud moins de 8 noeuds.

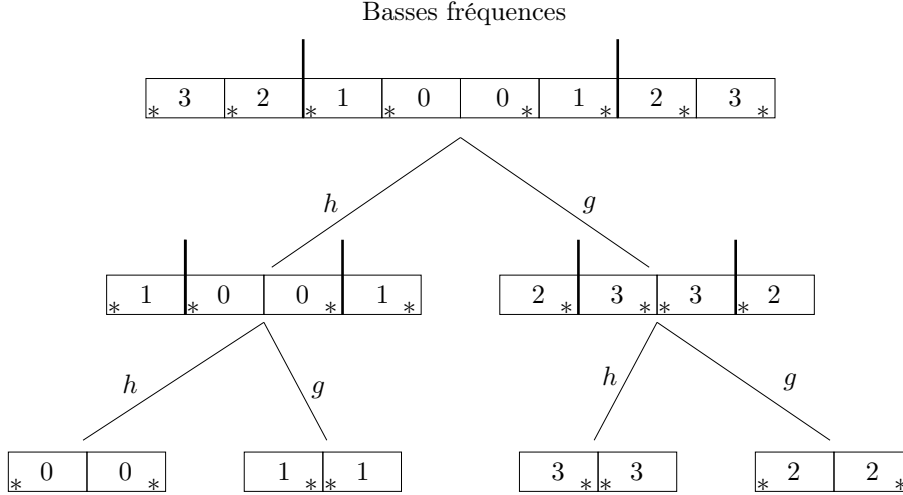


FIG. 1.20 – Suivit des différentes bandes de fréquences au cours des décompositions avec les filtres  $h$  et  $g$ .

Ainsi, lors d'une première décomposition on obtient le résultat schématisé sur la Figure 1.20 :

- La transformée de Fourier du résultat de la décomposition utilisant  $h$  (c.à.d.  $\widehat{u^1}$ ) correspond aux basses fréquences de  $u$  (les bandes 0 et 1).
- La transformée de Fourier du résultat de la décomposition utilisant  $g$  (c.à.d.  $\widehat{u^2}$ ) correspond aux hautes fréquences de  $u$  (les bandes 2 et 3), mais l'ordre de ces hautes fréquences est inversé, du fait de la formule de Poisson.

Si l'on décompose à nouveau  $u^2$  (ce que l'on ne faisait pas avec les ondelettes mais que l'on peut faire avec les paquets d'ondelettes) :

- Les basses fréquences de  $u^2$  (c.à.d. les bandes 3) correspondent à la transformée de Fourier de la décomposition avec  $h$ .
- Les hautes fréquences de  $u^2$  (c.à.d. les bandes 2) correspondent à la transformée de Fourier de la décomposition avec  $g$ .

On généralise l'observation qui est faite ci-dessus de la façon suivante : Un paquet d'ondelettes décomposé au niveau  $j$  va être localisé sur une bande de fréquence

$$\left\{ -(k+1)\frac{N}{2^{j+1}}, \dots, -k\frac{N}{2^{j+1}} \right\} \cup \left\{ -k\frac{N}{2^{j+1}}, \dots, (k+1)\frac{N}{2^{j+1}} \right\},$$

pour un certain  $k \in \{0, \dots, 2^j - 1\}$  bien choisi.

Pour établir la correspondance entre un des éléments de la base de paquets d'ondelettes du niveau  $j$  et le bon nombre entier  $k \in \{0, \dots, 2^j - 1\}$ , on numérote les  $2^j$  noeuds possibles au niveau  $j$  de gauche à droite. Le noeud numéroté par  $p$  correspond à la bande de fréquence repéré par  $k = G(p)$ , où la fonction  $G$  est construite par récurrence :

$$G(0) = 0, G(1) = 1$$

et pour  $j' = 1, 2, \dots, j-1$  et  $p \in \{2^{j'-1}, \dots, 2^{j'} - 1\}$

$$G(2p) = \begin{cases} 2G(p) & , \text{ si } G(p) \text{ est pair} \\ 2G(p) + 1 & , \text{ si } G(p) \text{ est impair} \end{cases}$$

et

$$G(2p+1) = \begin{cases} 2G(p) + 1 & , \text{ si } G(p) \text{ est pair} \\ 2G(p) & , \text{ si } G(p) \text{ est impair.} \end{cases}$$

Les premier éléments de la suite  $G$  sont donnés dans le Tableau 1.1

On peut donc atteindre n'importe quelle partition du domaine fréquentiel. Bien sûr, si l'on cherche à être trop précis dans le domaine fréquentiel, on aboutira à des niveaux de décomposition élevé et donc à une faible localisation en espace (ce qui n'est généralement pas souhaitable).

p	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G(p)	0	1	3	2	7	6	4	5	15	14	12	13	8	9	11	10

TAB. 1.1 – Correspondance entre le numéro du paquet d'ondelettes et le numéro de la bande de fréquence correspondante

Enfin cette analyse n'est pertinente que dans la mesure où les filtres  $h$  et  $g$  sont eux même bien localisés en fréquences (i.e. ils ont un nombre élevé de moments nuls). Cette analyse n'a, par exemple, qu'une portée très limitée avec l'ondelette de Haar qui n'est pas du tout localisée en fréquence.

### 1.5.4 Les dictionnaires de paquets d'ondelettes

L'un des intérêts des paquets d'ondelettes est qu'ils permettent de définir des dictionnaires de paquets d'ondelettes. Les dictionnaires ("frame" en anglais) sont des systèmes redondants.

Dans le cadre des paquets d'ondelette, un arbre définit une base. On peut aisément construire des dictionnaires en considérant plusieurs arbres. On peut, par exemple, considérer l'union des bases de paquets d'ondelettes complètement décomposés de niveau  $j$ , pour  $j \in \{1, \dots, J\}$ .

L'un des attraits particuliers des dictionnaires de paquets d'ondelettes est que les calculs des coefficients pour  $j$  petit sont des calculs intermédiaires pour le calcul des coefficients du niveau  $j + 1$ . Ils ne sont donc pas à refaire lorsque l'on calcule ces coefficients. Ainsi le coût de calcul des coefficients dans un dictionnaire de paquets d'ondelettes contenant tous les paquets d'ondelettes jusqu'au niveau  $J$  est le même que le coût de calcul dans une (unique!) base de paquets d'ondelette complètement décomposé jusqu'au niveau  $J$ .

Il en va évidemment de même pour la reconstruction.

### 1.5.5 Le passage aux dimensions 2 et plus

Pour définir les paquets d'ondelettes d'images ou, plus généralement, de données en dimension supérieure à 1. Comme pour les ondelettes, on utilise un simple produit tensoriel des éléments de la base. Algorithmiquement :

- pour chaque noeud de l'arbre : si le noeud a des fils on le décompose.
- Décomposer un noeud se fait
  - en appliquant la décomposition successivement sur toutes les lignes de l'image (on obtient deux images) ;
  - en appliquant la décomposition successivement sur toutes les colonnes des deux images.

Concernant les localisations spatiales et fréquentielles, on a des propriétés similaires à la dimension 1. Une souplesse que l'on n'a pas est que les éléments d'une base de paquets d'ondelettes en dimension 2 ont un niveau unique qui s'applique à ces lignes et ses colonnes. Ainsi, le support d'un élément d'une base de paquet d'ondelette est toujours plus ou moins carré. Il en va de même pour la localisation en fréquence.

On peut cependant, comme en dimension 1, atteindre toutes les partitions du domaine fréquentiel. Ceci se fait éventuellement au détriment de la localisation en espace des éléments de la base de paquets d'ondelettes.

## 1.6 D'autres bases

Nous avons vu jusqu'ici les bases de Fourier (voir la Section 1.2), de cosinus et cosinus locaux (voir la Section 1.3), d'ondelettes (voir la Section 1.4) et plusieurs de paquets d'ondelettes (voir la Section 1.5).

Il existe, en fait, d'autres bases (et dictionnaires) dont l'étude n'est pas faite dans ce cours. Les lecteurs intéressés pourront trouver des détails sur internet à partir des mots clés : ridgelet, curvelet, contourlet, bandlet, brushlet.

Chacune de ces bases est adaptée à un contenu particulier, éventuellement à une échelle particulière. La plupart d'entre elles cherchent à mieux représenter (i.e. à représenter avec le moins de coordonnées possible) les bords des images.



FIG. 1.21 – Décomposition de l'image "barbara" dans une base de paquets d'ondelettes complètement décomposée jusqu'au niveau 1. Les paquets sont présentés dans l'ordre fréquentiel.

## 1.7 La représentation dans un dictionnaire

### 1.7.1 Introduction

Dans les chapitres précédents, nous avons envisagé d'exprimer notre image dans une base. Ce faisant, nous avons vu plusieurs bases. Certaines bases étaient bien adaptées pour représenter des textures périodiques sur toute l'image (Fourier), des objets de petite taille (les ondelettes), des textures périodiques (certains paquets d'ondelettes). Nous avons aussi mentionné des bases (ou dictionnaires) permettant de représenter des bords étendus.

En fait ces différentes structures (bords, textures...) sont souvent présentes sur une même image. Par exemple, sur l'image *Barbara* (voir Figure 1.2) on a simultanément des bords (par exemple à la frontière entre le pied de la table et le sol) des textures périodiques (par exemple sur le pantalon), des petits objets (par exemple le point blanc au bas du pied de la table ou les marques des livres). On voudrait donc idéalement représenter chaque zone de l'image avec les éléments de la base la mieux adaptée à cette zone.

Par exemple, si on a deux bases  $\mathcal{B}$  et  $\mathcal{B}'$  de  $\mathbb{R}^{N^2}$ , on cherche à approximer une image  $u \in \mathbb{R}^{N^2}$  en utilisant le moins de coordonnées possible dans  $\mathcal{B} \cup \mathcal{B}'$ . On veut donc idéalement résoudre le problème variationnel suivant :

$$\begin{cases} \text{Min } l^0((\lambda_\varphi)_{\varphi \in (\mathcal{B} \cup \mathcal{B}')} ) \\ \| \sum_{\varphi \in (\mathcal{B} \cup \mathcal{B}')} \lambda_\varphi \varphi - u \| \leq \tau, \end{cases}$$



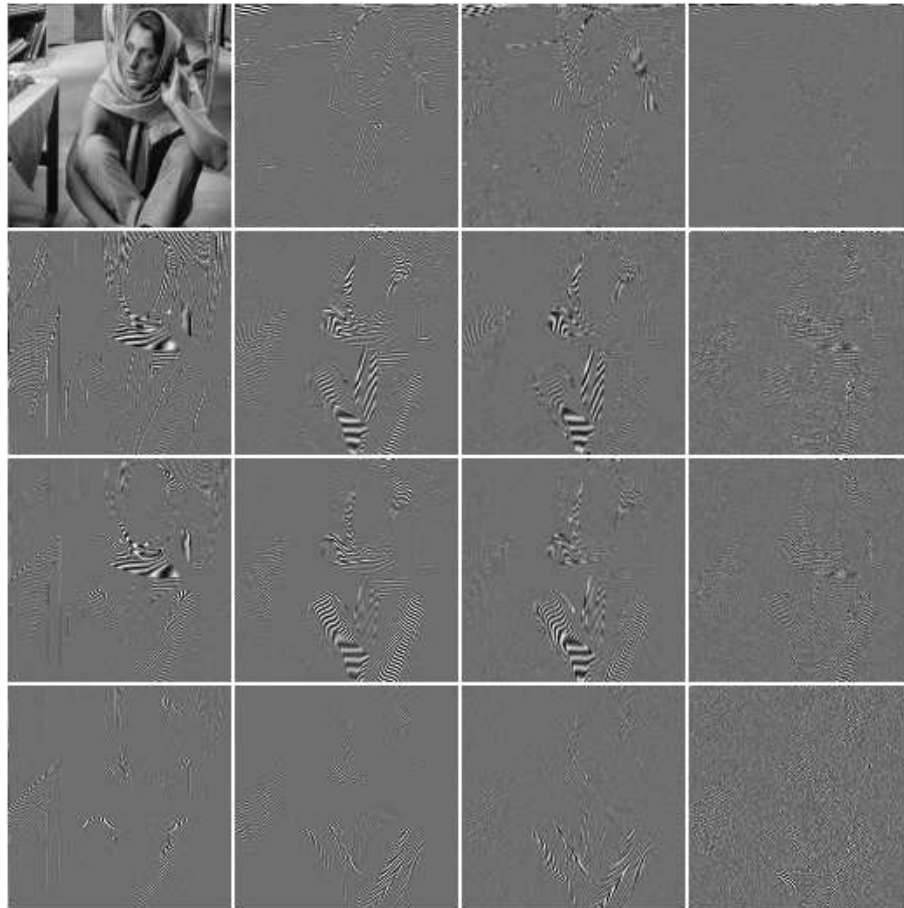


FIG. 1.22 – Décomposition de l'image "barbara" dans une base de paquets d'ondelettes complètement décomposée jusqu'au niveau 2. Les paquets sont présentés dans l'ordre fréquentiel.

où  $\tau \geq 0$ ,

$$\|v\| = \sqrt{\sum_{i,j=0}^{N-1} |v_{i,j}|^2}, \quad \forall v \in \mathbb{R}^{N^2}$$

et

$$l^0((\lambda_\varphi)_{\varphi \in (\mathcal{B} \cup \mathcal{B}')} ) = \#\{\varphi \in (\mathcal{B} \cup \mathcal{B}'), \lambda_\varphi \neq 0\},$$

représente le nombre de coordonnées  $\lambda_\varphi$  non nulles. (On rappelle que  $\#$  désigne le cardinal d'un ensemble.)

Ci dessus, on n'a considéré que le cas où le dictionnaire est constitué de deux bases. Plus généralement, si on considère un dictionnaire  $\mathcal{D}$  (i.e. un système linéaire fini générant  $\mathbb{R}^{N^2}$ ), on cherche un système de coordonnées résolvant le problème suivant :

$$\begin{cases} \text{Min } l^0((\lambda_\varphi)_{\varphi \in \mathcal{D}}) \\ \|\sum_{\varphi \in \mathcal{D}} \lambda_\varphi \varphi - u\| \leq \tau, \end{cases} \quad (1.27)$$

pour  $\tau \geq 0$ .

Il n'est pas très difficile de voir que le problème (1.27) à une solution et qu'il atteint cette solution. En effet, comme  $\mathcal{D}$  génère  $\mathbb{R}^{N^2}$ , il existe (au moins) un jeu de coordonnées permettant de construire  $u$ . Ainsi

$$\left\{ (\lambda_\varphi)_{\varphi \in \mathcal{D}} \in \mathbb{R}^{\mathcal{D}}, \left\| \sum_{\varphi \in \mathcal{D}} \lambda_\varphi \varphi - u \right\| \leq \tau \right\} \neq \emptyset.$$

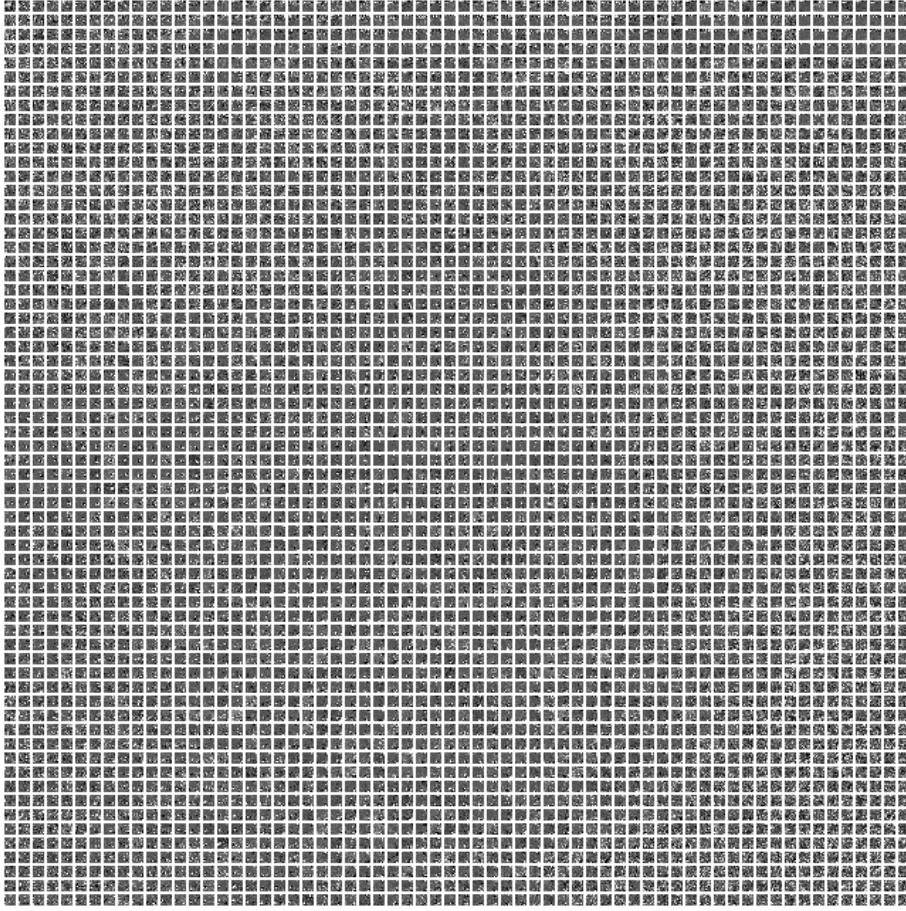


FIG. 1.23 – Décomposition de l'image "barbara" dans une base de paquets d'ondelettes complètement décomposée jusqu'au niveau 6. Les paquets sont présentés dans l'ordre fréquentiel.

Par ailleurs, comme la fonction  $l^0$  est à valeur dans  $\mathbb{N}$  (un ensemble discret et minoré), le problème (1.27) atteint sa valeur minimum pour un jeu de coordonnées  $(\lambda_\varphi^*)_{\varphi \in \mathcal{D}}$ .

Enfin, il n'est non plus très difficile de voir que le problème (1.27) n'a généralement pas une unique solution. Le principal défaut de (1.27) est qu'il n'est ni convexe ni différentiable. Ceci rend sa résolution approchée par des algorithmes itératifs très difficile. Il est par ailleurs NP-difficile. Ceci signifie que (en général) le calcul d'une solution exacte de (1.27) est (et sera toujours) trop long pour être réalisé.

Il existe évidemment un certain nombre de situations dans lesquels on sait calculer une solution de (1.27). Par exemple, si  $\mathcal{D}$  est une base orthonormale, il suffit de seuiller les coordonnées de l'image à la bonne valeur.

Ceci conduit la communauté travaillant dans ce domaine à proposer différents algorithmes permettant d'approximer une solution de (1.27) en un temps acceptable.

Les principaux mots clefs de ce domaine sont : représentation parcimonieuse (sparse representation, sparsity), dictionnaire (dictionary, frame), échantillonnage compressé (compressed/compressive sensing).

### 1.7.2 Préliminaires mathématiques et notations

Jusque là, nous avons utilisé des notations différentes pour les différentes bases que nous avons considérées. Nous allons dans la suite de ce chapitre utiliser des notations matricielles. Elles sont, en effet, bien pratiques et majoritairement utilisées dans le domaine des représentations parcimonieuses.

Pour cela on considérera que l'image  $u \in \mathbb{R}^{N^2}$  est un unique vecteur colonne. Ce vecteur est obtenu en faisant la concaténation de toutes les colonnes de l'image. On considère également que les éléments du

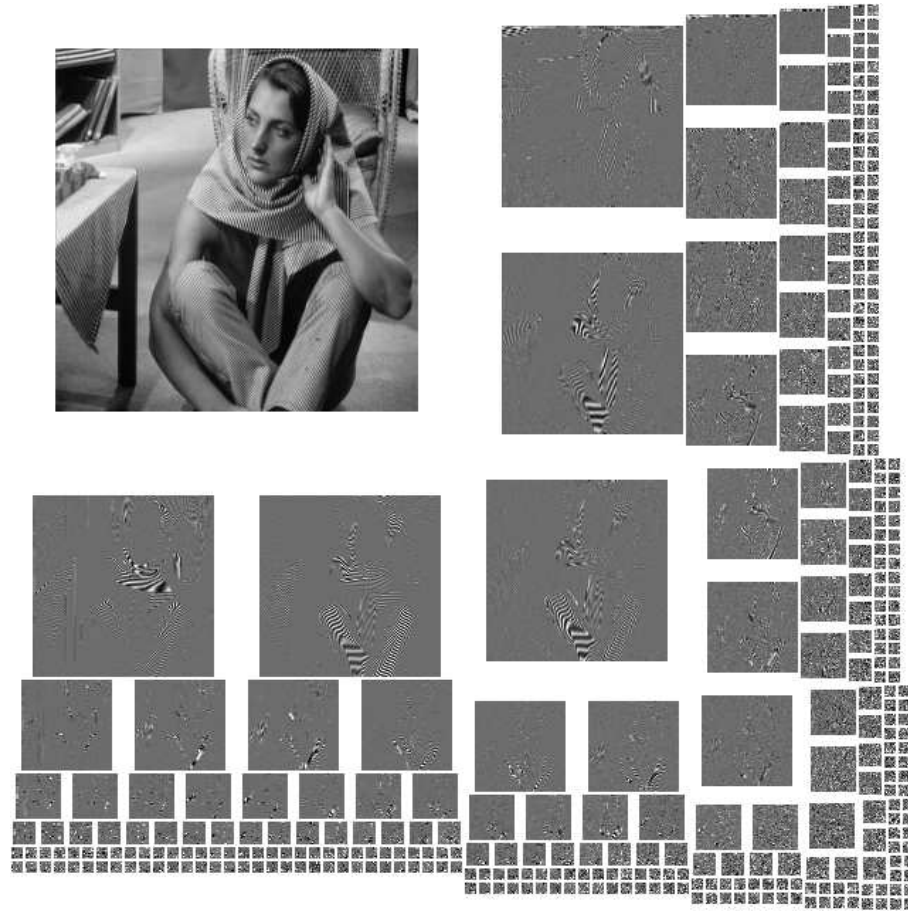


FIG. 1.24 – Décomposition de l'image "barbara" dans une base de paquets d'ondelettes miroir d'ondelette de niveau 6. Les paquets sont présentés dans l'ordre fréquentiel.

dictionnaire  $\mathcal{D}$  sont des vecteurs colonnes obtenus en faisant la concaténation des colonnes des images de la base. On note  $A$  la matrice obtenue en mettant côte à côte tous les éléments du dictionnaire. La matrice  $A$  est donc une matrice de taille  $N^2 \times \#\mathcal{D}$  (c'est une matrice rectangulaire avec plus de colonnes que de lignes). Si on met les coordonnées  $(\lambda_\varphi)_{\varphi \in \mathcal{D}} \in \mathbb{R}^{\#\mathcal{D}}$  dans un vecteur colonne  $x$ , on a alors

$$\sum_{\varphi \in \mathcal{D}} \lambda_\varphi \varphi = Ax.$$

Avec ces notations, le problème idéal (1.27) s'écrit

$$\begin{cases} \text{Min } l^0(x) \\ \|Ax - u\| \leq \tau, \end{cases} \quad (1.28)$$

pour  $\tau \geq 0$ .

Un dessin de la situation dans laquelle on se trouve est présentée sur la Figure 1.26.

Les différents algorithmes que nous allons voir<sup>4</sup> utilisent deux opérateurs mettant en jeu le dictionnaire :

- la multiplication  $Ax$ , pour un vecteur  $x \in \mathbb{R}^{\#\mathcal{D}}$
- la multiplication  $A^t v$ , pour un vecteur  $v \in \mathbb{R}^{N^2}$  et pour  $A^t$  la transposée de  $A$ .

Le temps de calcul nécessaire à un algorithme construisant une approximation parcimonieuse est souvent majoritairement passé à faire ces multiplications. Nous détaillons ci-dessous comment faire ces calculs pour les bases et dictionnaires vus jusqu'ici.

---

<sup>4</sup>comme tous les algorithmes dans ce domaine

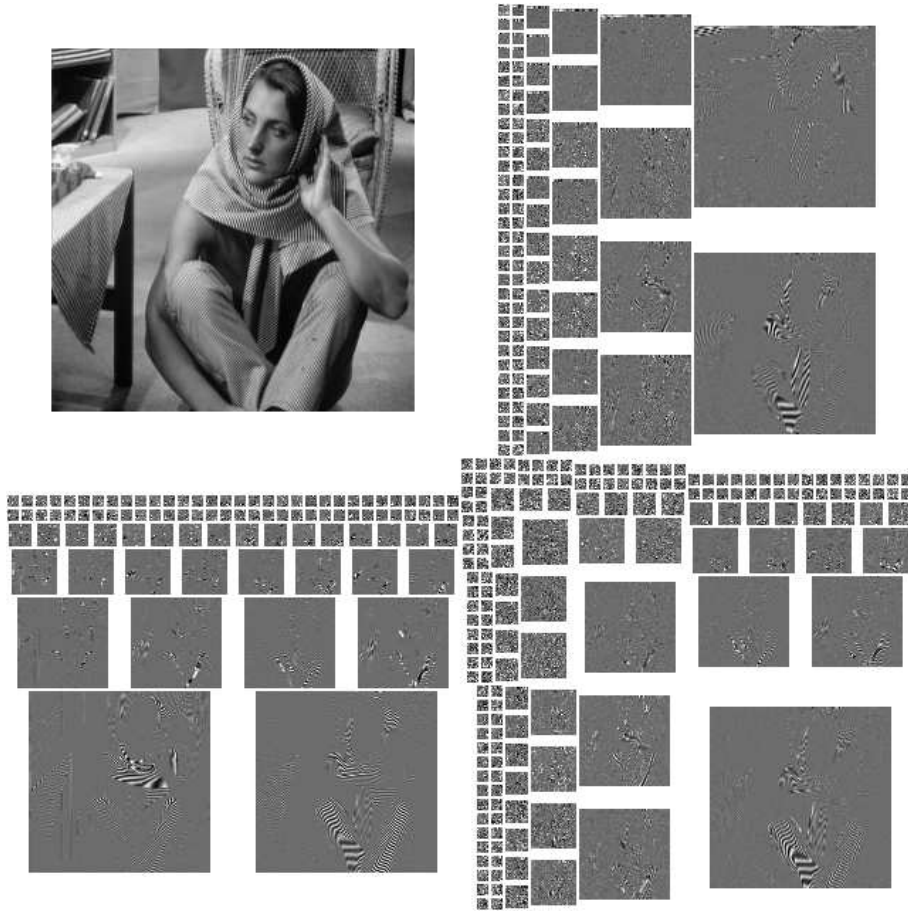


FIG. 1.25 – Décomposition de l'image "barbara" dans une base de paquets d'ondelettes miroir d'ondelette de niveau 6. Les paquets sont présentés dans leur ordre naturel.

#### Le cas des bases :

Dans les chapitres précédents, nous avons vu plusieurs bases. Si l'on utilise les notation matricielles, on sait calculer, pour chacune de ses bases  $Ax$  et son inverse  $A^{-1}v$ . Nous détaillons ci-dessous comment calculer les opérateurs  $Ax$  et  $A^t v$  pour ces différentes bases.

- Pour les **bases orthonormales** : C'est la cas des bases d'ondelettes et de paquets d'ondelettes orthogonales. Nous pouvons appliquer à  $v$  l'algorithme calculant la décomposition pour calculer  $A^t v$ . Nous pouvons calculer  $Ax$ , en utilisant l'algorithme de reconstruction.
- Pour les **bases orthogonales dont tous les éléments ont la même norme  $\delta$**  : C'est la cas de la base de Fourier ( $\delta = \frac{1}{N}$ ), de la base de cosinus, de la base de cosinus locaux. Il n'est pas difficile de voir que l'on peut calculer  $Ax$  avec l'algorithme de reconstruction habituel et que l'on peut calculer  $A^t v$  en multipliant le résultat de l'algorithme de décomposition habituel par  $\delta^2$ .
- Pour les **bases biorthogonales** : Dans le cas particulier des bases d'ondelettes et de paquets d'ondelettes biorthogonales, on calcule la décomposition et son inverse à l'aide de filtres  $(h, g, \tilde{h}, \tilde{g})$  (voir les Section 1.4.2 et 1.4.3). Pour calculer  $Ax$ , on recompose les coefficients avec les filtres  $\tilde{h}$  et  $\tilde{g}$  (respectivement  $h$  et  $g$ ) et pour calculer  $A^t v$  on décompose avec les filtres  $\tilde{h}$  et  $\tilde{g}$  (respectivement  $h$  et  $g$ ). Ainsi, contrairement à ce que l'on fait pour calculer une décomposition en ondelette biorthogonale et son inverse, on n'utilise que deux filtres pour les calculs de  $Ax$  et  $A^t v$  : soit  $\tilde{h}$  et  $\tilde{g}$  ; soit  $h$  et  $g$ .

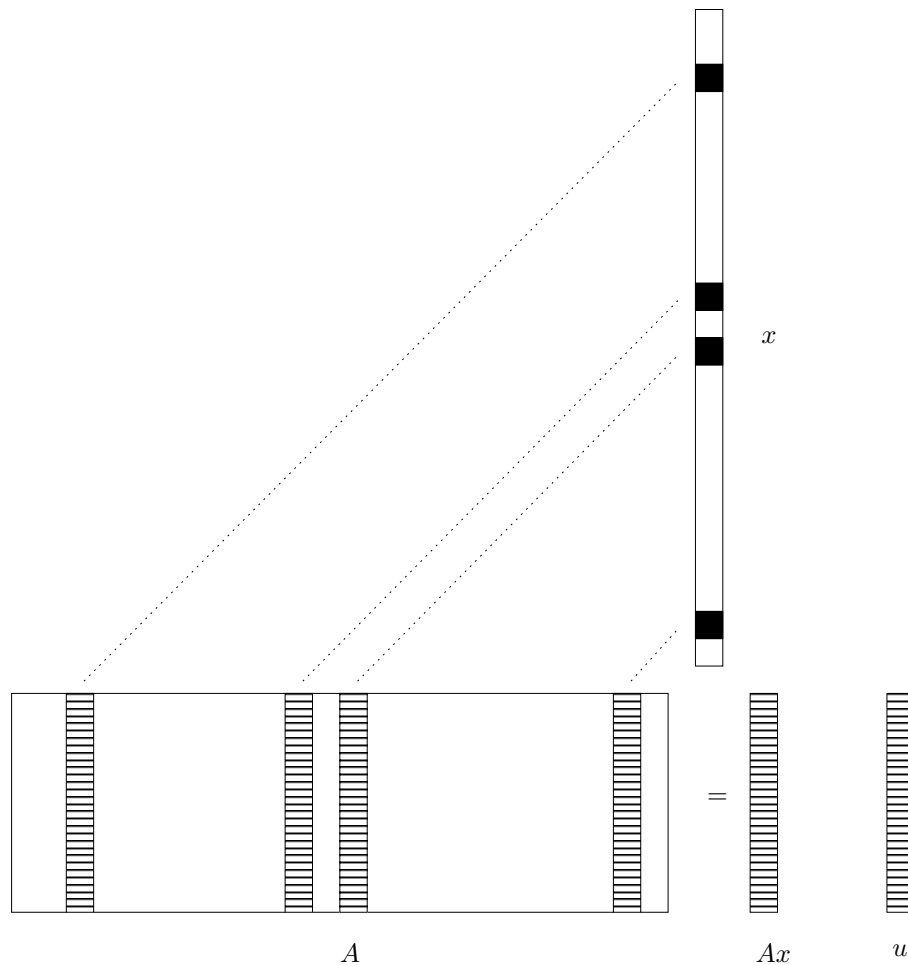


FIG. 1.26 – Représentation des matrices et vecteurs manipulés dans le problème (1.28). Les cases noires de  $x$  représentent les coordonnées non-nulles.

### L'union de bases

Ici, on considère un dictionnaire constitué de plusieurs bases  $\mathcal{D} = \mathcal{B}_1 \cup \dots \mathcal{B}_K$ . On considère, pour  $i = 1, \dots, K$  et pour chaque base  $\mathcal{B}_i$ , la matrice  $A_i$  correspondante. La matrice  $A$  correspondant à  $\mathcal{D}$  est alors la concaténation (en ligne) des matrices  $A_i$ , pour  $i = 1, \dots, K$ . Le calcul de  $Ax$ , où  $x$  est la concaténation des vecteurs  $x_1, \dots, x_K$  se fait donc avec la formule suivante :

$$Ax = A_1x_1 + A_2x_2 + \dots + A_Kx_K.$$

Le calcul de  $A^tv$ , pour  $v \in \mathbb{R}^{N^2}$  se fait en concaténant tous les vecteurs  $A_i^tv$ , pour  $i = 1, \dots, K$ .

### 1.7.3 La minimisation $l^1$

Comme nous l'avons vu, le problème (1.28) est trop difficile pour être résolu. Une alternative possédant quelques similarités avec le problème (1.28) consiste à remplacer le terme  $l^0$  de (1.28) par un terme  $l^1$ . On obtient alors le problème

$$\begin{cases} \text{Min } l^1(x) \\ \|Ax - u\| \leq \tau, \end{cases} \quad (1.29)$$

pour  $\tau \geq 0$  et pour la norme  $l^1$  définie par

$$l^1(x) = \sum_{i=1}^{\#\mathcal{D}} |x_i|.$$

L'avantage de ce problème est que l'on peut calculer ses solutions de manière exacte en exploitant la structure de ce problème. Par contre, ce calcul n'est faisable en un temps acceptable que dans les cas où sa solution reste très parcimonieuse (i.e. à très peu de coordonnées non nulles). Lorsque ce n'est pas le cas, on peut utiliser un algorithme itératif pour approcher une solution de (1.29). Ce problème est, en effet, convexe (mais non différentiable).

Un nombre important d'algorithmes ont été proposées pour résoudre (1.29). Ils reposent souvent sur un argument de dualité lagrangienne qui permet de récrire (1.29) sous la forme

$$l^1(x) + \frac{\beta}{2} \|Ax - u\|^2, \quad (1.30)$$

pour une valeur de  $\beta$  bien choisie.

Les problèmes (1.30) et (1.29) sont généralement appelés “minimisation  $l^1$ ” ou “Basis Pursuit Denoising”. Ils sont souvent considérés comme les meilleures alternatives convexe du problème (1.27). Ceci est notamment lié au fait qu'ils ont le meilleur comportement observé vis à vis dans le cadre d'une des applications de (1.27) et (1.29) : le “compressed sensing”.

#### 1.7.4 Les algorithmes de type “Orthogonal Matching Pursuit”

Une alternative purement algorithmique au problème (1.28) est l'orthogonal matching pursuit (OMP). En fait, il y a toute une famille d'algorithmes se ressemblant : Matching Pursuit, stagewise OMP, CO-SAMP...

Le point commun entre ces algorithmes est de sélectionner à chaque itération une ou plusieurs coordonnées parmi les plus grandes coordonnées (en valeur absolue) de  $A^t r$ , où  $r = Ax - u$  et  $x$  est la solution courante. Intuitivement,  $r$  contient la partie de  $u$  qui n'est pas encore présente dans  $Ax$ . Chaque coordonnée de  $A^t r$  représente donc la corrélation entre un élément du dictionnaire et ce qu'il reste à exprimer. Cela revient à choisir les vecteurs “pointant” le plus dans la direction de  $r$ .

Une fois qu'un ensemble d'éléments du dictionnaire est sélectionné, on construit la solution en projetant  $u$  orthogonalement sur le sous-espace vectoriel engendré par ces éléments. Plus précisément, on note

- $\mathcal{S} \subset \mathcal{D}$ ,
- $\text{Vect}(\mathcal{S})$  le sous-espace vectoriel de  $\mathbb{R}^{N^2}$  engendré par les éléments de  $\mathcal{S}$
- $P_{\text{Vect}(\mathcal{S})}(v)$  la projection orthogonale d'un élément  $v \in \mathbb{R}^{N^2}$  sur  $\text{Vect}(\mathcal{S})$ .

Dans les algorithmes de la famille Orthogonal Matching Pursuit, on construit  $\mathcal{S}$  comme indiqué ci-dessus et on calcule les coordonnées  $x$  de  $P_{\text{Vect}(\mathcal{S})}(u)$ . On impose évidemment aux coordonnées  $x$  d'être nulles pour les vecteurs de  $\mathcal{D}$  qui ne sont pas dans  $\mathcal{S}$ . Dans certains cas (notamment lorsque la solution n'est pas très parcimonieuse), le calcul des coordonnées  $x$  de  $P_{\text{Vect}(\mathcal{S})}(u)$  peut être un peu long. Bien souvent, une algorithmique particulière est nécessaire pour accélérer ce calcul.

Nous présentons l'Orthogonal Matching Pursuit dans la Table 1.2.

- \* Entrées : L'image  $u$ , un dictionnaire  $\mathcal{D}$ , un paramètre de critère d'arrêt.
- \* Sortie : Les coordonnées  $x$
- \* L'algorithme :
  - On initialise  $\mathcal{S} = \emptyset$ ,  $r = u$
  - Tant que le critère d'arrêt n'est pas satisfait :
    - On calcule  $A^t r$
    - On cherche  $\varphi$  correspondant au maximum de  $|A^t r|$
    - On met à jour  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\varphi\}$
    - On calcule les coordonnées  $x$  de la projection  $P_{\text{Vect}(\mathcal{S})}(u)$  et le nouveau résidu  $r = P_{\text{Vect}(\mathcal{S})}(u)$

TAB. 1.2 – L'algorithme Orthogonal Matching Pursuit.





## Chapitre 2

# La restauration d'images

On appelle *restauration* d'images l'action qui consiste à inverser une ou plusieurs des dégradations que l'on a décrites au chapitre 1.1. En mathématique, on parle de résolution de *problèmes inverses*. Nous aborderons les problèmes du *débruitage* (qui consiste à enlever le bruit), de la *déconvolution* (on inverse l'opérateur de convolution) et de *l'interpolation* (on inverse l'opérateur d'échantillonnage).

### 2.1 Le débruitage par convolution

Comme nous l'avons vu au chapitre 1.1.5, une image contient toujours un bruit. Celui-ci peut être plus ou moins important. Dans certains cas, on peut vouloir l'enlever ou pour le moins l'atténuer. C'est une opération que l'on appelle débruitage. La problématique du débruitage est d'arriver à séparer une image en deux parties : une comprenant surtout du bruit, l'autre devant ressembler à une image naturelle et contenir toutes les informations initialement contenues dans l'image. Il convient donc d'être modéré car enlever plus de bruit conduit toujours à perdre des informations initialement présentes dans l'image.

Le bruit Gaussien étant irrégulier, atténuer le bruit implique une régularisation de l'image bruitée. Pour ce faire, il semble tout naturel d'effectuer une convolution de l'image avec un noyau de convolution qui atténue les hautes fréquences (on appelle un noyau de ce type un *filtre passe bas*). Nous avons, en effet, déjà dit qu'une image (ou fonction) est d'autant plus régulière que le module de sa transformée de Fourier décroît rapidement.

Cependant, lorsque l'on fait cette convolution, on ne fait qu'atténuer le bruit présent dans l'image. L'image contient toujours du bruit. Il est toutefois possible d'évaluer le bruit résiduel. C'est l'objet de la proposition suivante.

**Proposition 6** Soit  $(b_{m,n})_{1 \leq m,n \leq N}$  un bruit Gaussien de variance  $\sigma^2$  (rappel : c'est une suite finie de réalisations indépendantes de la loi de probabilité Gaussienne, d'espérance nulle et de variance  $\sigma^2$ ) et  $(h_{m,n})_{1 \leq m,n \leq N}$  une suite finie. Alors  $((h * b)_{m,n})_{1 \leq m,n \leq N}$  est une variable aléatoire d'espérance nulle et de variance  $\|h\|^2 \sigma^2$ , avec  $\|h\|^2 = \sum_{m,n=1}^N (h_{m,n})^2$ .

*Preuve.* Cette démonstration est analogue à celle montrant que la transformée de Fourier d'un bruit est un bruit (voir chapitre 1.2.5).

La seule différence réside dans le fait que cette fois, il est plus difficile de préciser la loi de la convolution du bruit. Cependant, on peut calculer son espérance.

Soit  $(m, n) \in \{1, \dots, N\}^2$ , on a

$$\begin{aligned}
 \mathbb{E}((h * b)_{m,n}) &= \mathbb{E}\left(\sum_{m',n'=1}^N h_{m-m',n-n'} b_{m',n'}\right) \\
 &= \sum_{m',n'=1}^N h_{m-m',n-n'} \mathbb{E}(b_{m',n'}) \\
 &= \sum_{m',n'=1}^N h_{m-m',n-n'} 0 \\
 &= 0.
 \end{aligned}$$

En ce qui concerne la variance, on a , pour tout  $(m, n) \in \{1, \dots, N\}^2$ ,

$$\begin{aligned}
 \mathbb{E}\left(\left((h * b)_{m,n}\right)^2\right) &= \mathbb{E}\left(\left(\sum_{m',n'=1}^N h_{m-m',n-n'} b_{m',n'}\right)^2\right) \\
 &= \mathbb{E}\left(\left(\sum_{m',n'=1}^N h_{m-m',n-n'} b_{m',n'}\right)\left(\sum_{m'',n''=1}^N h_{m-m'',n-n''} b_{m'',n''}\right)\right) \\
 &= \sum_{m',n'=1}^N \sum_{m'',n''=1}^N h_{m-m',n-n'} h_{m-m'',n-n''} \mathbb{E}(b_{m',n'} b_{m'',n''}).
 \end{aligned}$$

Mais, comme les  $b_{m,n}$  sont indépendants les uns des autres et sont d'espérance nulle, on a, pour  $(m', n') \neq (m'', n'')$ ,

$$\mathbb{E}(b_{m',n'} b_{m'',n''}) = \mathbb{E}(b_{m',n'}) \mathbb{E}(b_{m'',n''}) = 0.$$

De plus, comme les  $b_{m,n}$  sont d'espérance nulle et de variance  $\sigma^2$ , on a , pour  $(m', n') = (m'', n'')$ ,

$$\mathbb{E}(b_{m',n'} b_{m'',n''}) = \sigma^2.$$

On a donc finalement

$$\begin{aligned}
 \mathbb{E}\left(\left((h * b)_{m,n}\right)^2\right) &= \sum_{m',n'=1}^N h_{m-m',n-n'} h_{m-m',n-n'} \sigma^2 \\
 &= \sigma^2 \sum_{m'=m-N}^{m-1} \sum_{n'=n-N}^{n-1} (h_{m',n'})^2.
 \end{aligned}$$

Encore une fois, on fait la somme des termes d'une suite finie périodique sur une période. On peut donc réarranger la somme pour obtenir

$$\begin{aligned}
 \mathbb{E}\left(\left((h * b)_{m,n}\right)^2\right) &= \sigma^2 \sum_{m'=1}^N \sum_{n'=1}^N (h_{m',n'})^2 \\
 &= \sigma^2 \|h\|^2.
 \end{aligned}$$

□

Cette proposition montre que l'on obtient un bruit d'autant plus faible, après débruitage, que  $\|h\|^2$  est petite. Le filtre  $h \equiv 0$  est donc idéal pour enlever le bruit. Par contre, le résultat du débruitage sera une image constante, elle aussi, égale à zéro. Ceci ne présente donc pas beaucoup d'intérêt. Il convient donc

d'adopter un ou plusieurs critères pour nous assurer que la convolution n'engendre pas trop de pertes d'informations.

Par exemple, on peut imposer que la somme des  $h_{m,n}$  soit égale à 1. Comme ceci la moyenne de l'image débruitée est égale à la moyenne de l'image de départ. On a aussi constaté que plus  $h$  a un support étendu, plus le résultat de la convolution est flou. On peut donc imposer une contrainte sur la taille du support de  $h$ . Par ailleurs, il n'y a pas de raison pour prendre un noyau de convolution dont le support soit *anisotropique* (dépendant de la direction). En effet, si le support de  $h$  est très allongé dans une direction, les bords orthogonaux à cette direction seront très flous. Ces contraintes suffisent à déterminer un noyau de convolution. Elles contraignent, en effet, le noyau de convolution à avoir un support inclus dans un disque

$$\mathcal{D}_r = \{(m, n) \in \{1, \dots, N\}^2, m^2 + n^2 \leq r^2\}$$

où  $r > 0$  est un paramètre fixant le "niveau de flou" que l'on est prêt à accepter après le débruitage. La détermination complète de  $h$  découle de la proposition suivante.

**Proposition 7** *Soit  $r > 0$  et*

$$\mathcal{H} = \{h \in \mathbb{R}^{N^2}, \text{ tels que, } \forall (m, n) \notin \mathcal{D}_r, h_{m,n} = 0 \text{ et } \sum_{m,n=1}^N h_{m,n} = 1\}.$$

*Alors le noyau  $h^{min} \in \mathcal{H}$ , défini, pour  $(m, n) \in \{1, \dots, N\}^2$  par*

$$h_{m,n}^{min} = \frac{1}{\#\mathcal{D}_r} \mathbf{1}_{|\mathcal{D}_r}(m, n),$$

*où  $\#\mathcal{D}_r$  désigne le nombre de points de  $\mathcal{D}_r$ , appartient à  $\mathcal{H}$  et vérifie*

$$\|h^{min}\|^2 = \frac{1}{\#\mathcal{D}_r} \leq \|h\|^2,$$

*pour tout  $h \in \mathcal{H}$ .*

*Preuve.* Soit  $h \in \mathcal{H}$ , on note, pour tout  $(m, n) \in \mathcal{D}_r$ ,

$$h_{m,n} = \frac{1}{\#\mathcal{D}_r} + h'_{m,n}.$$

La contrainte sur la somme des  $h_{m,n}$  se traduit par

$$\begin{aligned} 1 &= \sum_{(m,n) \in \mathcal{D}_r} h_{m,n} \\ &= \sum_{(m,n) \in \mathcal{D}_r} \frac{1}{\#\mathcal{D}_r} + \sum_{(m,n) \in \mathcal{D}_r} h'_{m,n} \\ &= 1 + \sum_{(m,n) \in \mathcal{D}_r} h'_{m,n}. \end{aligned}$$

On a donc

$$\sum_{(m,n) \in \mathcal{D}_r} h'_{m,n} = 0. \quad (2.1)$$

Si l'on exprime maintenant  $\|h\|^2$ , on obtient

$$\begin{aligned}
\|h\|^2 &= \sum_{(m,n) \in \mathcal{D}_r} (h_{m,n})^2 \\
&= \sum_{(m,n) \in \mathcal{D}_r} \left( \frac{1}{\#\mathcal{D}_r} + h'_{m,n} \right)^2 \\
&= \sum_{(m,n) \in \mathcal{D}_r} \left( \left( \frac{1}{\#\mathcal{D}_r} \right)^2 + \frac{2}{\#\mathcal{D}_r} h'_{m,n} + (h'_{m,n})^2 \right) \\
&= \frac{1}{\#\mathcal{D}_r} + \frac{2}{\#\mathcal{D}_r} \sum_{(m,n) \in \mathcal{D}_r} h'_{m,n} + \sum_{(m,n) \in \mathcal{D}_r} (h'_{m,n})^2.
\end{aligned}$$

En appliquant (2.1), on obtient

$$\begin{aligned}
\|h\|^2 &= \frac{1}{\#\mathcal{D}_r} + \sum_{(m,n) \in \mathcal{D}_r} (h'_{m,n})^2 \\
&\geq \frac{1}{\#\mathcal{D}_r},
\end{aligned} \tag{2.2}$$

En appliquant (2.2) à  $h^{min}$  on voit que  $\|h^{min}\|^2 = \frac{1}{\#\mathcal{D}_r}$  et qu'elle minimise bien  $\|h\|^2$ , sous la contrainte  $h \in \mathcal{H}$ .  $\square$

Ceci nous assure donc, que dans le cadre des contraintes que nous avons fixées avant cette proposition, le noyau de convolution constant sur  $\mathcal{D}_r$ , valant  $\frac{1}{\#\mathcal{D}_r}$ , est le meilleur choix possible.

## 2.2 La déconvolution

### 2.2.1 Le filtre inverse

Le problème de la déconvolution est en fait assez proche de celui du débruitage. En effet, il consiste à partir de la suite finie

$$u_{m,n} = (h * v)_{m,n} + b_{m,n}$$

et à chercher à retrouver les valeurs de  $v_{m,n}$ . Si l'on exprime ceci dans le domaine de Fourier, on cherche à retrouver  $\hat{v}_{k,l}$ , à partir de

$$\begin{aligned}
\hat{u}_{k,l} &= \widehat{(h * v)}_{k,l} + \hat{b}_{k,l} \\
&= \hat{h}_{k,l} \hat{v}_{k,l} + \hat{b}_{k,l}.
\end{aligned}$$

On a donc, (tout du moins, si la transformée de Fourier de  $h$  ne s'annule pas)

$$\frac{\hat{u}_{k,l}}{\hat{h}_{k,l}} = \hat{v}_{k,l} + \frac{\hat{b}_{k,l}}{\hat{h}_{k,l}}. \tag{2.3}$$

Ceci revient donc à faire un débruitage d'un bruit déconvolué (il n'est plus blanc).

Cette formule peut aussi être utile pour faire la déconvolution. Il semble suffisant de

- calculer les coefficients de Fourier de l'image et du noyau de convolution
- faire le rapport
- reconstruire l'image, à partir de ses coefficients de Fourier.

Cependant, la déconvolution sera d'autant plus mauvaise que la transformée de Fourier de  $h$  sera petite (la formule (2.3) n'est même pas valide si elle est nulle). En effet, aux fréquences où  $\hat{h}$  est petit,  $\frac{1}{\hat{h}}$  est fort. Ces coefficients de Fourier contiennent donc essentiellement du bruit.

La façon la plus naïve de résoudre ce problème est de “tronquer”  $\frac{1}{h}$  afin de ne pas trop augmenter le bruit. On introduit donc un paramètre  $\tau > 0$  et on adopte l’algorithme suivant (pour simplifier les notations, nous ne donnons l’algorithme que dans le cas où  $\hat{h}$  est réel, ceci correspond à un noyau symétrique) :

- calculer les coefficients  $\hat{u}_{k,l}$  et  $\hat{h}_{k,l}$  de l’image et du noyau de convolution
- on pose

$$\hat{v}_{k,l} = \begin{cases} \frac{\hat{u}_{k,l}}{\hat{h}_{k,l}} & , \text{ si } |\hat{h}_{k,l}| \geq \tau \\ \frac{\hat{u}_{k,l}}{\tau} & , \text{ si } 0 < \hat{h}_{k,l} < \tau \\ 0 & , \text{ si } \hat{h}_{k,l} = 0 \\ -\frac{\hat{u}_{k,l}}{\tau} & , \text{ si } -\tau < \hat{h}_{k,l} < 0 \end{cases}$$

- reconstruire l’image  $\tilde{v}$ , à partir de ses coefficients de Fourier  $\hat{v}_{k,l}$ .

Cette approche, bien que très naïve, permet de bien comprendre le problème de la déconvolution. Elle peut, de plus, donner de bons résultats lorsque la dégradation n’est pas trop importante (le bruit est faible, et la transformée de Fourier du noyau de convolution ne prend pas de valeurs trop faibles). Elle est théoriquement surpassée par le filtre de Wiener, que nous allons voir, cependant ce dernier nécessite d’avoir un modèle statistique de l’image  $v$ .

### 2.2.2 Le filtre de Wiener

Pour construire le filtre de Wiener, on suppose connu un modèle d’images. Plus précisément, on suppose que l’image de départ appartient à un ensemble d’images qui ont toutes la même covariance (voir le TP 3). On suppose donc connu, pour tout  $(k, l) \in \{1, \dots, N\}^2$ ,

$$\mathbb{E}(|\hat{v}_{k,l}|^2).$$

Cette hypothèse est raisonnable, si l’on sait par avance la nature de l’image que l’on cherche (la covariance d’une image astronomique est très différente de celle d’une image satellite représentant une ville).

À partir de cette simple hypothèse, on peut déterminer le noyau de convolution donnant “en moyenne” le meilleur résultat possible.

**Proposition 8** *Soit  $v$ , une suite finie, définie sur  $\{1, \dots, N\}^2$ , et construite comme une réalisation d’un tirage aléatoire, de covariance donnée. Soit  $h$  un noyau de convolution et  $b$  un bruit Gaussien de variance  $\sigma^2$ . On sait alors que  $\hat{h}$ , défini par sa transformée de Fourier*

$$\hat{h}_{k,l} = \frac{\hat{h}_{k,l}^* \mathbb{E}(|\hat{v}_{k,l}|^2)}{|\hat{h}_{k,l}|^2 \mathbb{E}(|\hat{v}_{k,l}|^2) + N^2 \sigma^2}, \quad (2.4)$$

*pour  $(k, l) \in \{1, \dots, N\}^2$ , minimise, en moyenne, la norme de l’erreur au carrée. Plus précisément,  $\tilde{h}$  minimise*

$$\mathbb{E}_{v,b} \left( \|\tilde{h} * u - v\|^2 \right),$$

*où  $u = h * v + b$  et  $\mathbb{E}_{v,b}$  représente l’espérance sur les deux variables aléatoires  $v$  et  $b$ .*

*Preuve.*

On a, en effet,

$$\begin{aligned}
\mathbb{E}_{v,b} \left( \|\tilde{h} * u - v\|^2 \right) &= \frac{1}{N^2} \mathbb{E}_{v,b} \left( \|\hat{h}\hat{u} - \hat{v}\|^2 \right) \\
&= \frac{1}{N^2} \mathbb{E}_{v,b} \left( \|\hat{h}(\hat{h}\hat{v} + \hat{b}) - \hat{v}\|^2 \right) \\
&= \frac{1}{N^2} \mathbb{E}_{v,b} \left( \|(\hat{h}\hat{h} - 1)\hat{v} + \hat{h}\hat{b}\|^2 \right) \\
&= \frac{1}{N^2} \mathbb{E}_{v,b} \left( \sum_{k,l=1}^N \left( (\hat{h}_{k,l}\hat{h}_{k,l} - 1)\hat{v}_{k,l} + \hat{h}_{k,l}\hat{b}_{k,l} \right) \left( (\hat{h}_{k,l}^*\hat{h}_{k,l}^* - 1)\hat{v}_{k,l}^* + \hat{h}_{k,l}^*\hat{b}_{k,l}^* \right) \right) \\
&= \frac{1}{N^2} \mathbb{E}_{v,b} \left( \sum_{k,l=1}^N |\hat{h}_{k,l}\hat{h}_{k,l} - 1|^2 |\hat{v}_{k,l}|^2 + (\hat{h}_{k,l}\hat{h}_{k,l} - 1)\hat{v}_{k,l}\hat{h}_{k,l}^*\hat{b}_{k,l}^* \right. \\
&\quad \left. + (\hat{h}_{k,l}^*\hat{h}_{k,l}^* - 1)\hat{v}_{k,l}^*\hat{h}_{k,l}\hat{b}_{k,l} + |\hat{h}_{k,l}\hat{b}_{k,l}|^2 \right) \\
&= \frac{1}{N^2} \sum_{k,l=1}^N |\hat{h}_{k,l}\hat{h}_{k,l} - 1|^2 \mathbb{E}_{v,b}(|\hat{v}_{k,l}|^2) + |\hat{h}_{k,l}|^2 \mathbb{E}_{v,b}(|\hat{b}_{k,l}|^2).
\end{aligned}$$

Car, comme les  $\hat{b}_{k,l}^*$  et les  $\hat{v}_{k,l}$  sont indépendants, on a, pour tout  $(k,l) \in \{1, \dots, N\}^2$ ,

$$\mathbb{E}_{v,b} \left( (\hat{h}_{k,l}\hat{h}_{k,l} - 1)\hat{v}_{k,l}\hat{h}_{k,l}^*\hat{b}_{k,l}^* \right) = \hat{h}_{k,l}^*(\hat{h}_{k,l}\hat{h}_{k,l} - 1) \mathbb{E}_{v,b}(\hat{v}_{k,l}) \mathbb{E}_{v,b}(\hat{b}_{k,l}^*) = 0$$

(même chose pour l'autre terme).

On en déduit donc que

$$\mathbb{E}_{v,b} \left( \|\tilde{h} * u - v\|^2 \right) = \frac{1}{N^2} \sum_{k,l=1}^N |\hat{h}_{k,l}\hat{h}_{k,l} - 1|^2 \mathbb{E}_v(|\hat{v}_{k,l}|^2) + |\hat{h}_{k,l}|^2 N^2 \sigma^2. \quad (2.5)$$

Or, pour tout  $(k,l) \in \{1, \dots, N\}^2$ ,

$$\begin{aligned}
|\hat{h}_{k,l}\hat{h}_{k,l} - 1|^2 &= (\hat{h}_{k,l}\hat{h}_{k,l} - 1)(\hat{h}_{k,l}^*\hat{h}_{k,l}^* - 1) \\
&= |\hat{h}_{k,l}|^2 |\hat{h}_{k,l}|^2 - \hat{h}_{k,l}\hat{h}_{k,l} - \hat{h}_{k,l}^*\hat{h}_{k,l}^* + 1.
\end{aligned}$$

On obtient donc finalement

$$\begin{aligned}
\mathbb{E}_{v,b} \left( \|\tilde{h} * u - v\|^2 \right) &= \frac{1}{N^2} \sum_{k,l=1}^N |\hat{h}_{k,l}|^2 \left( |\hat{h}_{k,l}|^2 \mathbb{E}_v(|\hat{v}_{k,l}|^2) + N^2 \sigma^2 \right) \\
&\quad - \left( \hat{h}_{k,l}\hat{h}_{k,l} + \hat{h}_{k,l}^*\hat{h}_{k,l}^* \right) \mathbb{E}_v(|\hat{v}_{k,l}|^2) + \mathbb{E}_v(|\hat{v}_{k,l}|^2). \quad (2.6)
\end{aligned}$$

Notons, pour tout  $(k,l) \in \{1, \dots, N\}^2$ ,

$$A_{k,l} = |\hat{h}_{k,l}|^2 \mathbb{E}_v(|\hat{v}_{k,l}|^2) + N^2 \sigma^2$$

et

$$B_{k,l} = \mathbb{E}_v(|\hat{v}_{k,l}|^2),$$

on peut séparer, dans (2.6), les termes venant des parties réelles et imaginaires de  $\hat{h}_{k,l}$ . On obtient alors

$$\begin{aligned}
\mathbb{E}_{v,b} \left( \|\tilde{h} * u - v\|^2 \right) &= \frac{1}{N^2} \sum_{k,l=1}^N A_{k,l} \left( \operatorname{Re}(\hat{h}_{k,l})^2 + \operatorname{Im}(\hat{h}_{k,l})^2 \right) \\
&\quad - 2B_{k,l} \left( \operatorname{Re}(\hat{h}_{k,l})\operatorname{Re}(\hat{h}_{k,l}) - \operatorname{Im}(\hat{h}_{k,l})\operatorname{Im}(\hat{h}_{k,l}) \right) + B_{k,l}, \quad (2.7)
\end{aligned}$$

car

$$\begin{aligned}\hat{h}_{k,l}\hat{h}_{k,l} + \hat{h}_{k,l}^*\hat{h}_{k,l}^* &= 2\operatorname{Re}(\hat{h}_{k,l}\hat{h}_{k,l}) \\ &= 2\left(\operatorname{Re}(\hat{h}_{k,l})\operatorname{Re}(\hat{h}_{k,l}) - \operatorname{Im}(\hat{h}_{k,l})\operatorname{Im}(\hat{h}_{k,l})\right).\end{aligned}$$

Ainsi, on peut récrire (2.7) sous la forme

$$\mathbb{E}_{v,b}\left(\|\tilde{h} * u - v\|^2\right) = \frac{1}{N^2} \sum_{k,l=1}^N f_{k,l}^-\left(\operatorname{Re}(\hat{h}_{k,l})\right) + f_{k,l}^+\left(\operatorname{Im}(\hat{h}_{k,l})\right),$$

avec

$$f_{k,l}^-(t) = A_{k,l}t^2 - 2B_{k,l}\operatorname{Re}(\hat{h}_{k,l})t + \frac{B_{k,l}}{2}$$

et

$$f_{k,l}^+(t) = A_{k,l}t^2 + 2B_{k,l}\operatorname{Im}(\hat{h}_{k,l})t + \frac{B_{k,l}}{2}.$$

Comme toute parabole,  $f_{k,l}^-$  atteint son minimum au point qui annule sa dérivée (on a bien  $A_{k,l} > 0$ ). Ce point est solution de l'équation

$$2A_{k,l}t - 2B_{k,l}\operatorname{Re}(\hat{h}_{k,l}) = 0$$

On obtient donc que la meilleure valeur possible pour  $\operatorname{Re}(\hat{h}_{k,l})$  est

$$\operatorname{Re}(\hat{h}_{k,l}) = \frac{B_{k,l}\operatorname{Re}(\hat{h}_{k,l})}{A_{k,l}} = \frac{\mathbb{E}_v(|\hat{v}_{k,l}|^2)\operatorname{Re}(\hat{h}_{k,l})}{|\hat{h}_{k,l}|^2\mathbb{E}_v(|\hat{v}_{k,l}|^2) + N^2\sigma^2}.$$

De même, on a

$$\operatorname{Im}(\hat{h}_{k,l}) = \frac{-B_{k,l}\operatorname{Im}(\hat{h}_{k,l})}{A_{k,l}} = \frac{\mathbb{E}_v(|\hat{v}_{k,l}|^2)\operatorname{Im}(\hat{h}_{k,l})}{|\hat{h}_{k,l}|^2\mathbb{E}_v(|\hat{v}_{k,l}|^2) + N^2\sigma^2}.$$

On a donc finalement que

$$\hat{h}_{k,l} = \frac{\mathbb{E}_v(|\hat{v}_{k,l}|^2)\hat{h}_{k,l}^*}{|\hat{h}_{k,l}|^2\mathbb{E}_v(|\hat{v}_{k,l}|^2) + N^2\sigma^2}$$

minimise

$$\mathbb{E}_{v,b}\left(\|\tilde{h} * u - v\|^2\right).$$

□

L'esprit de cette proposition est bien illustré par (2.5). En effet, on voit que pour chaque  $(k,l) \in \{1, \dots, N\}^2$  :

- soit  $\hat{h}_{k,l}$  est proche de  $\frac{1}{\hat{h}_{k,l}}$ , auquel cas on perd peu d'informations mais, par contre, le bruit est fort.
- soit  $\hat{h}_{k,l}$  est proche de 0, auquel cas on perd beaucoup d'informations mais, par contre, le bruit est faible.

Il y a une balance entre ces deux cas. Elle est résolue de façon optimale par le filtre de Wiener.

En ce qui concerne le filtre de Wiener lui même, on voit que si  $|\hat{h}_{k,l}|$  est faible ou nul, au dénominateur le terme  $N^2\sigma^2$  est prépondérant. Quant au numérateur, il est lui aussi faible ou nul.

Par contre, aux fréquences contenant beaucoup d'informations (celles où  $\mathbb{E}(|\hat{v}_{k,l}|^2)$  est grande) et  $|\hat{h}_{k,l}|$  est fort, le terme  $N^2\sigma^2$  n'intervient presque plus. Pour ces fréquences, on a  $\hat{h}_{k,l} \simeq \frac{1}{\hat{h}_{k,l}}$ .

Bien sûr, le filtre de Wiener s'applique aussi au cas du débruitage. En effet, le débruitage correspond à la déconvolution avec un noyau de convolution tel que, pour tout  $v$ ,  $h * v = v$ . Si l'on exprime ceci en

fréquence, on voit qu'il faut prendre, pour avoir ces égalités,  $\hat{h} \equiv 1$  (car on cherche à avoir  $\hat{h}\hat{v} = \hat{v}$ ). En appliquant (2.4), on trouve la transformée de Fourier du filtre de Wiener, pour le débruitage. Elle vaut

$$\hat{h}_{k,l} = \frac{\mathbb{E}_v(|\hat{v}_{k,l}|^2)}{\mathbb{E}_v(|\hat{v}_{k,l}|^2) + N^2\sigma^2},$$

pour  $(k, l) \in \{1, \dots, N\}^2$ .

En ce qui concerne la matrice covariance, idéalement, si on sait que l'on cherche toujours à déconvoluer des images d'un certain type (par exemple : uniquement des visages), on peut faire la moyenne, pour un très grand nombre d'images de ce type, du carré du module des transformées de Fourier de ces images, pour obtenir une bonne approximation des  $\mathbb{E}(|\hat{v}_{k,l}|^2)$ .

## 2.3 L'interpolation

D'une façon générale, l'*interpolation* consiste à chercher une valeur raisonnable d'une fonction en un point, à partir des valeurs de cette fonction en d'autres points. Dans ce contexte, le cas des images est un peu particulier puisque nous connaissons l'image (tout du moins en partie) sur une grille régulière. Une autre caractéristique particulière aux images est qu'elles contiennent essentiellement des basses fréquences. Ainsi, il est légitime d'appliquer le théorème de Shannon.

### Proposition 9 (Shannon)

Soit  $v \in L^1([0, N]^2)$ , une fonction périodique de période  $N$ , telle que, pour tout  $(k, l) \notin \{-\frac{N}{2}+1, \dots, \frac{N}{2}-1\}^2$ ,  $\hat{v}_{\frac{k}{N}, \frac{l}{N}} = 0$ . Alors, pour tout  $(x, y) \in [0, N]^2$ ,

$$v(x, y) = \sum_{m, n=1}^N v(m, n) \text{sinc}^D(x - m) \text{sinc}^D(y - n),$$

avec

$$\text{sinc}^D(t) = \begin{cases} \frac{\sin(\pi \frac{(N-1)t}{N})}{N \sin(\pi \frac{t}{N})} & , \text{ si } t \notin N\mathbb{Z} \\ \frac{N-1}{N} & , \text{ sinon.} \end{cases}$$

Il y a évidemment un lien entre ce sinus cardinal discret et celui que nous avons déjà vu. La seule différence est due à la normalisation. Ce que dit cette proposition est que, sous réserve de l'hypothèse que nous faisons sur la transformée de Fourier de  $v$ , on peut déterminer sa valeur en tout point de  $[0, N]^2$  à partir de ses valeurs aux points de  $\{1, \dots, N\}^2$ . Ainsi, si une image analogique vérifie l'hypothèse (ce qui revient à dire qu'il n'y a pas d'aliasing au cours de l'échantillonnage), on connaît l'image analogique à partir de l'image numérique (modulo les autres dégradations). Ceci constitue un argument supplémentaire pour choisir dans la chaîne image, un couple noyau de convolution/échantillonnage permettant d'annuler (ou pour le moins de réduire) l'aliasing.

Il nous manque quelques éléments pour écrire la preuve dans le cadre de fonctions de  $L^1([0, N]^2)$ . Nous allons donc énoncer un résultat analogue dans le cas de suites finies. Nous pouvons, en effet, faire la preuve complète dans ce cadre.

### Proposition 10 (Shannon discret)

Soit  $v$ , une suite finie, définie sur  $\{1, \dots, KN\}^2$  telle que, pour tout  $(k, l) \notin \{-\frac{N}{2}+1, \dots, \frac{N}{2}-1\}^2$ ,  $\hat{v}_{k,l} = 0$ . Alors, pour tout  $(m, n) \in \{1, \dots, KN\}^2$ ,

$$v_{m,n} = \sum_{m', n'=1}^N v_{Km', Kn'} (\text{sinc}^D)_{m-Km'} (\text{sinc}^D)_{n-Kn'}, \quad (2.8)$$

avec, pour tout  $m \in \{1, \dots, KN\}$ ,

$$(\text{sinc}^D)_m = \begin{cases} \frac{\sin(\pi \frac{(N-1)m}{KN})}{N \sin(\pi \frac{m}{KN})} & , \text{ si } m \notin KN\mathbb{Z} \\ \frac{N-1}{N} & , \text{ sinon.} \end{cases}$$



*Preuve.*

Notons

$$w_{m,n} = \sum_{m',n'=1}^N v_{Km',Kn'} (\text{sinc}^D)_{m-Km'} (\text{sinc}^D)_{n-Kn'}.$$

Si l'on note  $u$  la suite finie, définie pour  $(m,n) \in \{1, \dots, KN\}^2$ , par

$$u_{m,n} = \begin{cases} v_{m,n} & , \text{ s'il existe } (m',n') \in \mathbb{Z}, \text{ tel que } (m,n) = (Km',Kn') \\ 0 & \text{ sinon.} \end{cases}$$

on a

$$w_{m,n} = (u * \text{sinc}^{2D})_{m,n},$$

avec  $(\text{sinc}^{2D})_{m,n} = (\text{sinc}^D)_m (\text{sinc}^D)_n$ , pour tout  $(m,n) \in \{1, \dots, KN\}^2$ .

On a donc, pour tout  $(k,l) \in \{1, \dots, KN\}^2$ ,

$$\hat{w}_{k,l} = \widehat{u_{k,l}(\text{sinc}^{2D})}_{k,l} \quad (2.9)$$

On peut aussi montrer que

$$\widehat{(\text{sinc}^{2D})}_{k,l} = \begin{cases} K^2 & , \text{ si } (k,l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2 \\ 0 & , \text{ sinon.} \end{cases}$$

Ceci n'est pas difficile à voir. Il suffit de montrer que la transformée de Fourier inverse du terme de droite vaut la fonction  $\text{sinc}^{2D}$  apparaissant dans le terme de gauche. (La fonction  $\text{sinc}^{2D}$  a, en fait, simplement été construite à partir de la transformée de Fourier ci-dessus.) Le calcul est similaire à celui que nous avons fait dans la deuxième partie de la preuve de la proposition sur l'inversion de la transformée de Fourier.

Par ailleurs, on a, pour tout  $(k,l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2$ ,

$$\begin{aligned} \hat{u}_{k,l} &= \sum_{m,n=1}^{KN} u_{m,n} e^{-2i\pi \frac{km+ln}{KN}} \\ &= \sum_{m',n'=1}^N v_{Km',Kn'} e^{-2i\pi \frac{kKm'+lKn'}{KN}} \\ &= \sum_{m',n'=1}^N v_{Km',Kn'} e^{-2i\pi \frac{km'+ln'}{N}}. \end{aligned}$$

On reconnaît la transformée de Fourier du sous-échantillonnage, dans un rapport  $K$ , de  $v$ . On peut donc appliquer la Proposition 5. On obtient alors, pour tout  $(k,l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2$ ,

$$\begin{aligned} \hat{u}_{k,l} &= \frac{1}{K^2} \sum_{t_1=0}^{K-1} \sum_{t_2=0}^{K-1} \hat{v}_{k+t_1N, l+t_2N} \\ &= \frac{1}{K^2} \hat{v}_{k,l}, \end{aligned}$$

car, si  $t_1 \neq 0$  ou  $t_2 \neq 0$ , comme  $(k,l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2$ ,  $(k+t_1N, l+t_2N) \notin \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2$  et donc  $\hat{v}_{k+t_1N, l+t_2N} = 0$ .

Finalement, on obtient que, pour tout  $(k,l) \in \{1, \dots, KN\}^2$ ,

$$\hat{w}_{k,l} = \hat{v}_{k,l}.$$

On a donc bien  $v = w$ . □

### 2.3.1 Application : Le zoom

L'application la plus évidente de la Proposition 10 est le zoom d'une image. On peut obtenir cet effet en rajoutant des pixels entre les pixels. Si l'image dont on dispose a été convenablement échantillonnée (pas ou peu d'aliasing), on peut directement appliquer le Proposition 10.

Par contre, comme le support (en espace) du  $\text{sinc}^D$  est étendu sur toute l'image, appliquer directement (2.8) nous conduirait à une complexité de l'ordre de  $N^2(KN)^2$ . On préfère donc calculer l'image zoomée en utilisant la formule équivalente (2.9). On peut ainsi faire la convolution dans le domaine de Fourier, et on obtient une complexité de l'ordre de  $(KN)^2 \log(KN)$ .

Pour zoomer une suite finie  $u$ , définie sur  $\{1, \dots, N\}^2$ , on aboutit à l'algorithme suivant :

- On construit une image  $v$  de taille  $KN \times KN$  telle que

$$v_{m,n} = \begin{cases} u_{m',n'} & , \text{ s'il existe } (m', n') \in \{1, \dots, N\}^2, \text{ tel que } (m, n) = (Km', Kn') \\ 0 & , \text{ sinon.} \end{cases}$$

- On calcule la transformée de Fourier de  $v$
- On annule toutes les fréquences de  $\hat{v}$  qui n'appartiennent pas à  $\{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2$  (par exemple en multipliant  $\hat{v}$  par le masque que l'on a construit dans le TP 3).
- On calcule la transformée de Fourier inverse du résultat.

Notez que l'on aurait pu gagner du temps de calcul, en calculant simplement la transformée de Fourier de  $u$  et en plaçant les coefficients de Fourier au bon endroit dans le domaine de Fourier d'une image de taille  $KN \times KN$ .

La plupart des algorithmes pour zoomer des images cherchent à calculer l'interpolation que nous venons de faire (qui est considérée comme optimale, du fait de la proposition 10). L'objectif dans ce domaine est de gagner du temps, soit en faisant les calculs de façon plus astucieuse, soit en approximant cet algorithme. Les algorithmes basés sur les splines semblent aujourd'hui être les plus efficaces.

### 2.3.2 Application : La translation d'une image

Une autre application possible est la translation d'une image. Plus précisément, la translation par un vecteur  $(t_x, t_y) \in \mathbb{R}^2$  revient à chercher les valeurs d'une image aux points  $(m - t_x, m - t_y)$ . Ceci ne pose pas de difficulté lorsque  $(t_x, t_y) \in \mathbb{Z}^2$ , mais nous force à interpoler la suite finie, que l'on a initialement, lorsque  $(t_x, t_y) \notin \mathbb{Z}^2$ . On peut faire cette interpolation à l'aide de la proposition 9.

Pour une suite finie  $u$ , définie sur  $\{1, \dots, N\}^2$ , on note  $INT(u)$  la fonction de  $L^1([0, N]^2)$  telle que

$$(\widehat{INT(u)})_{\frac{k}{N}, \frac{l}{N}} = \begin{cases} \hat{u}_{k,l} & , \text{ si } (k, l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2 \\ 0 & , \text{ sinon.} \end{cases} \quad (2.10)$$

On définit alors notre opérateur de translation  $\tau_{t_x, t_y}$  par

$$(\tau_{t_x, t_y} u)_{m,n} = INT(u)(m - t_x, n - t_y).$$

En fait, ici nous n'avons pas besoin de calculer entièrement  $INT(u)$ . On peut raisonner dans le domaine de Fourier. En effet, si l'on note  $\tilde{\tau}_{t_x, t_y}$  la translation des fonctions de  $L^1([0, N]^2)$ , définie par

$$\tilde{\tau}_{t_x, t_y} f(x, y) = f(x - t_x, y - t_y),$$

on a,

$$(\tau_{t_x, t_y} u)_{m,n} = (\tilde{\tau}_{t_x, t_y} INT(u))_{\frac{k}{N}, \frac{l}{N}}(m, n).$$

De plus, pour tout  $(k, l) \in \mathbb{Z}$ ,

$$\begin{aligned} \widehat{\tau_{t_x, t_y} u}_{\frac{k}{N}, \frac{l}{N}} &= \int_0^N \int_0^N \tilde{\tau}_{t_x, t_y} f(x, y) e^{-2i\pi \frac{kx+ly}{N}} dx dy \\ &= \int_0^N \int_0^N f(x - t_x, y - t_y) e^{-2i\pi \frac{kx+ly}{N}} dx dy \\ &= \int_{-t_x}^{N-t_x} \int_{-t_y}^{N-t_y} f(x, y) e^{-2i\pi \frac{kx+ly}{N}} e^{-2i\pi \frac{kx+ly}{N}} dx dy \\ &= e^{-2i\pi \frac{kt_x+lt_y}{N}} \hat{f}_{\frac{k}{N}, \frac{l}{N}} \end{aligned}$$

(on intègre sur une période et il est donc possible de réarranger le domaine d'intégration de manière à le ramener à  $[0, N]^2$ ).

De plus, pour tout  $(k, l) \in \{-\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}^2$ ,

$$\begin{aligned} \widehat{(\tau_{t_x, t_y} u)}_{k, l} &= \widehat{(\tilde{\tau}_{t_x, t_y} INT(u))}_{\frac{k}{N}, \frac{l}{N}} \quad (\text{car, du fait de (2.10), il n'y a pas d'aliasing}) \\ &= e^{-2i\pi \frac{kt_x + lt_y}{N}} \widehat{INT(u)}_{\frac{k}{N}, \frac{l}{N}} \\ &= e^{-2i\pi \frac{kt_x + lt_y}{N}} \hat{u}_{k, l}. \end{aligned}$$

Il est donc possible d'appliquer l'algorithme suivant

- On calcule les coefficients de Fourier  $\hat{u}_{k, l}$  de  $u$
- On multiplie chaque coefficient  $\hat{u}_{k, l}$  par  $e^{-2i\pi \frac{kt_x + lt_y}{N}}$
- On calcule la transformée de Fourier inverse du résultat.

Notez ici que pour être rigoureux, il faut aussi annuler les fréquences de  $u$  telles que  $k = \frac{N}{2}$  ou  $l = \frac{N}{2}$ . En effet l'image  $u$  ne satisfait en général pas pleinement l'hypothèse du théorème de Shannon.



## Chapitre 3

# Survol de problèmes et solutions du traitement des images

Nous aborderons dans ce chapitre certains problèmes du traitement des images. Ces problèmes nous ont semblé intéressants de part leurs applications et par le fait que des solutions permettent de les résoudre (de façon plus ou moins satisfaisante).

### 3.1 Le recalage

Le but du *recalage* d'images ("*registration*" ou "*optic flow*", en anglais) est de faire la correspondance entre les points d'une première image et ceux d'une deuxième image représentant la même scène (à quelques mouvements près). De telles images se rencontrent fréquemment (deux images successives dans un film, images satellite d'une même région,...). On peut aussi vouloir comparer une image à une image de référence (en imagerie médicale, notamment).

#### 3.1.1 La méthode de Horn et Schunck

La première hypothèse faite en recalage d'images est qu'un point sur un objet de l'image a toujours la même couleur sur les deux prises de vues. Il est facile de trouver des contre-exemples à cette hypothèse (ex : un objet bouge et n'est plus éclairé de la même façon). Le contre exemple le plus gênant est le cas des *occlusion*. En effet, entre deux images, un objet peut, d'une image à l'autre, être partiellement caché. Dans un tel cas, l'hypothèse que nous venons de faire est clairement fausse (et fausse de façon importante) car toute une partie de  $v^1$  ou de  $v^2$  peut avoir disparu. Nous verrons que l'on "assouplira" cette hypothèse.

Ainsi, si l'on considère deux fonctions  $v^1$  et  $v^2$  de  $L^1([0, N]^2)$  et  $t : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , le déplacement d'un objet entre l'image  $v^1$  et  $v^2$  (tel que pour tout  $(x, y) \in [0, N]^2$ ,  $(x, y) + t(x, y) \in [0, N]^2$ ), on a

$$v^2(x, y) = v^1((x, y) + t(x, y)). \quad (3.1)$$

Le but du recalage est de retrouver  $t(x, y) = (t^1(x, y), t^2(x, y))$ .

En pratique, on ne cherchera pas à avoir exactement (3.1). On commence même par l'approximer par son développement de Taylor

$$v^2(x, y) = v^1(x, y) + \langle \nabla v^1(x, y), t(x, y) \rangle, \quad (3.2)$$

où  $\nabla v^1(x, y) = (\frac{\partial v^1}{\partial x}(x, y), \frac{\partial v^1}{\partial y}(x, y))$  représente le gradient de  $v^1$ . (Notez ici qu'une fonction de  $L^1([0, N]^2)$  n'est pas forcément dérivable. Pour éviter d'introduire une nouvelle notation sur les espace de fonctions, nous supposons implicitement que c'est le cas.)

En effet, en pratique, on ne connaît  $v^1$  que sur une grille discrète  $\{1, \dots, N\}^2$ . L'approximation (3.2) permet d'éviter les calculs de  $v^1((x, y) + t(x, y))$  (le point  $(x, y) + t(x, y)$  n'est, à priori, pas sur la grille). Cette approximation permet aussi d'obtenir une approximation de  $v^1((x, y) + t(x, y))$  qui est linéaire en

$t$ . Ceci se révélera, par la suite, capital. Cette approximation condamne la méthode à n'être appliquée que dans des cas où  $t$  est petit.

Notez que le calcul de  $v^1((x, y) + t(x, y))$  pourrait être fait à l'aide de l'interpolation que nous avons vue au chapitre 2.3 (si l'image ne contient pas d'aliasing, le calcul serait même exact). Par contre dans un tel cas, ce calcul ne serait plus du tout linéaire en  $t$ . Ceci compliquerait considérablement le calcul d'une solution tel que nous le ferons dans la suite (ce calcul devient un problème mathématique difficile).

Par ailleurs, comme nous avons vu que la contrainte est, dans bien des cas, trop rigide, on l'assouplit en cherchant  $t$  tel que

$$\|v^1(x, y) + \langle \nabla v^1(x, y), t(x, y) \rangle - v^2(x, y)\|^2 \leq \tau^2. \quad (3.3)$$

Même avec  $\tau = 0$ , ceci ne suffit pas à déterminer  $t$  (ex : si  $v^1$  et  $v^2$  sont constantes sur une région (respectivement sur  $\Omega^1$  et  $\Omega^2$ ), n'importe quel  $t$  transformant  $\Omega^1$  en  $\Omega^2$  marche). Pour le déterminer complètement, on va utiliser le fait que si un objet bouge dans l'image, on peut s'attendre à avoir un  $t(x, y)$  extrêmement régulier. On va donc chercher, parmi tous les  $t$  qui satisfont (3.3), celui qui est le plus régulier ; dans le sens où il minimise

$$\|\nabla t(x, y)\|^2.$$

Il est possible de montrer que ce problème a une solution dès qu'il existe un  $t$  satisfaisant (3.3). Cette solution est de plus unique, sauf si  $v^1$  est une image constante, auquel cas quel que soit  $(t^1, t^2) \in \mathbb{R}^2$ ,  $(t^1, t^2)$  est solution).

On peut aussi montrer que ceci revient à minimiser (Théorème des multiplicateurs de Lagrange)

$$\|v^1(x, y) + \langle \nabla v^1(x, y), t(x, y) \rangle - v^2(x, y)\|^2 + \lambda \|\nabla t(x, y)\|^2,$$

pour un paramètre  $\lambda$ . Ainsi, on a échangé notre ancien paramètre  $\tau$  avec un autre paramètre  $\lambda$ . Le rôle  $\lambda$  est simple à comprendre, plus  $\lambda$  est grand, plus le terme  $\|\nabla t(x, y)\|^2$  a d'importance (et donc  $t$  est régulier) ; plus  $\lambda$  est petit, plus  $\|v^1(x, y) + \langle \nabla v^1(x, y), t(x, y) \rangle - v^2(x, y)\|^2$  a d'importance (au risque d'avoir des problèmes pour recalculer les objets dont l'intensité a changé).

Numériquement, on est ramené à minimiser

$$\sum_{m,n=1}^N (v_{m,n}^1 + \langle \nabla v_{m,n}^1, t_{m,n} \rangle - v_{m,n}^2)^2 + \lambda (|\nabla t_{m,n}^1|^2 + |\nabla t_{m,n}^2|^2),$$

avec

$$\nabla v_{m,n} = (v_{m,n} - v_{m-1,n}, v_{m,n} - v_{m,n-1}),$$

pour toute suite finie  $v$ . (Afin de ne pas alourdir inutilement les notations, nous ne détaillons pas la définition du gradient discret aux bords de l'image.)

Il existe un bon nombre de techniques standards pour trouver le minimum d'une énergie de ce type. Cette dernière vérifie, en effet, toutes les propriétés nécessaires au bon fonctionnement de ces méthodes (elle est convexe<sup>1</sup> et coercive).

### 3.1.2 Applications

#### L'augmentation de résolution

Cette application est souvent utilisée pour retrouver des détails sur un film de mauvaise qualité. En effet, en recalant deux images successives (on peut en fait recalculer, sans difficultés, toute une séquence du film), on obtient une image définie sur la grille irrégulière

$$\{(x, y) \in [0, N]^2, \text{ il existe } (m, n) \in \{1, \dots, N\}^2, \text{ tel que } (x, y) = (m, n) \text{ ou } (x, y) = (m, n) + t_{m,n}\},$$

où  $t$  représente le résultat du recalage entre les deux images de départ.

Il faut encore restaurer cette image de manière à l'avoir sur une nouvelle grille régulière (par exemple  $\{1, \dots, 2N\}^2$ ). On peut aussi profiter de cette étape pour déconvoluer et débruiter l'image. Notez que nous

---

<sup>1</sup>L'énergie ne serait pas convexe si on n'avait pas approximé  $v^1((x, y) + t(x, y))$  convenablement (en l'occurrence de façon linéaire en  $t$ ).

n'avons pas vu de méthode pour faire cette opération (qui s'appelle un *ré-échantillonnage*). Notamment, toutes les méthodes basées sur la transformée de Fourier ne peuvent pas s'appliquer car cette dernière n'a les bonnes propriétés que nous avons mentionnées dans le chapitre 1.2 que si l'image est définie sur une grille régulière.

### Le codage de vidéos

Le principe de ce genre de codage est assez simple. Il part du constat que dans un film, les images successives sont proches les unes des autres. De plus, entre deux images successives, un objet solide (ou le fond de l'image) a un déplacement facilement codable (par exemple : entre deux images, l'objet, sur le film, est simplement translaté). Ainsi, le résultat du recalage entre ces deux images devrait être facile à coder (par exemple :  $t$  est constant sur plusieurs régions de l'image).

De ce fait, pour compresser une séquence de film, il semble moins coûteux de coder la première image et de coder ensuite les différents recalages entre les images de la séquence.

On voit ici une autre application du recalage qui est de *segmenter* (séparer en plusieurs éléments) une image en objets distincts (ceux qui subissent plus ou moins le même déplacement entre les différentes images d'un film). Ceci peut notamment être utile pour suivre un objet dans une scène.

### La stabilisation de films

En effet, dans beaucoup de films amateurs, la caméra n'est pas fixe et le film présente des oscillations. Il est possible d'enlever ces oscillations à l'aide du recalage. Dans un tel cas, le décalage entre deux images prend la forme

$$v^2(x, y) = v^1((x, y) + t_m(x, y) + t_o)$$

où  $t_m$  est le déplacement dans l'image dû aux mouvements des objets filmés et  $t_o$  est le déplacement dû aux mouvements de la caméra entre les deux prises de vues. (On suppose ici qu'il y a simplement une translation, le mouvement est donc uniforme sur l'image. On pourrait supposer d'autres types de mouvement sans que cela pose de réelle difficulté.)

On peut alors stabiliser le film en calculant la moyenne du déplacement sur l'image. On peut aussi (c'est souvent plus réaliste) restreindre cette moyenne à des points ayant un déplacement proche (il est même possible ici de demander à un utilisateur quelle est la partie de l'image qu'il souhaite garder immobile). Une fois calculée une approximation  $t_o$ , il ne reste plus qu'à translater la deuxième image (comme nous l'avons vu dans le chapitre 2.3.2). Pour un meilleur rendu, il est souhaitable d'enlever la partie du bord des images ne se trouvant pas dans toutes les images.

### La stéréoscopie

La stéréoscopie consiste à reconstruire le relief à partir de deux vues, prises de deux points de vues différents d'une même scène. Cette application est importante en robotique et en géographie (pour faire des cartes du relief à partir de photos satellites).

Avec un modèle de *camera pinhole* ("trou d'épingle" en français), et dans un cas où tous les paramètres de calibration des appareils sont connus (il faut les déterminer dans le cas général), on peut déterminer la distance entre les appareils de mesure et l'objet observé.

Nous allons le détailler dans le cas simple : en dimension 1, quand les appareils de mesure sont orientés dans une même direction, sont tous les deux situés dans un plan orthogonal à cette direction et ont la même distance focale (qui est connue). Cette situation est représentée sur la Figure 3.1.

Dans la suite, on prend les notations décrites sur la Figure 3.1, où  $P$  désigne le point observé,  $p_1$  et  $p_2$  la position de  $P$  sur les images,  $O_1$  et  $O_2$  dépendent de l'appareil de mesure et  $C_1$  et  $C_2$  sont les centres des images. Le tout est en dimension 1. Le passage à la dimension 2 ne pose pas de réelle difficulté.

On a alors en appliquant le Théorème de Thalès

$$\frac{\overline{O_1 V_1}}{\overline{O_1 C_1}} = \frac{\overline{V_1 P}}{\overline{C_1 p_1}},$$

donc

$$\overline{O_1 V_1} \overline{C_1 p_1} = \overline{V_1 P} \overline{O_1 C_1}.$$

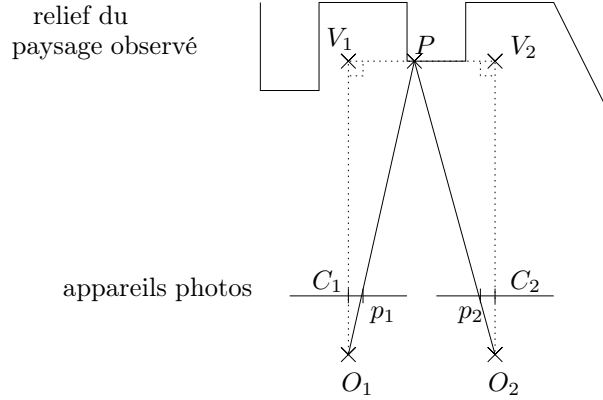


FIG. 3.1 – Stéréoscopie (dessin en dimension 1): un même point est vu par deux appareils de mesure.

De même, on a

$$\overline{O_2 V_2} \overline{C_2 p_2} = \overline{V_2 P} \overline{O_2 C_2}.$$

On a donc, comme  $\overline{O_1 V_1} = \overline{O_2 V_2}$  (même altitude) et  $\overline{O_1 C_1} = \overline{O_2 C_2}$  (même distance focale)

$$\begin{aligned} \overline{O_1 V_1} (\overline{C_1 p_1} - \overline{C_2 p_2}) &= \overline{O_1 C_1} (\overline{V_1 P} - \overline{V_2 P}) \\ &= \overline{O_1 C_1} \overline{V_1 V_2} \\ &= \overline{O_1 C_1} \overline{O_1 O_2}. \end{aligned}$$

Ainsi, comme  $\overline{C_1 p_1} - \overline{C_2 p_2}$  représente précisément le décalage calculé lors du recalage (modulo le fait que nous sommes ici en dimension 1), on obtient que  $\overline{O_1 V_1}$ , qui représente l'altitude du point  $P$ , vaut

$$\overline{O_1 V_1} = \frac{\overline{O_1 C_1} \overline{O_1 O_2}}{\overline{C_1 p_1} - \overline{C_2 p_2}},$$

$\overline{O_1 C_1}$  et  $\overline{O_1 O_2}$  étant supposés connus.

## 3.2 L'indexation par le contenu

L'*indexation* a pour but de retrouver une image dans une base de données d'images. Ceci peut se faire dans des contextes très divers. On peut notamment en distinguer deux :

- Ceux où l'utilisateur entre une image et attend du logiciel qu'il retrouve des images similaires sans interaction homme/machine (recherche d'empreintes digitales, reconnaissance de visages,...). Dans ce cas, l'utilisateur n'est pas sensé donner son avis ou n'a pas la compétence pour avoir un avis.
- Ceux où l'utilisateur peut guider la recherche des bonnes images par une action simple (par exemple : en notant les images résultats qui lui sont proposées et en itérant les recherches). C'est le cas dès que l'utilisateur connaît le type d'images qu'il veut trouver (exemple : images de déserts, de villes, d'animaux,...)

Les méthodes que nous considérerons visent à répondre au premier cas d'utilisation. On supposera ainsi que l'utilisateur fournit une image au système et que ce dernier retrouvera dans la base les images les plus proches de cette *image requête*.

Ce problème est très récent (début 1990) et les solutions apportées ne sont toujours pas pleinement satisfaisantes. Nous allons ici décrire le principe des méthodes aujourd'hui envisagées. Nous l'illustrerons par un exemple.

Le premier problème rencontré est la taille des données. En effet, comparer des images entre elles est une opération trop longue pour être effectuée sur toute une base d'images. La première opération consiste donc à résumer un type d'information contenu dans l'image. On appelle ce résumé la *signature de l'image*. Le temps de calcul de cette signature dépend de la nature de la base de données, mais, si l'on suppose



que celle-ci est déjà constituée, il peut éventuellement être long. En effet on peut calculer les signatures des images de la base de données une fois pour toutes. Au moment de la recherche, ce calcul n'est donc effectué qu'une seule fois, sur l'image requête. De plus, il est souhaitable que la signature consomme peu de mémoire pour qu'elles puissent toutes être chargées en mémoire centrale.

Ensuite, il faut être capable de comparer les signatures entre elles. Ainsi, le système peut retourner à l'utilisateur les images dont les signatures sont les plus proches de celle de l'image requête. Cette comparaison doit être effectuée une fois entre la signature de l'image requête et chaque signature de la base. Elle doit donc être rapide.

### 3.2.1 Les signatures d'images

Ce problème consiste à résumer, à partir de manipulations de l'image, le contenu de cette dernière. C'est un problème délicat (il faut donner un sens à un tableau de réels). Les trois types principaux de contenus que l'on cherche aujourd'hui à exploiter sont :

- La couleur
- La forme
- La texture

Le choix du contenu dépend évidemment de l'application visée (Ex : pour retrouver des empreintes digitales, on préférera la texture). Une indexation idéale devrait pouvoir prendre en compte plusieurs de ces contenus (Ex : Un utilisateur cherchant des images de ballon rouge ne peut y arriver que si la signature de l'image contient au moins ces deux informations (forme et couleur)).

Un autre problème de la définition de la signature est que l'information contenue dans l'image peut, d'une image à une autre image, avoir subi des modifications (ex : translation dans l'image, rotation, changement de contraste, changement d'échelle, bruit sur les pixels...). Il convient donc, autant que possible, de définir une signature dépendant le moins possible de ces modifications. (Tout du moins, si elle en dépend, il faut choisir une mesure de similarité entre les signatures qui n'en dépende pas.)

Nous n'avons pas vu les différents éléments d'analyse d'images nous permettant de définir des signatures de forme et de texture. De plus, les signatures les plus utilisées sont celles qui rendent compte de la couleur. Ces signatures sont basées sur l'histogramme des couleurs.

#### L'histogramme des couleurs

Le calcul de l'histogramme des couleurs consiste simplement à compter, pour chaque couleur  $c$  le nombre de pixels ayant la couleur  $c$  (voir Figure 3.2, pour un histogramme d'image en niveaux de gris). Il faut, bien entendu, pour cela, définir l'ensemble des couleurs que l'on considère. Il faut aussi que cet ensemble de couleurs soit compatible avec ceux des images dont on dispose dans la base d'images.

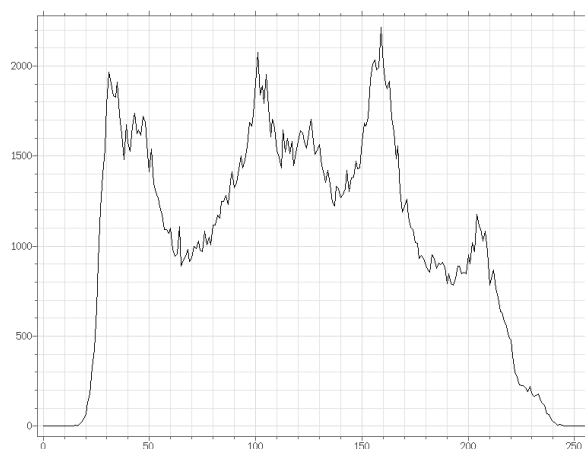


FIG. 3.2 – Histogramme de l'image *barbara.tiff*: le pas de quantification est 1.

De plus, pour raccourcir la signature et renforcer la robustesse vis à vis des faibles changements de contraste et du bruit, il est souhaitable de limiter le nombre de couleurs considérées. On peut pour cela choisir un espace de couleurs  $\mathcal{Col}$  ayant peu d'éléments (un seul pour le bleu, un seul pour le vert, ...). La Figure 3.3 montre un tel histogramme pour une image en niveaux de gris.

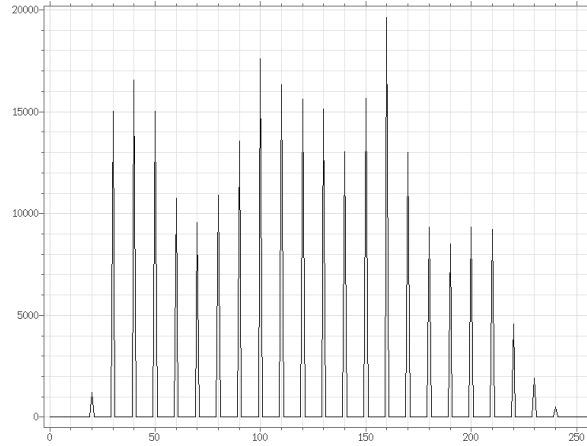


FIG. 3.3 – Histogramme de l'image *barbara.tiff*: le pas de quantification est 10.

Enfin, pour rendre cette signature invariante par changement d'échelle (qui est représentée ici par le nombre de pixels), on peut simplement diviser l'histogramme des couleurs par le nombre de pixels de l'image. On a ainsi la proportion d'une couleur dans une image.

### Remarques sur l'histogramme des couleurs

La première constatation à faire sur l'histogramme des couleurs est que cette signature ne contient aucune information sur la répartition spatiale des couleurs. Ainsi cet histogramme est invariant par toute permutation des pixels.

Ceci peut sembler étrange à première vue, mais ne pose en pratique pas trop de problèmes lorsque la base de données contient effectivement des images. De fait, les pixels de couleurs similaires seront, en général, proches les uns des autres.

Par contre, ce manque d'information spatiale ne permet pas de distinguer un carré rouge d'un cercle rouge. Plus grave, deux images contenant des ballons rouges peuvent avoir des signatures très différentes pour deux raisons :

- L'histogramme des couleurs n'est pas invariant pour des changements d'échelles locaux. Ainsi, si le ballon est représenté avec une taille différente suivant l'image les signatures seront elles aussi différentes.
- L'histogramme des couleurs est une information globale. Ainsi, les couleurs qui entourent le ballon rouge peuvent être très différentes. L'impact des pixels correspondant au ballon peut être mineur. <sup>2</sup>

### 3.2.2 Comparaison de signatures

On peut imaginer beaucoup de méthodes pour comparer des signatures. Les plus simples étant les normes usuelles :

$$\|\mathcal{S}_r - \mathcal{S}_i\|_1 = \sum_{k=1}^K |\mathcal{S}_r(k) - \mathcal{S}_i(k)|, \quad (3.4)$$

<sup>2</sup>Ce cas pourrait être résolu si la comparaison entre les images ne porte que sur la couleur rouge. (On chercherait alors toutes les images ayant une certaine proportion de rouge.) Cependant, une telle comparaison serait inintéressante pour chercher d'autres types d'image. Il est possible en s'autorisant des interactions homme/machine de faire comprendre à la machine que l'utilisateur est intéressé par une couleur donnée.

$$\|\mathcal{S}_r - \mathcal{S}_i\|_2^2 = \sum_{k=1}^K |\mathcal{S}_r(k) - \mathcal{S}_i(k)|^2,$$

$$\|\mathcal{S}_r - \mathcal{S}_i\|_\infty = \max_{k=1, \dots, K} |\mathcal{S}_r(k) - \mathcal{S}_i(k)|,$$

où  $\mathcal{S}_r$  et  $\mathcal{S}_i$  sont respectivement les signatures (ayant  $K$  composantes) de l'image requête et d'une image de la base de données.

Pour une signature basée sur l'histogramme de couleurs, la comparaison la plus souvent utilisée est la norme 1 (celle définie par (3.4)). Ceci est dû au fait qu'elle pénalise autant plusieurs petites erreurs qu'une grosse erreur (par exemple la norme 2 pénalise beaucoup moins plusieurs petites erreurs qu'une grosse).

Comme nous l'avons déjà dit, on peut, pour une application particulière, favoriser une couleur particulière en rajoutant des paramètres  $\alpha(k)$ . La distance serait alors, par exemple,

$$\sum_{k=1}^K \alpha(k) |\mathcal{S}_r(k) - \mathcal{S}_i(k)|.$$

(Comme nous l'avons déjà dit, de tels paramètres n'apporteraient pas grand chose si l'on n'est pas capable de les régler en utilisant une interaction homme/machine.)

Malgré ce que nous venons de dire, il faut avoir à l'esprit que la notion de proximité entre deux images ne devrait idéalement pas être une distance (au sens mathématique du terme). Notamment, il n'y a pas de raison d'avoir

$$C(\mathcal{S}_{i_1}, \mathcal{S}_{i_3}) \leq C(\mathcal{S}_{i_1}, \mathcal{S}_{i_2}) + C(\mathcal{S}_{i_2}, \mathcal{S}_{i_3}),$$

où  $C$  serait le critère de comparaison choisi.

En effet si l'on considère :

- $i_1$  : une image contenant un rond rouge,
- $i_2$  : une image contenant un carré rouge,
- $i_3$  : une image contenant un carré vert,

$i_1$  n'a rien à voir avec  $i_3$  alors que  $i_1$  est proche de  $i_2$  (dans le sens où l'objet représenté a la même couleur) et  $i_2$  est proche de  $i_3$  (dans le sens où l'objet représenté a la même forme).

Ceci est surtout vrai lorsque l'on cherche à créer des signatures représentant des informations de nature différente (couleur, forme et texture).



# Bibliographie

- [1] A. Cohen. *Ondelettes et traitement numérique du signal*. Masson, Paris, 1992.
- [2] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, Boston, 1998.
- [3] M. Misiti, Y Misiti, G. Oppenheim, and J.M. Poggi. *Les ondelettes et leurs applications*. Hermes science, Paris, 2003.