Ecole doctorale de mathématiques, informatique, sciences et technologies de l'information
Université Joseph Fourier-Grenoble
UFR IMA

Problèmes d'échelle dans la segmentation par ondelettes d'images texturées

THÈSE

présentée et soutenue publiquement le 6 décembre 2001

pour l'obtention du

Doctorat de l'université Joseph Fourier – Grenoble 1
(spécialité mathématiques appliquées)

par

Sylvain Meignen

Composition du jury

Président :	Yves Meyer
Rapporteurs :	Raul Gouet Jean-Louis Coatrieux
Examinateurs :	Valérie Perrier Paulo Goncalves
Directeur :	Jacques Demongeot

Laboratoire TIMC-IMAG — UMR 5525

Mis en page avec la classe thloria.

Remerciements

Je tiens tout d'abord à remercier Valérie Perrier, professeur INPG, pour l'intérêt constant qu'elle a témoigné pour mon travail ainsi que pour la disponibilité dont elle a toujours fait preuve.

Je souhaite adresser mes plus vifs remerciements à Yves Meyer, membre de l'Académie des sciences, pour avoir bien voulu présider ce jury.

J'exprime aussi mes remerciements à Jacques Demongeot de m'avoir poussé à faire une thèse.

Je remercie Raul Gouet, professeur à l'université du Chili, d'avoir bien voulu être rapporteur de mon travail de thèse. Je tiens, par ailleurs, à le remercier une nouvelle fois de m'avoir si bien accueilli dans son pays.

J'adresse également mes remerciements à Jean-Louis Coatrieux, professeur à l'université de Rennes I, d'avoir accepté de rapporter mon travail de thèse.

Je remercie enfin Paulo Goncalves, chargé de recherche à l'INRIA, d'avoir bien voulu participer au jury.

Je dédie cette thèse
à Samia et Arnaud,
à mes parents
et à ma famille.

Table des matières

Table des figures

Introduction générale

L'analyse d'images a connu durant ces vingt dernières années un développement considérable. A cela deux raisons : la première est liée au développement d'outils informatiques permettant d'envisager le traitement de structures complexes et la seconde tient au fait que l'analyse d'images se trouve à la confluence de nombreux domaines, par exemple, la biologie, les mathématiques et l'informatique dont les interactions sont grandissantes.

Les questionnements des biologistes sur le fonctionnement de l'oeil humain a donné naissance à de nombreuses recherches en informatique et en mathématiques. Celles-ci portent sur la reconnaissance des formes, le suivi d'objets en mouvement, le débruitage d'images. Tous ces domaines ont le point commun de vouloir modéliser, à l'aide d'outils mathématiques, des phénomènes biologiques pour mieux comprendre leur nature. Parfois, les modèles sont si bien pensés que ceux-ci guident les biologistes dans leur recherches futures.

Citons, par exemple, les découvertes du physiologiste Gray [40] sur le cortex visuel du chat. Celles-ci portent sur la mise en évidence de comportements particuliers lorsque le chat est soumis à des stimuli lumineux. Il est apparu que nous avons, de même que le chat, une faculté plus grande à discerner les lignes verticales ou horizontales que les lignes obliques, ou encore les lignes droites que les lignes courbes.

Des modèles mathématiques permettant de rendre compte de ces phénomènes ont naturellement vu le jour. En termes scientifiques, les cellules du cortex visuel semblent opérer un filtrage sélectif des signaux lumineux que l'on peut modéliser facilement à l'aide de filtres de Gabor [8].

La compréhension du mode de réception des signaux au niveau du cortex visuel est une étape indispensable à la construction de modèles visant à comprendre comment notre cerveau analyse les images. Le domaine de l'analyse d'images est très varié, mais cette thèse s'insère dans celui particulier de la reconnaissance des formes et des structures. La question posée est comment notre cerveau reconnaît dans une image les formes ou les structures qui la composent ?

La réponse la plus souvent rencontrée est que nous procédons par analogie, c'est à dire que l'on repère des zones homogènes au sein des images et les objets sont repérés par la rupture de cette homogénéité. Celle-ci peut revêtir des natures très diverses et parfois complexes. Nous pensons que, selon la nature des objets rencontrés, les mécanismes mis en place pour caractériser leur homogénéité sont différents. Il est à ce jour impossible de caractériser l'ensemble des objets à l'aide d'un seul outil d'analyse.

Dans cette thèse, nous ne tenterons d'analyser qu'un seul type d'objets que l'on nomme textures. Nous nous limiterons à l'étude de certaines d'entre elles qui suscitent le plus grand questionnement quant à leur nature. Notre approche se fera au travers de la construction de modèles permettant de donner sens à la notion d'homogénéité texturale. Puis, nous nous intéressons au problème de segmentation non supervisée d'images texturées. On entend par là que l'on cherche à découper une image en régions homogènes, chacune correspondant à une texture, sans savoir quelle est la nature de ces régions ; c'est l'action effectuée par notre cerveau lorsque l'on cherche à déterminer les objets présents dans une image, sans rien connaître de leur nature. Il faudra donc construire des modèles suffisamment généraux qui permettent de retenir des paramètres discriminants entre les régions.

Dans la deuxième partie de la thèse, nous quitterons le domaine de la reconnaissance des formes pour nous intéresser à des objets particuliers que sont les singularités dans les signaux. Celles-ci peuvent n'avoir aucune signification si, par exemple, il s'agit de bruit ou d'artefacts quelconques dûs au mode d'enregistrement. En revanche, elles peuvent être synonymes de changement de régime dans les signaux ou de présence de bords, en analyse d'images. Nous rejoindrons alors le questionnement de la première partie, en ce que l'on se demandera s'il est préférable d'analyser les stuctures des objets ou leurs bords pour les segmenter. Nous pensons, comme Chan [16], que l'analyse conjointe ces deux types d'informations est profitable.

Dans les exemples que nous traiterons, nous regarderons comment l'analyse des singularités seules peut fournir des informations pertinentes. Nous considèrerons des signaux contenant des portions de nature différente, et nous verrons en quoi un modèle d'analyse de singularités permet de donner du sens à ces différences.

Une troisième et dernière partie de la thèse portera sur l'étude de petits modèles d'oscillateurs neuronaux que l'on cherchera à synchroniser ou à désynchroniser. De nombreuses études sont menées sur ces phénomènes de synchronisation dans le but de comprendre comment sont générées certaines représentations mentales [103]. Néanmoins, le modèle proposé ne prétendra pas à une quelconque plausibilité biologique.

Première partie

Analyse de textures et ondelettes

1

Introduction et Plan

1.1 Introduction à la segmentation de textures

L'analyse des textures et la segmentation d'images texturées ne sont pas des centres d'intérêt nouveaux en traitement d'images. Depuis en effet une trentaine d'années, de nombreux algorithmes ont été proposés dans la littérature afin d'analyser la structure des textures.
La première phase de l'analyse consiste en la définition des textures dont l'on attribue la paternité à Julesz [54], qui a introduit la notion de textons. Ceux-ci peuvent être définis comme la juxtaposition de structures ordonnées qui nous semblent similaires [33].
L'objectif de l'analyse texturale est alors de savoir de quel ordre il s'agit, et, dans un premier temps, de déterminer si la valeur du niveau de gris des pixels ainsi que leur répartition spatiale est de nature stochastique ou déterministe, donnant ainsi naissance à deux approches pour l'analyse texturale, l'une déterministe et l'autre statistique.
Les méthodes statistiques consistent toujours à construire un modèle de texture, à calculer les caractéristiques de la texture, puis à sélectionner les plus pertinentes selon les critères choisis. De nombreux critères devant permettre de distinguer les textures ont été définis [42] [84], les plus simples étant calculés directement sur les niveaux de gris de l'image. On peut citer des critères d'entropie, de contraste, d'énergie ou ceux liés aux matrices de co-occurrence des niveaux de gris de l'image. Dans le cas des approches structurales, il s'agit de définir un ensemble de propriétés spatiales susceptibles de caractériser la texture. De nombreuses études ont porté sur la détection de vallées, de pics ou d'autres caractéristiques géométriques au sein d'une image [86].
Au fil des ans, les caractéristiques directement extraites de l'image sont apparues trop pauvres pour rendre compte de la diversité des phénomènes rencontrés. Concernant l'approche statistique du problème, de nouveaux modèles de texture ont vu le jour sous la forme de champs de Markov [15], Gaussiens ou autobinomiaux. Ces dernières années, l'approche multirésolution est venue sophistiquer les modèles par l'introduction de champs Markoviens liés à différentes échelles de résolution [57].
Parallèlement, l'extraction de caractéristiques par le biais de filtres s'est beaucoup développée, en raison de la mise en évidence du fait que l'action des cellules du cortex visuel pouvait être modélisée par des filtres de Gabor [25]. De nombreux modèles ont alors été définis à partir de ce type de filtres [8] [48] [3]. Dunn [29] a montré l'importance de l'adaptation des paramètres des filtres au problème traité, en particulier pour la segmentation d'images texturées. Le problème majeur posé par l'utilisation de ces filtres étant lié au nombre trés important de paramètres.
Plus récemment, l'avènement des ondelettes et la découverte d'une méthode de calcul rapide de la transformée orthogonale associée [64] ont permis l'application naturelle de l'analyse multiré-

solution au traitement des images et, en particulier, des textures. Chang [16] fut le premier à proposer un algorithme fondé sur la structure naturelle d'arbre découlant de la décomposition en ondelettes et en paquets d'ondelettes pour la classification des textures. En 1995, Unser [95] proposa d'utiliser les frames d'ondelettes, rejoint en cela par Laine [59]. Ce dernier proposa aussi une méthode d'extraction de caractéristiques fondée sur les paquets d'ondelettes [58].
Les approches espace-échelle proposent des décompositions pertinentes des images et fournissent des caractéristiques plus discriminantes que celles associées à une unique échelle [95]. Ce type d'approche nous servira à definir de nouvelles caractéristiques de textures.

1.2 Plan de l'étude

Dans le second chapitre, nous allons présenter les différentes transformations en ondelettes puis, faire un bref historique des méthodes de classification des textures par ondelettes et, enfin, établir le lien avec les bancs de filtres. Nous mettrons alors en évidence, dans un troisième chapitre, les propriétés que ceux-ci devront posséder pour permettre leur utilisation pour le traitement des textures.
Dans un troisième chapitre, nous présenterons les propriétés classiques des bancs de filtres que nous aurons introduits. Nous passerons alors à l'étape de filtrage des images de textures et procéderons, ensuite, à la détermination des caractéristiques de textures. Cette dernière étape nous apparaîtra comme l'étape primordiale du processus de segmentation. Le principe de l'étude sera de construire de nouvelles méthodes de sélection des caractéristiques en se basant sur une analyse critique des méthodes existantes.

Nous nous intéresserons notamment à la segmentation de macro-textures, que nous définissons comme des textures qui sont reconnaissables uniquement à une échelle relativement grande, pour l'analyse et la segmentation desquelles la plupart des algorithmes proposés dans la littérature sont inefficaces.

Ayant mis en place ces nouveaux algorithmes de segmentation, nous nous pencherons enfin sur les propriétés de régularité des textures et de leurs bords.

2

Décompositions en ondelettes et frames d'ondelettes

Dans ce chapitre, nous allons montrer comment s'appliquent les principes d'analyse multi-résolution, de construction base orthonormée d'ondelettes et de frames d'ondelettes au cas des images. Nous présenterons alors des algorithmes classiques de classification texturale et de segmentation d'images texturées.

2.1 L'analyse temps-fréquence des signaux

En analyse des signaux, il est souvent nécessaire de connaître le contenu fréquentiel local d'un signal. Une des méthodes les plus classiques pour obtenir une telle information est fournie par la transformée de Fourier à fenêtre du signal $f(t)$ en t_0 et à la fréquence ω :

$$W_F(t_0, \omega) = \int_{-\infty}^{+\infty} f(t)g(t-t_0)e^{-i\omega t}dt$$

$g(t)$ étant appelée fonction fenêtre. Celle-ci doit être choisie de telle sorte que le signal $f(t)g(t-t_0)$ contienne des informations fréquentielles pertinentes. De ce fait, la fenêtre utilisée doit être adaptée au problème traité.
Cependant, ce choix effectué, demeure le problème que la taille de la fenêtre est fixée une fois pour toutes ; or, il apparaît que l'analyse de structures de taille variable au sein d'un même signal nécessite des fenêtres dont le support doit être variable. Ceci est l'objet de l'analyse temps-échelle.

2.2 L'analyse temps-échelle des signaux

L'analyse par ondelettes des signaux permet de prendre en compte des événements de taille différente par décomposition d'un signal $f(t)$ sur une famille d'ondelettes obtenue par dilatation et translation d'une même ondelette communément appelée, "ondelette mère".
Une ondelette est une fonction Ψ de $L^2(\mathbb{R})$, de moyenne nulle :

$$\int_{-\infty}^{+\infty} \Psi(t)dt = 0$$

telle que $\|\Psi\|_{L^2} = 1$.
Les translatées et dilatées de Ψ sont définies par :

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi(\frac{t-b}{a})$$

où a est un paramètre d'échelle et b une translation temporelle. Les coefficients d'ondelettes sont alors définis par :

$$Wf(b,a) =< f(t), \Psi_{a,b}(t) >= \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{a}}\Psi(\frac{t-b}{a})dt$$

Lorsque a et b varient dans $\mathbb{R}$ et que le signal est continu, cette transformation est appelée transformation continue en ondelettes.
Si la fonction Ψ est réelle et satisfait la condition d'admissibilité :

$$C_\Psi = \int_0^{+\infty} \frac{|\hat{\Psi}(\omega)|^2}{\omega}d\omega < +\infty$$

on dispose d'une formule de reconstruction du signal à partir des coefficients d'ondelettes.

$$f(t) = \frac{1}{C_\Psi}\int_0^{+\infty}\int_{-\infty}^{+\infty} Wf(b,a)\frac{1}{\sqrt{a}}\Psi(\frac{t-b}{a})db\frac{da}{a^2}$$

2.3 Analyse multirésolution et base d'ondelettes

Un des inconvénients majeur de la transformée continue en ondelettes est son temps de calcul élevé, dû au fait que a et b sont respectivement des réels positif et quelconque. L'idée d'associer à une analyse multirésolution, une famille d'ondelettes est due à Y.Meyer [71]. Une analyse multirésolution de $L^2(\mathbb{R})$ est définie par un ensemble de sous-espaces fermés $(V_j)_{j\in\mathbb{Z}}$ tel que :

i) $\forall j \in \mathbb{Z} \quad V_{j+1} \subset V_j$

ii) $\forall (j,k) \in \mathbb{Z}^2, \quad f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j$

iii) $f(t) \in V_j, \quad f(\frac{t}{2}) \in V_{j+1}$

iv) $\lim_{j\to+\infty} V_j = \bigcap_{j=-\infty}^{j=+\infty} V_j = \{\varnothing\}$

v) $\lim_{j\to-\infty} V_j = \overline{\bigcup_{j=-\infty}^{j=+\infty} V_j} = L^2(\mathbb{R})$

vi) Il existe θ tel que $\theta(t-n)_{n\in\mathbb{Z}}$ est une base de Riesz de V_0

On peut alors définir une famille d'ondelettes, construite selon le principe d'une analyse multirésolution, qui soit une base orthonormée de $L^2(\mathbb{R})$. Pour cela, on peut associer à la fonction θ une fonction échelle Φ, d'intégrale égale à 1, dont les translatées forment une base orthonormée de V_0. Il s'ensuit immédiatement que :

$$\Phi_{j,n} = \frac{1}{\sqrt{2^j}}\Phi(\frac{t-2^j n}{2^j})$$

forment une base orthonormée de V_j. Considérons alors l'injection de V_1 dans V_0. On obtient alors l'équation de dilatation suivante :

$$\frac{1}{\sqrt{2}}\Phi(\frac{t}{2}) = \sum_{n=-\infty}^{+\infty} h[n]\Phi(t-n) \quad (1)$$

Dans le domaine de Fourier, cette égalité se réécrit :

$$\hat{\Phi}(\omega) = \frac{1}{\sqrt{2}}\hat{h}(\frac{\omega}{2})\hat{\Phi}(\frac{\omega}{2})$$

Mallat et Meyer [66] ont alors montré que la série de Fourier,$\hat{h}$, des coefficients $h[n]$ (appelé filtre h) vérifie :

$$\forall \omega \in \mathbb{R}, \quad |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2$$
$$\hat{h}(0) = \sqrt{2}$$

Inversement, partant d'une suite $h[n]$ telle que $\hat{h}(\omega)$ vérifie les deux égalités au-dessus et telle que :

$$\inf_{\omega \in [-\frac{\pi}{2}, \frac{\pi}{2}]} |\hat{h}(\omega)| > 0,$$

la fonction :

$$\hat{\Phi}(\omega) = \prod_{p=1}^{\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}}$$

est la transformée de Fourier d'une fonction échelle. Cette propriété permet de définir le filtre h à partir de la fonction échelle Φ et inversement.
On considère alors les espaces $\{W_j\}_{j\in\mathbb{Z}}$ définis par :

$$V_{j-1} = W_j \oplus^{\perp} V_j$$

et l'on cherche une base de W_j (appelée base d'ondelettes). Pour cela, on cherche une fonction (appelée ondelette) Ψ, telle que $\Psi(t-n)_{n\in\mathbb{Z}}$ soit une base orthonormée de W_0. On aura alors :

$$\frac{1}{\sqrt{2}}\Psi(\frac{t}{2}) = \sum_n g[n]\Phi(t-n) \quad (2)$$

Mallat [66] a montré qu'il est possible de construire une telle ondelette Ψ et qu'il existe alors une base orthonormée d'ondelettes de W_j :

$$\Psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\Psi(\frac{t-2^j n}{2^j}).$$

Ainsi, on dispose d'une base orthonormée de V_0 pour tout $J \geq 0$ en considérant :

$$(\Phi_{J,n}(t), \Psi_{j,n}(t))_{0<j\leq J, n\in\mathbb{Z}}$$

Dans la suite, le terme profondeur de décomposition sera employé pour désigner J, et j sera associé au terme d'échelle de décomposition. Les liens entre h et g vont être détaillés dans la section suivante.

2.4 Ondelettes et bancs de filtres

Dans cette section, nous allons introduire la décomposition en ondelettes continue puis la décomposition en ondelettes discrètes. Considérons alors un signal $f \in V_0$. On appelle coefficients d'ondelettes, à la profondeur J, les coefficients de la décomposition de f sur la base :

$$(\Phi_{J,n}(t), \Psi_{j,n}(t))_{0<j\leq J, n\in\mathbb{Z}}.$$

où Φ et Ψ correspondent respectivement à la fonction échelle et à l'ondelette introduite à la section précédente. Les coefficients associés à la décomposition de f sur $\Phi_{J,n}$ sont notés $c_{J,n}$, et sont appelés coefficients d'approximation à la profondeur J ; $d_{j,n}$ sont les coefficients associés à décomposition sur $\Psi_{j,n}$, et sont appelés coefficients des détails à l'échelle j. Ces coefficients vérifient les relations de récurrence selon les échelles (en utilisant (1)) :

$$c_{J,n} =< f(t), \Phi_{J,n}(t) >=< f(t), \frac{1}{\sqrt{2^J}}\Phi(\frac{t}{2^J} - n) >$$

$$= \sum_k h[k] < f(t), \frac{1}{\sqrt{2^{J-1}}}\Phi(\frac{t-2^J n}{2^{J-1}} - k) >= \sum_k h[k-2n] < f(t), \Phi_{J-1,k}(t) >= \sum_k h[k-2n] c_{J-1,k}$$

où h est le filtre introduit dans l'équation de dilatation de la section précédente. De la même manière (en utilisant (2)) :

$$d_{j,n} =< f(t), \Psi_{j,n}(t) >= \sum_k g[k-2n] c_{j-1,k}$$

Lorsque les filtres h et g sont des filtres parfaits orthogonaux dont nous donnerons une définition au chapitre suivant, on dispose de la formule de reconstruction :

$$c_{j,k} = \sum_n c_{j+1,n} h[k-2n] + \sum_n d_{j+1,n} g[k-2n]$$

L'hypothèse d'appartenance de f à V_0 induit des erreurs d'approximation lorsque le signal analysé n'appartient pas à cet espace. Trouver d'autres espaces de projection mieux adaptés a suscité beaucoup d'intérêt ([1],[108],[109], [90],[106],[17]). Une façon différente, plus cohérente avec l'étude de signaux discrets est de considérer la suite $f[n]$ et non la fonction $f(t)$. On peut alors, à partir des fonctions échelles des filtres h et g construire des bases orthonormées de $l_2(\mathbb{Z})$, sous la forme [85] :

$$\left\{h_J[k-2^J n], g_j[k-2^J n]\right\}_{n\in\mathbb{Z},\quad 0<j\leq J}$$

est une base orthonormée de $l_2(\mathbb{Z})$, le mode de géneration de tels filtres ainsi que la signification de j et J seront détaillés au chapitre suivant.
On peut alors associer à chacun de ces filtres les sous-espaces de $l_2(\mathbb{Z})$ suivant :

$$V_J = vect\{h_J[k-2^J n]_{n\in\mathbb{Z}}\}$$

et

$$W_j = vect\{g_j[k-2^j n])_{n\in\mathbb{Z}}\}.$$

Comme précédemment, on parlera de l'espace de détails W_j à l'échelle j et de la décomposition à la profondeur J. Nous avons alors la décomposition de $l_2(\mathbb{Z})$, à la profondeur J :

$$l_2(\mathbb{Z}) = V_J \bigoplus_{j=1}^{j=J} W_j$$

Il est alors possible de construire une base orthonormée de $l_2(\mathbb{Z}^2)$ de la forme :

$$B = (h_J[n-2^J l]h_J[m-2^J k], h_j[n-2^j l]g_j[m-2^j k], g_j[n-2^j l]h_j[m-2^j k], g_j[n-2^j l]g_j[m-2^j k])_{(l,k)\in\mathbb{Z}^2}$$

Les détails de construction d'une telle base seront précisés au chapitre suivant ; la décomposition sur de telles bases sera appelée décomposition en ondelettes discrètes orthonormées. On dispose donc d'une décomposition de tout signal de $l_2(\mathbb{Z}^2)$, en écrivant, cette fois :

$$l_2(\mathbb{Z}^2) = V_0 = V_J \bigoplus_{j=1}^{j=J} W_j^1 \oplus W_j^2 \oplus W_j^3$$

où V_J est engendré par $h_J[n-2^J l]h_J[m-2^J k]$, W_j^1 est engendré par $h_j[n-2^j l]g_j[m-2^j k]$, W_j^2 par $g_j[n-2^j l]h_j[m-2^j k]$ et W_j^3 par $g_j[n-2^j l]g_j[m-2^j k]$.

2.5 Transformée dyadique de Mallat

Nous introduisons ici la transformée dyadique de Mallat, en une dimension. Soit $f \in L^2(\mathbb{R})$, on définit la transformée dyadique en ondelettes de f en u à l'échelle 2^j par :

$$Wf(u, 2^j) = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{2^j}}\Psi(\frac{t-u}{2^j})dt =< f(t), \frac{1}{\sqrt{2^j}}\Psi(\frac{t-u}{2^j}) >$$

où Ψ est une ondelette.

2.6 Décompositions en ondelettes discrètes orthonormées et en paquets d'ondelettes discrètes des images

Dans cette partie, nous allons voir comment effectuer des décompositions en ondelettes discrètes orthonormées et en paquets d'ondelettes discrètes d'images. L'approche par ondelettes discrètes nous semble plus pertinente pour le traitement d'images car, il n'y a pas le problème d'initialisation de l'algorithme par projection sur un espace d'approximation.

2.6.1 Décomposition d'une image sur une base orthonormée de filtres

Considérons une image de taille $N \times N$, avec $N = 2^J$. Soit $f[n,m]$ la valeur du niveau de gris de l'image au pixel $[n,m]$. Un prolongement périodique de l'image permet d'obtenir facilement la périodicité des coefficients issus de la décomposition de f dans la base B.
En effet, d'après la périodicité de f, on a :

$$\begin{aligned} c_J[l,k] &=< f[n,m], h_J[n-2^J l]h_J[m-2^J k] >=< f[n+N,m], h_J[n-2^J l]h_J[m-2^J k] > \\ &=< f[n,m], h_J[n-2^J(l+1)]h_J[m-2^J k] >= c_J[l+1,k] \end{aligned}$$

Le coefficient associé à la plus grande échelle est 1 périodique, dans les deux directions. De façon générale, les coefficients de détails associés à l'espace W_j^1 vérifient :

$$\begin{aligned} d_{1,j}[l,k] &=< f[n,m], h_j[n-2^j l]g_j[m-2^j k] >=< f[n+N,m], h_j[n-2^j l]g_j[m-2^j k] > \\ &=< f[n,m], h_j[n-2^j(l+2^{J-j})]g_j[m-2^J k] >= d_{1,j}[l+2^{J-j},k] \end{aligned}$$

donc, le coefficient considéré est 2^{J-j} périodique dans les deux directions. Une telle étude peut être menée pour chacun des autres coefficients de détails ce qui nous autorise à écrire la décomposition sous la forme d'une image de taille $N \times N$ du type de celle présentée en Figure 2.1. La décroissance du nombre de coefficients en fonction de l'échelle permet la conservation du volume des données [64]. Le principal désavantage de ce type de décomposition étant la création d'artefacts aux bords de l'image qu'entraine la périodisation de f.

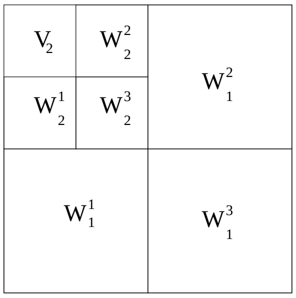

FIG. 2.1 – Coefficients de la décomposition sur banc de filtre avec $J = 2$

2.6.2 Décomposition sur des filtres associés à des paquets d'ondelettes

Les paquets d'ondelettes sont définis à partir des équations de dilatation suivante :

$$W_{2n}(x) = \sqrt{2}\sum_k h[k]W_n(2x-k)$$

et

$$W_{2n+1}(x) = \sqrt{2}\sum_k g[k]W_n(2x-k)$$

où les filtres h et g sont ceux introduits précédemment, et où $\{W_0(-n)\}_{n\in\mathbb{Z}}$ est une base orthonormée d'un espace d'approximation V_0. Si :

$$\frac{1}{\sqrt{2^p}}W_n(\frac{x}{2^p}-k)$$

est une base orthonormée de W_n^p, en considérant les fonctions :

$$\frac{1}{\sqrt{2^{p+1}}}W_{2n}(\frac{x}{2^{p+1}}-k)$$

et

$$\frac{1}{\sqrt{2^{p+1}}}W_{2n+1}(\frac{x}{2^{p+1}}-k)$$

qui engendrent respectivement les espaces W_{2n}^{p+1} et W_{2n+1}^{p+1}, on a alors :

$$W_n^p = W_{2n}^{p+1} \oplus^{\perp} W_{2n+1}^{p+1}$$

On peut donc associer aux paquets d'ondelettes un arbre binaire, dont la racine est l'espace d'approximation $V_0 = W_0^0$. Pour construire des bases orthonormées à partir de cette structure d'arbre, on introduit la notion d'arbre admissible. On appelle arbre admissible, un arbre issu de l'arbre binaire de la décomposition en ondelettes, dans lequel chaque feuille a 0 ou 2 fils dans l'arbre binaire. Les espaces correspondants à chaque feuille d'un arbre admissible sont orthonormés et leur somme vaut V_0. On donne un exemple d'arbre binaire en Figure 2.2. Les espaces pointés représentent un choix possible pour de base orthonormée de V_0.

En dimension 2, la construction est analogue , de la forme :

$$W_{2n,2p}(x,y) = 2\sum_{k,l} h[k]h[l]W_{n,p}(2x-k,2y-l)$$

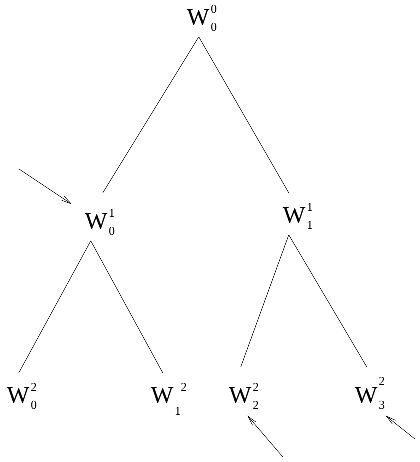

FIG. 2.2 – arbre binaire associé à la décomposition en ondelettes à la profondeur 2

$$W_{2n+1,2p}(x,y) = 2\sum_{k,l} g[k]h[l]W_{n,p}(2x-k, 2y-l)$$

$$W_{2n,2p+1}(x,y) = 2\sum_{k,l} h[k]g[l]W_{n,p}(2x-k, 2y-l)$$

$$W_{2n+1,2p+1}(x,y) = 2\sum_{k,l} g[k]g[l]W_{n,p}(2x-k, 2y-l)$$

Si l'on suppose maintenant que :

$$\frac{1}{2^j}W_{n,p}(\frac{x}{2^j}-k, \frac{y}{2^j}-l)$$

est une base orthonormée de $W_{n,p}^j$, on obtient :

$$W_{2n,2p}^{j+1} \oplus^{\perp} W_{2n,2p+1}^{j+1} \oplus^{\perp} W_{2n+1,2p}^{j+1} \oplus^{\perp} W_{2n+1,2p+1}^{j+1} = W_{n,p}^j$$

Cette décomposition donne naturellement naissance à un quadarbre ("quad tree"), dans lequel chaque noeud à 0 ou 4 fils. On donne en Figure 2.3, un exemple d'arbre associé à une décomposition en paquets d'ondelettes 2D ainsi que la représentation fréquentielle de la décomposition retenue.

Cette approche présente l'inconvénient majeur de créer des artefacts aux bords des images du fait qu'implicitement l'image est considérée périodique.

2.7 Application à la classification supervisée de textures

Dans cette partie, nous présentons différentes approches utilisant les ondelettes et les paquets d'ondelettes pour la classification supervisée des textures. Ces méthodes utilisent des décompositions en ondelettes et paquets d'ondelettes du type de celles présentées plus haut malgré les artefacts qu'elles introduisent aux bords des images. On envisage que les images à traiter sont des images de textures du type de celles présentées en Figure 2.4. Pour des tâches de classification texturale, les paquets d'ondelettes ont été utilisés par Chang [16] et Laine [58]. Dans [58], la stratégie adoptée est de décomposer des images d'une bibliothèque de textures en sous-images

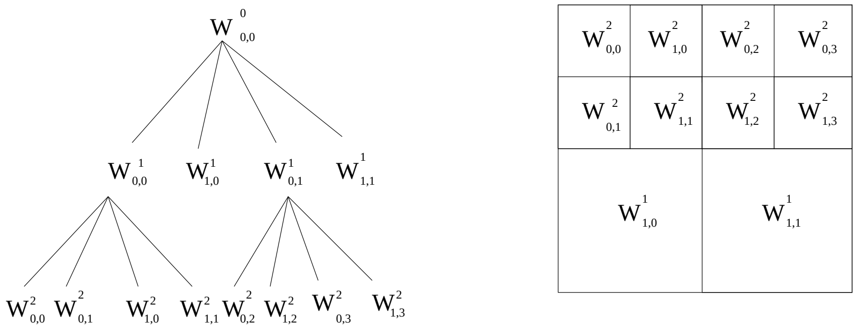

FIG. 2.3 – quadarbre associé à une décomposition particulière en paquets d'ondelettes et la décomposition fréquentielle associée

puis de décomposer complètement chacune d'entre elles en paquets d'ondelettes, jusqu'à une profondeur donnée, et de calculer en chacun des noeuds du quadarbre ainsi obtenu les énergies et les entropies associées aux espaces correspondants, selon :

$$\text{Energie : } \sum_{n,m} W[n,m]^2,$$

$$\text{Entropie : } -\sum_{n,m} W[n,m]^2 \ln(W[n,m]^2)$$

où $W[n,m]$ est un coefficient de la décomposition, et les sommes portent sur le sous-espace correspondant. Dans [58], les énergies et les entropies calculées au niveau des noeuds du quadarbre sont les caractéristiques de textures. Etant donné leur nombre, une sélection doit s'opérer ; dans [58], celle-ci se fait par le biais de simulations numériques. Le nombre de caractéristiques à retenir n'est pas discuté ; cependant, il ressort de cette étude que la précision des résultats n'est pas liée au nombre des caractéristiques. Une comparaison est faite avec les énergies et les entropies calculées sur des espaces associés à une décomposition en ondelettes orthonormées.
Par ailleurs, la distance euclidienne est utilisée pour tester de quelle texture de référence chaque sous-image de texture est la plus proche, ce qui ne prend pas en compte les ordres de grandeur des énergies et des entropies dans les différents sous-espaces. En effet, les coefficients ont tendance à être plus importants aux grandes échelles qu'aux petites et ce problème n'est pas discuté.

Dans [16], l'approche est un peu différente, dans la mesure où la décomposition en paquets d'ondelettes n'est pas considérée dans son intégralité. Partant de l'espace W_0^0, la décomposition à l'échelle supérieure ne s'effectue que si on a la propriété suivante :

$$\frac{1}{NM}\sum_{n,m} |W(n,m)| < Ce_{max}$$

où la somme porte sur un des noeuds du quadarbre, correspondant à un sous-espace particulier, et e_{max} est la valeur maximale, à l'échelle considérée, de la déviation moyenne des coefficients.
Grâce à cette technique, un choix d'espaces pertinents peut-être déterminé, et l'on dispose d'une structure d'arbre associée à ce choix. La structure finale de l'arbre obtenu dépend beaucoup du paramètre C. Les caractéristiques sont alors déterminées en considérant les regroupements d'espaces pour lesquels la déviation moyenne est la plus forte. Les regroupements réalisés ne peuvent contenir deux espaces associés à une même échelle. Il apparaît que les regroupements qui apparaissent le plus souvent en effectuant les calculs sur des sous-images d'une même texture

sont bien caractéristiques de la texture. Par ailleurs, les champs correspondant à des espaces proches de la racine de l'arbre sont plus stables en termes de caractéristiques.
La stabilité énergétique des sous-espaces n'est pas uniquement liée à la texture elle-même mais aussi au fait que les estimations se font sur des sous-images de plus en plus petites au fur et à mesure que l'on décompose l'image. Il est intéressant de remarquer que les champs les plus discriminants (ce qui sont retenus comme caractéristiques de la texture) sont associés aux déviations moyennes les plus fortes , ce qui n'est pas toujours judicieux. Dans le cas ou l'on doit caractériser des textures dont les énergies sont faibles, d'autres techniques ont été mises en place [78]. Cependant, pour les problèmes complexes d'identification texturale , les critères énergétiques prouvent leur efficacité.

Une fois que l'on a choisi les caractéristiques de la texture, se pose le problème du choix de la distance pour la comparaison des caractéristiques estimées sur des sous-images de texture avec celles estimées sur les textures de référence. Ensuite, dans [16], l'affectation s'effectue en considèrant d'abord l'espace correspondant aux énergies les plus fortes, pour la texture que l'on cherche à affecter, et regarde parmi les textures de référence celles pour lesquelles l'espace correspondant aux énergies les plus fortes est le même, les autres textures de référence étant éliminées. On réitère ensuite le procédé sur les espaces de plus faible énergie. Après un certain nombre d'itérations, la sous-image de texture est affectée à la texture de référence la plus proche au sens de la distance choisie. L'approche par ondelettes orthogonales et paquets d'ondelettes présentent deux inconvénients, l'un est lié au problème de bords dont nous avons déjà parlé et l'autre est dû à la préservation du volume des données, qui empêche une estimation correcte des caractéristiques aux grandes échelles.

2.8 Application à la segmentation non supervisée des textures

Dans cette partie nous présentons brièvement les méthodes de segmentation non supervisée utilisant des approches par ondelettes. Ces approches sont telles que les deux principaux inconvénients des approches par projection sur des bases orthonormées sont éliminés. Nous insistons alors sur l'objectif de notre travail et sur ses aboutissements.

La segmentation d'images texturées nécessite la construction de modèles plus précis que ceux utilisés en classification texturale, du fait des problèmes de détection de frontières entre textures.

Le choix d'utiliser une décomposition redondante associée à des frames d'ondelettes permet d'avoir une représentation spatiale plus précise des fréquences lorsque l'échelle de décomposition augmente. Par ailleurs, nous privilégions la parité des frames d'ondelettes utilisés en raison du traitement très rapide des problèmes de bords des images (par extension miroir). Ainsi, n'apparaissent pas aux bords des images des artefacts comme c'est le cas avec les décompositions orthogonales, ce point sera traité en détail au chapitre suivant.

Nous rejoignons, par les choix faits sur les filtres, l'étude faite par [58], mais nous nous distinguons du choix fait par Unser [95]. Le choix des frames d'ondelettes ainsi que leur définition sont abordés au chapitre suivant, dans lequel nous rappelons des théorèmes classiques sur les filtres associés au frame. Après avoir choisi un frame bien adapté à notre problème, nous nous attaquerons au traitement des coefficients issus de la décomposition pour effectuer une segmentation d'image texturée. Nous utiliserons toujours l'aspect énergétique pour effectuer une segmentation, ainsi que des analyses en composantes principales. Nous nous distinguerons des choix faits dans [95], et les conclusions concernant les espaces les plus discriminants seront différentes de celles de [16]. Dans un dernier chapitre, nous nous intéresserons à la segmentation des textures par l'étude de leurs bords et de leur régularité Lipschitzienne, que nous définirons alors.

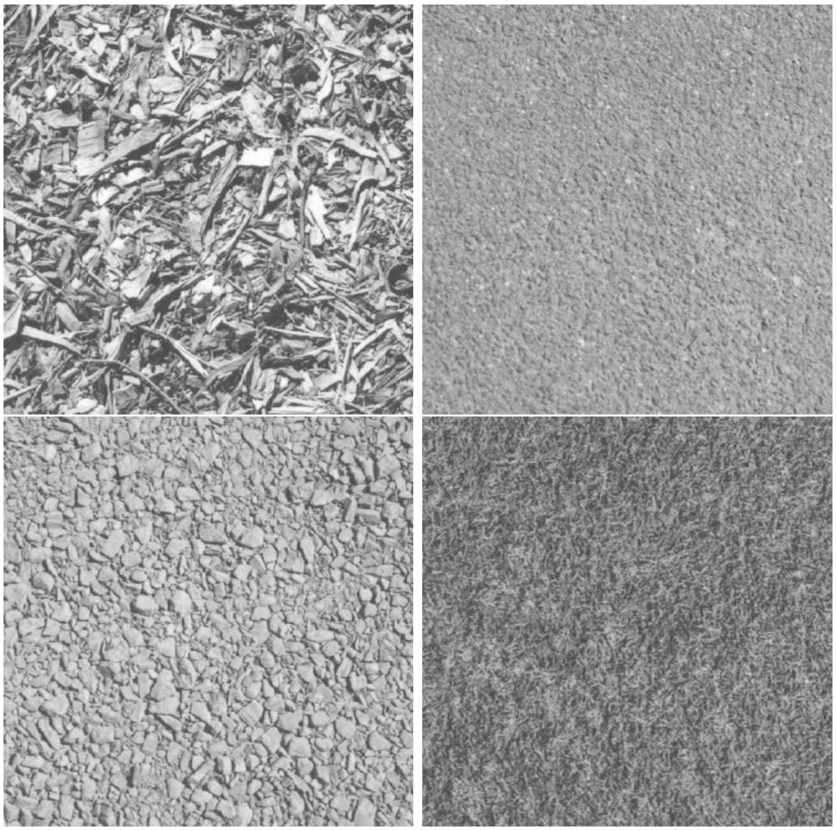

FIG. 2.4 – Textures issues de l'album Vitex-database [101], référencées respectivement comme Misc, Asphalt, Rock et Grass

3

Analyse de signaux et d'images par bancs de filtres

Dans ce chapitre, nous allons considérer la décomposition de signaux et d'images sur des bancs de filtres associés à des bases orthonormées d'ondelettes ou des frames d'ondelettes. L'approche concernant les bases orthonormées de filtres est un rappel de théorèmes existants. Nous nous plaçons directement dans le cadre de la décomposition d'un signal f discret appartenant à $l_2(\mathbb{Z})$ sur une base de filtres du type de ceux présentés au chapitre 2, i.e. composée de filtres g et h correspondants à la décomposition du signal f sur des espaces de détails W_j et d'approximation V_J. Ce schéma nous permettra d'introduire facilement les frames de $l_2(\mathbb{Z})$.

3.1 Filtres parfaits

Considérons les formules de décomposition d'un signal f obtenues par convolution avec les filtres h et g puis décimation :

$$c_1[n] = \sum_k f[k]h[k-2n] = f * \bar{h}[2n]$$

$$d_1[n] = \sum_k f[k]g[k-2n] = f * \bar{g}[2n]$$

où $\bar{h}[n] = h[-n]$. On cherche ensuite à reconstruire le signal f à l'aide de c_1 et d_1. Il a été montré, que sous certaines conditions sur h et g et, en utilisant des vecteurs du type :

$$\check{x}[n] = \begin{cases} x[p] \text{ si } n = 2p \\ 0 \text{ sinon} \end{cases}$$

il existe des filtres $\tilde{h}$ et $\tilde{g}$ tels que :

$$f[n] = \check{c}_1 * \tilde{h}[n] + \check{d}_1 * \tilde{g}[n]$$

Dans ce cas, on dit que l'on a une reconstruction exacte et que les couples de filtres (h, g) et $(\tilde{h}, \tilde{g})$ sont parfaits.

3.2 Conditions nécessaires et suffisantes d'obtention de filtres parfaits

La série de Fourier d'une séquence $x[n]$ est :

$$\hat{x}(\omega) = \sum_n x[n]e^{-in\omega}$$

Considérons le signal sous-échantillonné $y[n] = x[2n]$, alors :

$$\hat{y}(2\omega) = \sum_n x[2n]e^{-in2\omega} = \frac{1}{2}(\hat{x}(\omega) + \hat{x}(\omega + \pi))$$

Si on considère un signal sur-échantillonné du type $\check{x}$, alors :

$$\hat{y}(\omega) = \sum_n x[n]e^{-i2n\omega} = \hat{x}(2\omega)$$

On a alors le théorème suivant :

Théorème 1 (Vetterli [100]) *Les filtres h,g,$\tilde{h}$,$\tilde{g}$ réalisent une reconstruction exacte, si et seulement si :*

$$\hat{h}^*(\omega + \pi)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega + \pi)\hat{\tilde{g}}(\omega) = 0$$

$$\hat{h}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega)\hat{\tilde{g}}(\omega) = 2$$

Avec $\hat{g}^*(\omega) = \hat{g}(-\omega)$

Donnons alors quelques exemples classiques de filtres parfaits.

3.3 Exemples de filtres parfaits

3.3.1 Filtres miroirs en quadrature

h et g sont des filtres miroirs en quadrature si, par définition :

$$\hat{g}(\omega) = \hat{h}(\omega + \pi)$$

Ainsi, par exemple, $(\tilde{h}, \tilde{g})$ définis par :

$$\hat{\tilde{h}}(\omega) = \hat{h}(\omega)$$

$$\hat{\tilde{g}}(\omega) = -\hat{g}(\omega)$$

permettent une reconstruction exacte. Dans le cas où $\hat{h}$ est pair, le filtre $\hat{g}$ est le symétrique de $\hat{h}$ par rapport à $\frac{\pi}{2}$. L'un des principaux intérêts de ces filtres est que la parité de $\hat{h}$ entraine celle de $\hat{g}$. En revanche, il n'est pas possible d'obtenir des filtres miroirs en quadrature de longueur finie [24].
Dans ce cas, la seconde condition du théorème de Vetterli se réécrit :

$$|\hat{h}(\omega)|^2 - |\hat{h}(\omega + \pi)|^2 = 2$$

3.3.2 Filtres conjugués en quadrature

La différence majeure avec les filtres miroir en quadrature est que le filtre g est obtenu par conjugaison puis décalage, de la façon suivante :

$$\hat{g}(\omega) = e^{-i\omega}\hat{h}(-\omega + \pi).$$

Ainsi, par exemple, $(\tilde{h}, \tilde{g})$ définis par :

$$\hat{\tilde{h}}(\omega) = \hat{h}(-\omega)$$

$$\hat{\tilde{g}}(\omega) = \hat{g}(-\omega)$$

permettent une reconstruction exacte. Dans ce cas, la seconde condition du théorème de Vetterli devient :

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2$$

3.3.3 Filtres parfaits orthogonaux

Au lieu d'introduire les filtres grâce à une relation entre h et g, on cherche les filtres parfaits vérifiant $\tilde{h} = h$ et $\tilde{g} = g$. Nous verrons, ensuite, comment ce type de filtres peut-être utilisé pour construire des bases orthonormées. Les conditions pour avoir une reconstruction parfaite sont ici :

$$\hat{h}^*(\omega + \pi)\hat{h}(\omega) + \hat{g}^*(\omega + \pi)\hat{g}(\omega) = 0$$

$$\hat{h}^*(\omega)\hat{h}(\omega) + \hat{g}^*(\omega)\hat{g}(\omega) = 2$$

On peut réécrire cela sous forme matricielle :

$$\begin{pmatrix} \hat{h}(\omega) & \hat{g}(\omega) \\ \hat{h}(\omega + \pi) & \hat{g}(\omega + \pi) \end{pmatrix} \times \begin{pmatrix} \hat{h}^*(\omega) \\ \hat{g}^*(\omega) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

d'où l'on déduit, lorsque $\Delta(\omega) = \hat{h}(\omega)\hat{g}(\omega + \pi) - \hat{h}(\omega + \pi)\hat{g}(\omega)$ ne s'annule pas sur $[-\pi; \pi]$, que :

$$\begin{pmatrix} \hat{h}^*(\omega) \\ \hat{g}^*(\omega) \end{pmatrix} = \frac{2}{\Delta(\omega)} \begin{pmatrix} \hat{g}(\omega + \pi) \\ -\hat{h}(\omega + \pi) \end{pmatrix}$$

comme $\Delta(\omega + \pi) = -\Delta(\omega)$, dans le cas de filtres parfaits orthogonaux, les conditions de reconstruction parfaite s'écrivent :

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2$$

et, il existe $a \in \mathbb{R}$ et $l \in \mathbb{Z}$ tels que :

$$\hat{g}(\omega) = ae^{-i(2l+1)\omega}\hat{h}^*(\omega + \pi)$$

Une démonstration plus complète est fournie dans [66], la détermination de la forme de $\hat{g}$ est fondée sur l'étude de $\Delta(\omega)$. On voit alors que les filtres conjugués introduits dans la section précédente sont des cas particuliers avec $a = 1$ et $l = 0$, conduisant au filtre g :

$$g[n] = (-1)^{1-n}h[1 - n]$$

3.4 Base orthonormée de filtres

Nous allons étudier ici la construction de bases orthonormées de filtres, ce qui nous permettra de présenter les frames dans la section suivante. On considère un couple (h, g) de filtres parfaits orthogonaux. Pour plus de simplicité prenons $a = 1$ et $l = 0$, mais les calculs peuvent être conduits dans le cadre le plus général. On a alors les théorèmes suivants :

Théorème 2 (Rioul [85]) *Si les filtres h et g sont parfaits et orthogonaux, alors :*

$$\Big\{h[n-2l], g[n-2l]\Big\}_{l\in\mathbb{Z}}$$

forme une base orthonormée de $l_2(\mathbb{Z})$.

On peut alors générer des bases orthonormées de filtres par le théorème suivant :

Théorème 3 (Rioul [85]) *Si les filtres h et g sont parfaits et orthogonaux, alors :*

$$\Big\{h_J[n-2^J l], g_j[n-2^j l]_{1\leq j\leq J}\Big\}_{l\in\mathbb{Z}}$$

est une base orthonormée de $l_2(\mathbb{Z})$, en posant $h_0 = 1$ et :

$$\forall i \geq 0 \quad \begin{cases} h_{i+1}[k] &= h_{\uparrow 2^i} * h_i[k] &= \sum_n h[n]h_i[k-2^i n] \\ g_{i+1}[k] &= g_{\uparrow 2^i} * h_i[k] &= \sum_n g[n]h_i[k-2^i n] \end{cases}$$

Où $y = x_{\uparrow 2^i}$ désigne :

$$y[n] \quad = \quad \begin{cases} x[k] \ si \ n = 2^i k \\ 0 \ sinon \end{cases}$$

Pour la démonstration, voir annexe A.1

3.5 Frame orthogonal de $l_2(\mathbb{Z})$

On appelle frame de $l_2(\mathbb{Z})$, toute famille de suites $c_n[p] \in l_2(\mathbb{Z})$, telle que :

$$\forall f \in l_2(\mathbb{Z}),$$
$$A||f||^2 \leq \sum_n (\sum_p f[p]c_n[p])^2 \leq B||f||^2$$

Si $A = B$, le frame est dit étroit ; si $A = B = 1$, il est appelé frame orthogonal. Dans le cas où $c_n[p]$ est de norme 1 pour tout n, et si le frame est orthogonal alors c'est une base orthonormée. Les frames orthogonaux sont donc caractérisés par la propriété de conservation de carré de norme suivante :

$$\sum_n < f, c_n >^2 = ||f||^2$$

La construction de frames orthogonaux de $l_2(\mathbb{Z})$ à partir de filtres parfaits est due à Rioul [85]. Nous montrons, alors, l'équivalence suivante, qui autorise la construction de frames orthogonaux à partir de filtres qui ne sont pas parfaits :

Théorème 4 *La famille :*

$$\Big\{(\frac{1}{a})^{\frac{J}{2}} h_J[n-l], (\frac{1}{a})^{\frac{j}{2}} g_j[n-l]_{1\leq j\leq J}\Big\}_{l\in\mathbb{Z}}$$

est un frame orthogonal de $l_2(\mathbb{Z})$ si et seulement si

$$|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = a$$

Ces propriétés sont connus sous le nom de théorie de Littlewood-Paley. Pour plus de détails voir [47]. Nous proposons une démonstration en annexe A.2.

3.5.1 Frame orthogonal pair de longueur finie

La construction d'un frame orthogonal pair nécessite l'utilisation de filtre h et g pairs. La parité de h et g, que l'on suppose à coefficients réels, entraine que les filtres $\hat{h}$ et $\hat{g}$ sont réels pairs, ce qui impose sur les filtres la condition :

$$\hat{h}(\omega)^2 + \hat{g}(\omega)^2 = a,$$

Si maintenant on suppose que h et g sont de longueur finie, alors $\hat{h}$ et $\hat{g}$ sont des polynômes trigonométriques P et Q de degré respectif N et M, qui satisfont :

$$P^2(X) + Q^2(X) = a$$

Or, il n'existe pas deux polynômes de degré strictement positif vérifiant l'égalité souhaitée. Donc, il n'est pas possible d'obtenir un frame orthogonal et pair de longueur finie. En revanche, si l'on considère des filtres de longueur infinie, cette condition peut être remplie par des filtres pairs.
Exemple : les filtres associés aux ondelettes polynomiales splines, introduites par Battle et Lemarié, sont définis de la manière suivante :

$$\hat{h}(\omega) = \exp(\frac{-i\epsilon\omega}{2})\sqrt{\frac{S_{2m+2}(\omega)}{2^{2m+1}S_{2m+2}(2\omega)}},$$

avec :

$$S_n(\omega) = \sum_{-\infty}^{\infty} \frac{1}{(\omega + 2k\pi)^n}$$

et $\epsilon = 0$, si m impair, et 1, sinon. Pour m impair, le filtre $h[n]$ obtenu est pair. De plus :

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2$$

En considérant alors $\hat{g}(\omega) = \hat{h}(\omega + \pi)$, on obtient, si m est impair, la parité de $\hat{g}$.

3.5.2 Frame orthogonal non symétrique de longueur finie

D'après le théorème 4, nous nous intéressons à la construction de filtres vérifiant $|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = 1$, sans perte de généralités. La construction de tels filtres a été effectuée par I. Daubechies [24]. Elle se place ici dans le cas où les filtres sont des filtres parfaits orthogonaux. Le filtre $\hat{h}$ vérifie :

$$\hat{h}(\omega)^2 = (\cos^2(\frac{\omega}{2}))^N P(\sin^2(\frac{\omega}{2}))$$

$\hat{h}$ devant, par ailleurs, vérifier $\hat{h}(\omega)^2 + \hat{h}(\omega + \pi)^2 = 1$, le polynôme P vérifie alors :

$$y^N P(1-y) + (1-y)^N P(y) = 1$$

D'après le théorème de Bezout, il existe deux uniques polynômes q_1 et q_2 de degré inférieur ou égal à $N-1$, tels que :

$$(1-y)^N q_1(y) + y^N q_2(y) = 1$$

En remplaçant y par $1-y$, on obtient que $q_1(y) = q_2(1-y)$ et donc que $P(y) = q_1(y)$, dont on peut avoir une expression exacte. En effet, faisons un développement de Taylor à l'ordre N de q_1, il vient :

$$q_1(y) = (1-y)^{-N}(1 - y^N q_1(1-y))$$

$$= \sum_{k=0}^{N-1} C_{N+k-1}^{k} y^k + O(y^N)$$

Comme q_1 est de degré inférieur ou égal à $N-1$, il vient :

$$q_1(y) = \sum_{k=0}^{N-1} C_{N+k-1}^{k} y^k$$

Ainsi :

$$|\hat{h}(\omega)|^2 = \cos^{2N}(\frac{\omega}{2}) \sum_{k=0}^{N-1} C_{N+k-1}^{k} \sin^{2k}(\frac{\omega}{2})$$

On dispose alors du théorème suivant qui permet de trouver un filtre h convenable :

Théorème 5 (Fejer) *Soit A un polynôme trigonométrique positif invariant par passage de ω à $-\omega$, A est de la forme :*

$$A(\omega) = \sum_{m=0}^{M} a[m]\cos(m\omega) \ \textit{avec}\ a[m] \in \mathbb{R}$$

Alors, il existe un polynôme trigonométrique d'ordre M :

$$B(\omega) = \sum_{m=0}^{M} b[m]e^{im\omega} \ \textit{avec}\ b[m] \in \mathbb{R}$$

tel que $|B(\omega)|^2 = A(\omega)$

Un exemple simple de filtre h pour $N = 1$ est donné par :

$$\hat{h}(\omega) = e^{i\frac{\omega}{2}}\cos(\frac{\omega}{2})$$

et on peut alors choisir, pour le filtre g :

$$\hat{g}(\omega) = -ie^{i\frac{\omega}{2}}\sin(\frac{\omega}{2})$$

on obtient donc, comme coefficients pour les filtres h et g :

$$h_0 = 0.5, h_1 = 0.5$$

et

$$g_0 = 0.5, g_1 = -0.5.$$

Bien évidemment, ces filtres ne sont pas pairs.

3.6 Frame orthogonal en 2D

A partir du frame orthogonal obtenu en 1D, il est facile de construire un frame orthogonal de $l_2(\mathbb{Z}^2)$. Nous montrons alors l'équivalence suivante permettant de le définir à partir de filtres h et g :

Théorème 6 *La famille :*

$$\Big\{(\frac{1}{a})^J h_J[n-l]h_J[m-k],$$
$$\{(\frac{1}{a})^j g_j[n-l]h_j[m-k], (\frac{1}{a})^j h_j[n-l]g_j[m-k], (\frac{1}{a})^j g_j[n-l]g_j[m-k]\}_{1\le j\le J}\Big\}_{(l,k)\in\mathbb{Z}^2}$$

est un frame orthogonal de $l_2(\mathbb{Z}^2)$, si et seulement si $|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = a$.

Pour une démonstation voir l'annexe A.3

3.6.1 Calcul des coefficients de la décomposition et formule de reconstruction

Nous allons voir que, dans le cas où les filtres h et g sont pairs, la formule de décomposition est une convolution simple. Nous nous plaçons dans le cas $a = 1$, pour plus de simplicité, tous les calculs pouvant être repris pour tout a positif. On a donc :

$$s_J[l,m] =< x[k,p], h_J[k-l]h_J[p-m] >$$
$$= \sum_n h_{\uparrow 2^{J-1}}[n] \sum_q h_{\uparrow 2^{J-1}}[q] \sum_{k,p} x[k,p] h_{J-1}[k-l-n] h_{J-1}[p-m-q]$$

Dans le cas où le filtre h est pair, on obtient :

$$s_J[l,m] =< x[k,p], h_J[k-l]h_J[p-m] >$$
$$= \sum_n h_{\uparrow 2^{J-1}}[-n] \sum_q h_{\uparrow 2^{J-1}}[-q] \sum_{k,p} x[k,p] h_{J-1}[k-l-n] h_{J-1}[p-m-q]$$
$$= \sum_n h_{\uparrow 2^{J-1}}[n] \sum_q h_{\uparrow 2^{J-1}}[q] \sum_{k,p} x[k,p] h_{J-1}[k-l+n] h_{J-1}[p-m+q]$$
$$= h_{\uparrow 2^{J-1}}[l] * h_{\uparrow 2^{J-1}}[m] * s_{J-1}[l,m],$$

où les variables l (respectivement m) dans les filtres indiquent que la convolution porte sur la première (respectivement la seconde) variable. De la même manière, on obtient, en supposant g pair que les autres coefficients vérifient :

$$d_{1,j}[l,m] =< x[k,p], h_j[k-l]g_j[p-m] >= h_{\uparrow 2^{j-1}}[l] * g_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$
$$d_{2,j}[l,m] =< x[k,p], g_j[k-l]h_j[p-m] >= g_{\uparrow 2^{j-1}}[l] * h_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$
$$d_{3,j}[l,m] =< x[k,p], g_j[k-l]g_j[p-m] >= g_{\uparrow 2^{j-1}}[l] * g_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$

En pratique, les calculs des produits scalaires nécessitent de faire des troncatures sur les filtres, car ceux-ci sont nécessairement de longueur infinie comme nous l'avons vu précédemment.

Dans le cas général, on a la formule suivante :

$$s_J[l,m] =< x[k,p], h_J[k-l]h_J[p-m] >$$
$$= \sum_n h_{\uparrow 2^{J-1}}[n] \sum_q h_{\uparrow 2^{J-1}}[q] s_{J-1}(l+n, m+q)$$

et, de la même façon :

$$d_{1,j}[l,m] =< x[k,p], h_j[k-l]g_j[p-m] >$$
$$= \sum_n h_{\uparrow 2^{j-1}}[n] \sum_q g_{\uparrow 2^{j-1}}[q] s_{j-1}[l+n, m+q]$$
$$d_{2,j}[l,m] =< x[k,p], g_j[k-l]h_j[p-m] >$$
$$= \sum_n g_{\uparrow 2^{j-1}}[n] \sum_q h_{\uparrow 2^{j-1}}[q] s_{j-1}[l+n, m+q]$$
$$d_{3,j}[l,m] =< x[k,p], g_j[k-l]g_j[p-m] >$$
$$= \sum_n g_{\uparrow 2^{j-1}}[n] \sum_q g_{\uparrow 2^{j-1}}[q] s_{j-1}[l+n, m+q]$$

Une propriété intéressante d'un frame orthogonal est la simplicité de reconstruction des signaux à l'aide des coefficients du frame. Pour tout frame orthogonal $\{h_j\}_{j\in J}$ d'un espace de Hilbert H, on a la propriété suivante :

$$\forall x \in H \quad x = \sum_{j\in J} < x, h_j >_H h_j$$

Dans notre cas en 2D, on a donc :

$$x[k,p] = \sum_l \sum_m < x[k,p], h_J[k-l]h_J[p-m] > h_J[k-l]h_J[p-m] +$$

$$\sum_{j=1}^{J} \sum_l \sum_m < x[k,p], h_j[k-l]g_j[p-m] > h_j[k-l]g_j[p-m] +$$

$$\sum_l \sum_m < x[k,p], g_j[k-l]h_j[p-m] > g_j[k-l]h_j[p-m] +$$

$$\sum_l \sum_m < x[k,p], g_j[k-l]g_j[p-m] > g_j[k-l]g_j[p-m]$$

De plus, nous avons vu qu'il y a conservation du carré de la norme associée à $l_2(\mathbb{Z})$.

3.7 Analyse par banc de filtre et transformée dyadique de Mallat

Cette section sert à faire le lien entre l'approche continue et l'approche discrète en ce qui concerne les frames orthogonaux. Nous allons, dans un premier temps, montrer les différences entre la transformée dyadique en ondelettes en dimension un, et les frames orthogonaux de $l_2(\mathbb{Z})$. Nous étudierons ensuite les frames engendrés par la transformée dyadique en dimension deux.

3.7.1 Transformée monodimensionnelle et condition de frame orthogonal

Soit $Wf(u,2^j)$ la transformée dyadique en ondelettes introduite au chapitre précédent. Soit Ψ l'ondelette servant au calcul de la transformée. Nous avons le théorème suivant [66], qui donne des conditions pour que la famille de fonctions associée à la transformée soit un frame de $L^2(\mathbb{R})$.

Théorème 7 *S'il existe deux constantes A et B telles que*

$$\forall \omega \in \mathbb{R} - \{0\} \quad A \leq \sum_{j=-\infty}^{j=+\infty} |\hat{\Psi}(2^j\omega)|^2 \leq B$$

alors

$$A\|f\|^2 \leq \sum_{j=-\infty}^{+\infty} \frac{1}{2^j} \|Wf(u,2^j)\|^2 \leq B\|f\|^2.$$

Si, de plus, il existe une ondelette $\tilde{\Psi}$ vérifiant :

$$\forall \omega \in \mathbb{R} - \{0\}, \sum_{j=-\infty}^{j=+\infty} \hat{\Psi}^*(2^j\omega)\hat{\tilde{\Psi}}(2^j\omega) = 1$$

alors

$$f(t) = \sum_{j=-\infty}^{+\infty} \frac{1}{2^j} Wf(.,2^j) * \tilde{\Psi}_{2^j}(t)$$

Pour la démonstration voir [64]. Considérons alors une fonction échelle et une ondelette vérifiant les relations introduites au chapitre précédent, à savoir :

$$\Phi(x) = \sqrt{2}\sum_k h[k]\Phi(2x-k)$$

$$\Psi(x) = \sqrt{2}\sum_k g[k]\Phi(2x-k).$$

En supposant que $\tilde{\Psi}$ est une ondelette et $\tilde{\Phi}$ une fonction échelle et sont définies par les mêmes relations que Φ et Ψ et, en remplaçant respectivement h par $\tilde{h}$ et g par $\tilde{g}$, on a, alors, le théorème suivant :

Théorème 8

$$\forall\omega \in [-\pi,\pi], \quad \hat{\tilde{h}}(\omega)\hat{h}^*(\omega) + \hat{\tilde{g}}(\omega)\hat{g}^*(\omega) = 2$$

Alors, on a:

$$\forall\omega \in \mathbb{R} - \{0\}, \sum_{j=-\infty}^{j=\infty} \hat{\Psi}^*(2^j\omega)\hat{\tilde{\Psi}}(2^j\omega) = 1$$

Pour la démonstration voir [64]. Il résulte de ceci que, si les filtres sont tels que $\tilde{h} = h$ et $\tilde{g} = g$, alors :

$$\sum_{j=-\infty}^{j=+\infty} \frac{1}{2^j}\|Wf(u,2^j)\|^2 = \|f\|^2$$

et que :

$$f(t) = \sum_{j=-\infty}^{j=+\infty} \frac{1}{2^j} Wf(.,2^j) * \Psi_{2^j}(t)$$

Donc, le frame associé à la décomposition est orthogonal, si les filtres h et g vérifient :

$$|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = 2,$$

En revanche, la réciproque n'est pas vraie, i.e si le frame associé à la transformée dyadique est orthogonal, nous n'avons pas obligatoirement l'égalité au-dessus sur les filtres. L'équivalence est obtenue si l'on cherche à construire un frame orthogonal de V_0 à partir de la transformée dyadique, où V_0 est l'espace engendré par les translatées de Φ, introduite plus haut.

3.7.2 Transformée bidimensionnelle et condition de frame orthogonal

Dans cette partie, on regarde s'il est possible de construire un frame orthogonal en dimension deux de V_0, où cette fois V_0 est engendré par $\{\Phi(x-l)\Phi(y-k)\}_{(k,l)\in\mathbb{Z}^2}$, Φ étant une fonction échelle vérifiant une équation de dilatation du même type que celle introduite précédemment. En définissant alors Ψ par une relation de dilatation du même type que celle introduite précedemment, nous regardons à quelles conditions sur h et g la famille :

$$F = \{\frac{1}{a^{2J}}\Phi(\frac{x}{2^J})\Phi(\frac{y}{2^J}), \frac{1}{a^{2j}}\Psi(\frac{x}{2^j})\Phi(\frac{y}{2^j}), \frac{1}{a^{2j}}\Phi(\frac{x}{2^j})\Psi(\frac{y}{2^j})\}_{0<j\leq J}$$

avec $a > 0$ est un frame orthogonal de V_0.

Théorème 9 *La famille F est un frame orthogonal de V_0, si et seulement si :*

$$\forall(\omega_1,\omega_2), \quad |\hat{h}(\omega_1)\hat{h}(\omega_2)|^2 + |\hat{h}(\omega_1)\hat{g}(\omega_2)|^2 + |\hat{g}(\omega_1)\hat{h}(\omega_2)|^2 = a \quad a > 0$$

Pour la démonstration, voir annexe A.4. Il est facile de voir que, dans ce cas, il n'est pas non plus possible d'avoir $\hat{h}$ et $\hat{g}$ pairs et F frame orthogonal.

4

Analyse de textures : utilisation de frames de $l_2(\mathbb{Z}^2)$

Dans ce chapitre, nous allons regarder comment décomposer des images sur les frames de $l_2(\mathbb{Z}^2)$, introduits au chapitre précédent. Nous insisterons, particulièrement, sur les problèmes de bords que l'utilisation de telles décompositions induisent et nous proposerons un moyen simple de les résoudre en choisissant des frames appropriés.

La seconde partie du chapitre sera consacrée à la définition de nouvelles caractéristiques de textures construites à partir des coefficients de la décomposition sur le frame. Pour la construction de ces nouvelles caractéristiques nous ferons appel à des méthodes d'analyse en composantes principales (ACP). Après avoir fait une étude de stabilité des caractéristiques en fonction de la taille du voisinage d'estimation, sur différentes textures, nous construirons deux algorithmes de segmentation non supervisée d'images texturées.

La dernière partie du chapitre comparera les algorithmes proposés à des méthodes utilisant des histogrammes. Nous mettrons alors en évidence les ressemblances entre ces méthodes et celles que nous avons développées.

4.1 Traitement des problèmes de bords dans les images filtrées par des frames de $l_2(\mathbb{Z}^2)$

Dans cette section, on considère une image de taille $N \times M$ décomposée sur un frame à la profondeur J. Les coefficients de la décomposition correspondent à la projection dans $3J+1$ sous-espaces et sont notés comme au chapitre précédent :

$$(s_J[l,m], d_{1,j}[l,m], d_{2,j}[l,m], d_{3,j}[l,m])_{0<j\leq J,\quad (l,m)\in\mathbb{Z}^2}$$

Nous allons déterminer à chaque échelle j, les supports des images de coefficients. Nous nous plaçons dans le cadre restrictif de filtres h et g pairs et ayant le même support. On justifie ensuite l'utilisation de tels filtres et on en exhibe un que l'on utilisera par la suite.

4.1.1 Support des images de coefficients

Considérons des filtres h et g, pairs, de même support $[-r,r]$, où r est un entier correspondant à l'indice n maximal pour lesquels $h[n]$ et $g[n]$ sont non nuls. Rappelons, ensuite, que dans le cas de l'utilisation de filtres h et g pairs, les formules de décomposition vérifient les relations entre échelles :

$$s_j[l,m] =< x[k,p], h_j[k-l]h_j[p-m] >= h_{\uparrow 2^{j-1}}[l] * h_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$

$$d_{1,j}[l,m] =< x[k,p], h_j[k-l]g_j[p-m] >= h_{\uparrow 2^{j-1}}[l] * g_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$

$$d_{2,j}[l,m] =< x[k,p], g_j[k-l]h_j[p-m] >= g_{\uparrow 2^{j-1}}[l] * h_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$

$$d_{3,j}[l,m] =< x[k,p], g_j[k-l]g_j[p-m] >= g_{\uparrow 2^{j-1}}[l] * g_{\uparrow 2^{j-1}}[m] * s_{j-1}[l,m]$$

où s_0 est le signal initial. La définition de $h_{\uparrow 2^j}$ et de $g_{\uparrow 2^j}$ nous indique que leur support est $[-2^j r, 2^j r]$. On peut alors calculer la taille de chacune des images filtrées, connaissant la taille de l'image de départ.
En effet :
soit s_{j-1} définie sur $[0, N_{j-1}-1] \times [0, M_{j-1}-1]$. Utilisant la relation vérifiée par s_{j-1} et s_j, on déduit que $s_j(l,m)$ est non nul pour :

$$(l,m) \in [-2^{j-1}r, N_{j-1} - 1 + 2^{j-1}r] \times [-2^{j-1}r, M_{j-1} - 1 + 2^{j-1}r]$$

Donc, par récurrence, si l'on suppose que s_0 est non nul sur $[0, N-1] \times [0, M-1]$, on obtient que s_J est non nul pour :

$$(l,m) \in [\sum_{j=0}^{J-1} -2^j r, N - 1 + \sum_{j=0}^{J-1} 2^j r] \times [\sum_{j=0}^{J-1} -2^j r, M - 1 + \sum_{j=0}^{J-1} 2^j r]$$

Il en est alors de même pour :

$$d_{1,j} \quad d_{2,j} \quad d_{3,j},$$

qui ont le même support que s_j.

4.1.2 Traitement des problèmes de bords

On remarque, donc, que la taille des images de coefficients augmente avec la profondeur J de décomposition. Nous souhaiterions associer à chaque pixel de l'image initiale un pixel dans chaque image de coefficients. L'utilisation de filtres h et g pairs permet de ramener la taille de chaque image de coefficients à la taille de l'image initiale, par extension miroir.

Ceci revient à remplacer la valeur de chaque pixel des images de coefficients à l'échelle j, et se trouvant dans le support de l'image initiale, par la somme de la valeur du pixel considéré et du(des) pixel(s) symétrique(s) du premier point par rapport au(x) bord(s) initiaux de l'image. L'idée est alors d'associer à chaque pixel de l'image initiale les valeurs des pixels correspondants des images de coefficients, modifiées selon ce procédé.

Cette technique n'a de sens que du fait que les filtres utilisés sont pairs, et revient en fait à considérer que l'image initiale a été prolongée par symétrie par rapport aux bords puis filtrée. Le croquis de la Figure 4.1 aide à comprendre, en une dimension, en quoi la technique utilisée est équivalente à la prolongation de l'image par symétrie.

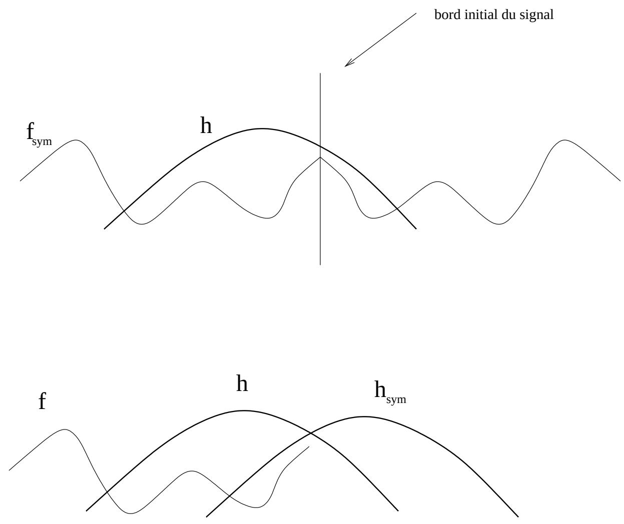

FIG. 4.1 – En haut : signal f symétrisé multiplié par h, En bas : signal f multiplié par h et le symétrique de h par rapport aux bords du signal, on a : $f_{sym}h = fh + fh_{sym}$, lorsque les filtres sont pairs

4.1.3 Choix des filtres

Comme il n'est pas possible en utilisant des filtres h et g à la fois pairs et de longueur finie de construire un frame orthogonal de $l_2(\mathbb{Z}^2)$ (cf section 3.5.1), si nous choisissons des filtres h et g vérifiant ces deux propriétés pour la construction des familles de filtres selon le procédé introduit par Rioul [85] et rappelé en section 3.6, le banc de filtres obtenu n'est en aucun cas un frame orthogonal.

Cependant, nous avons considéré des filtres h et g pairs dont le support est infini et qui génèrent des frames orthogonaux ; les filtres associés aux ondelettes de Battle et Lemarié dans les cas adéquats (cf section 3.5.1). En tronquant ces filtres, les familles de filtres que l'on génère en appliquant le procédé de la section 3.5.1 sont composées d'élements pairs mais ne forment plus un frame orthogonal (par troncature, on entend $\hat{h}(\omega) \approx \sum_{-P}^{P} h[n]e^{-in\omega}$). Bien que les familles de filtres ne forment plus un frame orthogonal, on constate que la spécifité fréquentielle des filtres est tout à fait acceptable, comme en témoigne la représentation de ces filtres à différentes échelles en Figure 4.2.

4.1.4 Validation du choix des filtres sur des images texturées

L'utilisation de filtres tronqués associés à un frame orthogonal fait que les formules de reconstruction associées ne sont que des approximations. Cependant, en pratique, les images re-

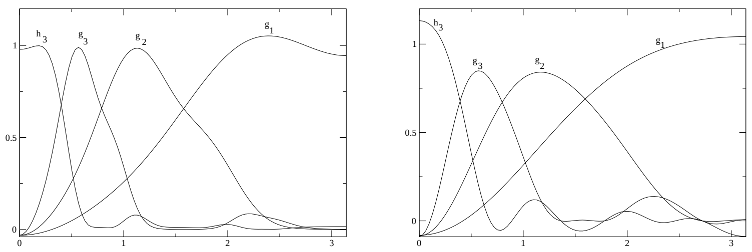

FIG. 4.2 – filtres de Battle et Lemarié tronqués à l'ordre $n = 4$ et $n = 2$, pour $J = 3$

construites, en utilisant ces formules sur des images de coefficients obtenues par décomposition sur un banc de filtres dérivé des filtres de Battle et Lemarié linéaires tronqués, sont tout à fait satisfaisantes, comme le montre la Figure 4.3. Le banc de filtres sera donc généré à partir d'un

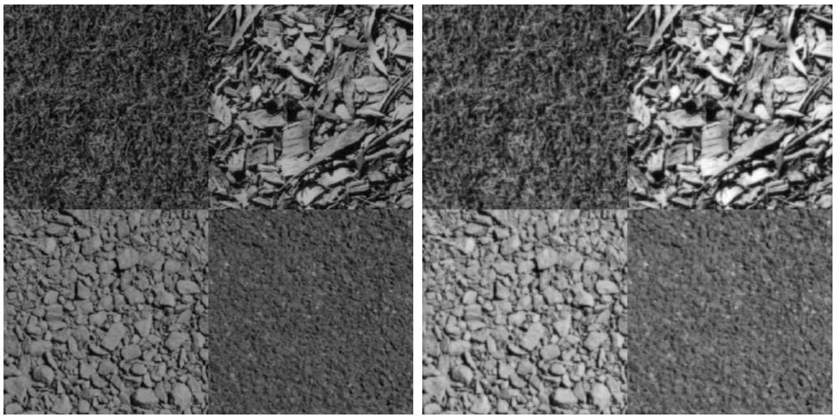

FIG. 4.3 – Image de droite : image initiale, Image de gauche : image obtenue par décomposition, à la profondeur $J = 1$, en utilisant un banc de filtres construit à partir d'un filtre de Battle et Lemarié linéaire tronqué ($n = 2$), puis recomposition en utilisant la formule associée au frame orthogonal

filtre de Battle et Lemarié linéaire tronqué et nous traiterons les problèmes de bords par extension miroir. Nous allons maintenant voir comment utiliser les coefficients de la décomposition sur les bancs de filtres associés pour la segmentation d'images texturées.

4.2 Notations utilisées dans les algorithmes

Dans les sections suivantes, on suppose que les coefficients aux différentes échelles issus d'une décomposition d'une image I sur le banc de filtres à la profondeur J, au point $[k, l]$, sont écris sous forme d'un vecteur :

$$v[k,l] = \begin{pmatrix} s_J[k,l] \\ d_{1,J}[k,l] \\ d_{2,J}[k,l] \\ d_{3,J}[k,l] \\ \vdots \\ d_{1,1}[k,l] \\ d_{2,1}[k,l] \\ d_{3,1}[k,l] \end{pmatrix}$$

On note $P = 3J+1$, le nombre de composantes de v.

4.3 Approche Bayésienne pour la segmentation non supervisée des textures

Dans cette partie, on suppose que l'image I est composée de plusieurs textures différentes dont on connaît le nombre k. Nous allons chercher à déterminer la localisation de chacune d'entre elles dans l'image.

4.3.1 Introduction

On considère une répartition initiale des pixels de l'image en k groupes. Les proportions de pixels appartenant aux différents groupes sont notées $p_1, p_2, \cdots, p_k$ et sont comprises entre 0 et 1. Chaque groupe est muni d'une densité de probabilité $f_j(v)$. On définit alors la probabilité d'appartenance au groupe G_j lorsque v est observé par (formule de Bayes) :

$$P(G_j/v) = \frac{p_j f_j(v)}{\sum\limits_{j=1}^{k} p_j f_j(v)}$$

La règle consiste alors à affecter l'observation v au groupe qui a la probabilité maximale, c'est-à-dire, comme les dénominateurs sont toujours les mêmes, à calculer :

$$\max_j p_j f_j(v)$$

Lorsque v est supposé être un vecteur gaussien qui suit une loi $N(\mu_j, \sum_j)$, où μ_j étant la moyenne empirique de chaque composante (centre de gravité), et $\sum_j$ l'estimation de la matrice de covariance, la densité associée est :

$$f_j(v) = \frac{1}{(2\pi)^{\frac{k}{2}}|det\sum_j|^{\frac{1}{2}}} \exp(-\frac{1}{2}(v-\mu_j)^T {\sum}_j^{-1}(v-\mu_j))$$

La règle Bayésienne consiste alors à minimiser :

$$(v-\mu_j)^T {\sum}_j^{-1}(v-\mu_j) - 2\ln(p_j) + \ln(|det{\sum}_j|)$$

Supposons que l'on considère des variables centrées, alors la formule se réécrit :

$$v'^T D^{-1} v' - 2\ln(p_j) + 2\sum_{i=1}^{N} \ln(\sigma_i^j)$$

Ces méthodes cherchent à caractériser les textures par la variance des composantes de v. Nous verrons que lorsque les variances et moyennes des composantes associées aux plus grandes échelles (les premières composantes de v) sont voisines, l'approche Bayésienne ne marche pas. Par ailleurs, il est à noter que le critère Bayésien pénalise les regroupements de petites tailles ($-\ln(p_j)$), ainsi que les grandes variances ($\ln(\sigma_i)$). Nous proposons dans la section suivante un algorithme simple de segmentation non supervisée dont l'analyse critique nous permettra de construire des algorithmes plus performants de segmentation non supervisée.

4.3.2 Méthodes des nuées dynamiques et algorithme de segmentation

Ces méthodes sont connues sous le nom d'algorithme "K means", en traitement d'images [22]. Elles supposent que l'on connaisse le nombre de textures présentes dans l'image. La démarche se fait en plusieurs étapes :

- choix de k germes aléatoires dans l'image (k nombre de textures)
- première affectation en utilisant la distance euclidienne
- affectations successives en utilisant le critère bayésien, jusqu'à "convergence" de la méthode.
- éventuellement choix de k nouveaux germes pour vérifier la validité du résultat.

En théorie, la convergence de la méthode n'est pas assurée, car le critère bayésien n'est pas une distance. En effet, considérons un découpage de l'image en sous images $E_1, ..., E_k$, pour avoir la convergence non devrions pouvoir écrire que :

$$\sum_{j=1}^{k}(\sum_{v\in E_j} v^T D_j^{-1} v - 2\ln(p_j) + 2\sum_{i=1}^{k}\ln(\sigma_i^j))^2 \geq \sum_{j=1}^{k}(\sum_{v\in E_j} v^T {D'}_j^{-1} v - 2\ln(p'_j) + 2\sum_{i=1}^{k}\ln({\sigma'}_i^j))^2$$

$$\geq \sum_{j=1}^{k}(\sum_{v\in B_j} v^T {D'}_j^{-1} v - 2\ln(p'_j) + 2\sum_{i=1}^{k}\ln({\sigma'}_i^j))^2$$

où D'_j, σ'^j et p'_j sont calculés à partir de l'ensemble E_j, et B_j est le meilleur regroupement des points de l'image en fonction de ces paramètres. Le premier membre considère une répartition particulière ; on calcule alors les caractéristiques associées à ces répartitions (D',p',σ') sans changer l'affectation des points, dans le second membre, et enfin on affecte les points en fonction de leurs caractéristiques les plus proches (troisième membre). L'inégalité de droite est toujours vraie à condition que les termes de la somme soient positifs (i.e : $x^T D_j^{-1} x - 2\ln(p'_j) + 2\sum_{i=1}^{N}\ln({\sigma'}_i^j) > 0$), puisque l'affectation dans le deuxième membre n'est pas optimale par rapport aux caractéristiques. Le principal problème est que la première inégalité n'est pas toujours vérifiée. Cependant, dans les cas que nous avons traités, la convergence se produit toujours, et semble indépendante des germes initiaux choisis.

4.3.3 Premiers résultats de segmentation

Dans cette partie, nous présentons un premier résultat de segmentation basé sur l'algorithme décrit plus haut (voir Figure 4.4). Nous nous sommes limités à une profondeur de décomposition $J = 3$, le filtre utilisé pour la détermination du banc de filtres est un filtre de Battle et Lemarié linéaire tronqué pair à cinq coefficients. Nous avons alors appliqué l'algorithme précédent à une image de deux textures en considérant un choix aléatoire de deux germes initiaux.

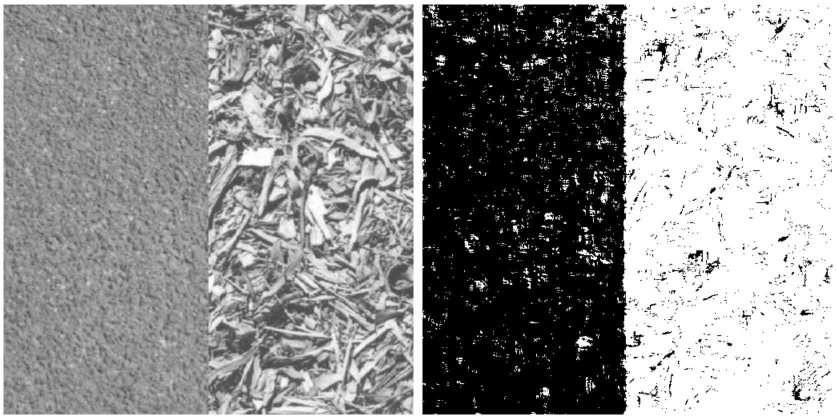

FIG. 4.4 – image de gauche : image composée des textures Asphalt et Misc, image de droite : résultat obtenu en utilisant le critère Bayésien

4.3.4 Interprétation du résultat

Une étude plus poussée montre que, si les résultats de segmentation sont relativement bons, cela est essentiellement dû au fait que les textures étudiées ont des natures fréquentielles très différentes (l'une haute fréquence, la texture Asphalt, et l'autre plutôt basse fréquence, la texture Misc) que le banc de filtres permet de bien caractériser. En utilisant toujours le même algorithme, mais avec des textures ayant des comportements voisins aux grandes échelles pour le banc de filtres utilisé (exemple : une image de textures comme celle de la Figure 4.7), la segmentation deviendra nettement moins fiable.
En étudiant les points frontières des textures, on constate par ailleurs que ceux-ci sont préférentiellement affectés à la texture contenant des énergies plus importantes aux grandes échelles (dans le cas de l'exemple de la Figure 4.4, un point frontière sera préférentiellement affecté à la texture Misc plutôt qu'à la texture Asphalt). De plus, de nombreux points sont mal affectés en raison de l'absence de relations spatiales entre coefficients. Nous allons développer, dans la section suivante, des algorithmes utilisant de manières diverses des ACPs, qui permettront de segmenter correctement des images texturées alors que l'analyse Bayésienne ne les traite pas correctement.

4.4 Approches déterministes pour la classification de textures

4.4.1 Méthodes de classification utilisant l'hypothèse gaussienne

Dans cette section, nous allons présenter plusieurs distances qui permettent la classification des textures. Toutes ces approches font l'hypothèse gaussienne sur le vecteur v, même si celle-ci apparaît clairement uniquement pour la première distance présentée.
Matrice de covariance et distance de Mahalanobis [16]
On suppose que l'image de texture I est caractérisée par le vecteur μ des moyennes de chaque composante de v et par la matrice de covariance M de v. Soit w un vecteur de caractéristiques obtenu par décomposition sur le même banc de filtre, alors on peut définir la distance, dite de Mahalanobis, de w à I par :

$$d^2(w, I) = (w - \mu)^T M^{-1}(w - \mu)$$

L'avantage d'une telle distance par rapport à la distance euclidienne est double. En mélangeant les échelles, cette distance, par le biais de la matrice de covariance, permet de prendre en compte les relations existantes entre les coefficients aux différentes échelles (redondance), dues à l'utilisation d'un frame. Ensuite, la matrice de covariance joue un rôle de normalisation des composantes.
Si l'on suppose de plus que chaque composante de v est de moyenne nulle ($\mu = 0$), nous pouvons faire le lien avec l'analyse en composantes principales, que nous détaillerons dans la section suivante. En effet, si on diagonalise, dans ce cas, la matrice de covariance M, en la supposant de rang P, on obtient :

$$d^2(w, I) = w^T B D^{-1} B^T w$$

puis, le changement de variables $w' = B^T w$ donne :

$$d^2(w, I) = w' D^{-1} w'$$

où D est une matrice diagonale telle que $D[i, i] = \sigma_i^2$. Donc, la distance peut se réécrire :

$$d^2(w, I) = \sum_{i=1}^{P} \frac{w'[i]^2}{{\sigma_i}^2}$$

Calculer la distance de Mahalanobis revient donc à calculer les coefficients de la décomposition, puis si ceux-ci sont de moyenne nulle, à calculer la matrice de covariance associée, pour ensuite calculer les facteurs principaux et, les composantes principales associées aux vecteurs de coefficients.

Un algorithme de classification supervisée consiste à déterminer sur un ensemble de textures de référence les paramètres μ et M et à affecter tout vecteur w à la texture de référence la plus proche au sens de la distance de Mahalanobis.

Variables centrées réduites et distance euclidienne [63]

L'image de texture I est cette fois caractérisée par la moyenne des composantes de v, $\mu_i \quad i \in [1, \cdots, P]$, ainsi que par leur variance $\sigma_i^2 \quad i \in [1, \cdots, P]$. La distance qu'on lui associe le plus souvent est la distance euclidienne sur les composantes centrées réduites. Cela peut être compris en remarquant que si les filtres sont suffisamment sélectifs, la corrélation entre les différentes échelles sera faible. Cette approximation sera d'autant mieux vérifiée que l'on tronque peu les filtres.
Considérons :

$$d^2(w, I) = \sum_{i=1}^{P} \frac{(w_i - \mu_i)^2}{\sigma_i^2}$$

Une technique de classification supervisée consiste alors à déterminer μ et σ pour un ensemble de textures de référence et à affecter tout vecteur w à la texture la plus proche.

Distance associée à la médiane et à l'écart absolu moyen [98]

L'image de texture I est cette fois caractérisée par la médiane des composantes de v et leur écart absolu moyen. les distances précédemment introduites présentent le désavantage que le calcul des moyennes et variances peut être soumis à des erreurs, du fait d'éléments qui s'écartent trop de la moyenne μ_i [98]. L'utilisation de la médiane et de l'écart absolu moyen est moins sensible aux valeurs aberrantes.
On note la médiane $m(v) = (m(v[1]), \cdots, m(v[P]))$ et l'écart absolu moyen $em(v)$ est donné par :

$$em(v[i]) = \frac{1}{|I|} \sum_{v \in I} |v[i] - m(v[i])|,$$

Alors, la distance à I d'un vecteur w défini comme précédemment s'écrit :

$$d(w,I) = \sum_{i=1}^{P} \frac{|v[i] - m(v[i])|}{em(v[i])}$$

On peut déduire alors un algorithme de classification non supervisée en utilisant le même principe qu'aux deux paragraphes précédents.

4.4.2 Motivation pour une approche déterministe sans hypothèse gaussienne

Comme il a été remarqué dans [68], la plupart des textures que l'on désire étudier ne sont, ni des échantillons gaussiens, ni des processus stationnaires. Par ailleurs, les caractéristiques extraites ne vérifient pas, non plus, ces propriétés. Lorsqu'une approche Bayésienne est utilisée ([3],[61]), l'hypothèse Gaussienne est pourtant souvent faite.
De même, l'hypothèse de stationnarité que l'on utilise lorsque l'on cherche à reconnaître localement un processus n'est pas toujours valable et celle-ci dépend, en tout cas, étroitement du voisinage de l'image sur lequel on se place.

En analyse de textures par utilisation de bancs de filtres, plusieurs points de vue contradictoires s'affrontent, quant à savoir si l'étape primordiale est la détermination du filtre sur lequel on décompose l'image [82] ou le traitement des coefficients ainsi obtenus [32]. Bien que les filtres doivent bien adaptés à l'image traitée, nous pensons que l'étape de traitement est primordiale. En effet, dans la plupart des algorithmes proposés, les coefficients obtenus par décomposition sur les filtres sont d'abord moyennés en utilisant des fenêtres carrés ([89],[95], [20]) ou des fenêtres gaussiennes, puis, ces nouveaux coefficients sont traités comme des variables Gaussiennes, ce qui est source d'erreurs. Il a été montré numériquement, dans [21], que l'hypothèse Gausienne est pratiquement toujours fausse, en étudiant la répartition des niveaux de gris de nombreuses textures.

4.4.3 Justification de la non-normalité

Pour vérifier l'invalidité de l'hypothèse gaussienne, on peut calculer le curtosis de l'ensemble des niveaux de gris de l'image, ainsi que leur asymétrie ("skewness"). On suppose que les niveaux de gris de l'image sont rangés dans un vecteur de taille N, noté $(x_1, \cdots, x_N)$, que l'on appelle échantillon. Si cet échantillon est Gaussien, on doit avoir chacune des deux quantités adimensionnelles suivantes égales à zéro :

$$curt(x_1, \cdots, x_N) = \frac{1}{N} \sum_{j=1}^{N} (\frac{x_j - \bar{x}}{\sigma})^4 - 3$$

$$asym(x_1, \cdots, x_N) = \frac{1}{N} \sum_{j=1}^{N} (\frac{x_j - \bar{x}}{\sigma})^3,$$

où $\bar{x}$ représente la moyenne du vecteur x. L'asymétrie indique que les niveaux de gris ne sont pas symétriquement répartis par rapport à leur moyenne, tandis que le curtosis étudie la décroissance des niveaux de gris par rapport à leur moyenne. On peut donc définir une mesure simple permettant de savoir si l'échantillon est gaussien en considérant :

$$G(x_1, \cdots, x_N) = \sqrt{curt(x_1, \cdots, x_N)^2 + asym(x_1, \cdots, x_N)^2}$$

Nous calculons pour les images de textures considérées la valeur de G (cf 4.5), pour vérifier l'hypothèse précédente. Etant donné la taille des échantillons, 512×512, le curtosis et l'asymètrie est très faible, et, pour toutes les textures étudiées le curtosis et l'asymétrie correspondent à une probabilité très faible (inférieure à 1 %) que les textures soient gaussiennes.

	G
Roc	0.81
Misc	1.19
Asphalt	0.68
Grass	0.64

FIG. 4.5 – valeur du paramètre G pour les textures de référence étudiées

Nous allons développer dans les sections suivantes des algorithmes de segmentation n'utilisant pas l'hypothèse gaussienne sur v. Ceux-ci utiliseront largement les décompositions en composantes principales (ACP) que nous présentons maintenant.

4.5 Analyse en composantes principales du vecteur de coefficients

On suppose que chacune des composantes de v est de moyenne nulle et l'on construit la matrice de covariance associée :

$$M = \frac{1}{|I|}\sum_{[k,l]\in I} v[k,l]v[k,l]^T$$

qui est symétrique définie positive (si elle est de rang P), de taille $P\times P$ et admet donc P valeurs propres positives que l'on range par ordre décroissant.

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_P \geq 0$$

On note alors $(u_1, u_2, \cdots, u_P)$ une base de vecteurs propres orthonormés associés, puis H_m la matrice formée par les m premiers vecteurs propres :

$$H_m = \begin{pmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_m^T \end{pmatrix}$$

Les éléments u_i s'appellent les facteurs principaux. On définit ensuite la projection orthogonale, sur le sous-espace engendré par les u_i, des vecteurs $v(k,l)$ par :

$$c[k,l] = H_m v[k,l] \quad ,\forall [k,l]\in I$$

Les éléments ainsi définis s'appellent les composantes principales. Ces composantes ont la propriété d'être décorrélées et leur variance correspond aux valeurs propres de la matrice de covariance. En effet, la variance de la ième composante de $c[k,l]$ est :

$$V(c_i[k,l]) = \frac{1}{|I|}\sum_{(k,l)\in I} c_i[k,l]^2$$

$$= u_i^T M u_i = \lambda_i u_i^T u_i = \lambda_i.$$

Si on calcule maintenant la covariance de deux composantes distinctes, nous obtenons :

$$\frac{1}{|I|} \sum_{(k,l) \in I} c_i[k,l] c_j[k,l]$$

$$= u_i^T M u_j = \lambda_j u_i^T u_j = 0$$

Regardons ensuite les choses du point de vue de l'approximation. Projetons v sur une famille de m vecteurs $(g_j)_{j \in [1,m]}$ orthonormés de $\mathbb{R}^P$. La décomposition s'écrit :

$$\bar{v} = \sum_{j=1}^{m} < v, g_j > g_j$$

où $<,>$ est le produit scalaire de l'espace $\mathbb{R}^P$. Regardons alors l'erreur quadratique commise au point $[k,l]$:

$$\epsilon[m](v) = \sum_{m+1}^{P} < v, g_j >^2$$

Par ailleurs, comme :

$$E\{< v, g_j >^2\} =< M g_j, g_j >,$$

on peut alors écrire :

$$E\{\epsilon[m](v)\} = \sum_{m+1}^{P} < M g_j, g_j >,$$

où E désigne la moyenne empirique. On montre alors facilement que la base qui minimise cette erreur est celle des vecteurs propres de M (facteurs principaux) associés aux $P - m$ valeurs propres les plus petites. La projection sur les sous-espaces associés aux valeurs propres les plus grandes permet donc de trouver la meilleure approximation, au sens de :

$$E\{\epsilon[m](v)\} = \min_{M_m} E\{\|v - v_m\|^2\},$$

où M_m est l'ensemble des familles orthonormées de taille m. Donc, la décomposition en composantes principales permet de trouver des caractéristiques, les composantes principales, réalisant l'écart minimal en moyenne quadratique avec le vecteur initial de coefficients.

Les composantes du vecteur v n'ont pas toutes le même ordre de grandeur et l'analyse en composantes principales va privilégier celles dont le module est le plus fort.
Exemple :

Supposons que la structure à analyser soit uniformément Lipschitzienne α, alors nous avons la majoration suivante pour les coefficients (voir chapitre 5) :

$$\forall (l,m) \in \mathbb{Z}^2 \quad \sqrt{d_{1,j}^2 + d_{2,j}^2} \leq A 2^{j\alpha}$$

Donc, de l'échelle j à l'échelle $j+1$, l'ordre de grandeur des coefficients est multiplié par un facteur 2^α. Autrement dit, les espaces correspondants aux basses fréquences (associés aux grandes échelles) sont favorisés par rapport à ceux correspondants aux hautes fréquences (petites échelles) pour des textures régulières. Le contraire se produisant pour des textures très irrégulières ($\alpha < 0$).

4.6 Algorithmes de segmentation non supervisée utilisant des ACPs

Dans cette section, nous allons calculer pour différentes images de texture les facteurs principaux et les valeurs propres associés à la matrice de covariance M. Nous mettrons en évidence, le fait qu'il est facile de distinguer des textures dont les directions des facteurs principaux associés aux plus fortes énergies sont relativement différentes, et, qu'en revanche les cas difficiles se posent lorsque les facteurs principaux associés aux grandes échelles sont très similaires. Puis, nous étudierons la stabilité des facteurs principaux en fonction de la taille des voisinages servant à leur calcul ; nous parlerons d'échelles pour désigner la taille des voisinages. Cette étude nous permettra ensuite de construire des algorithmes de segmentation non supervisée utilisant les facteurs principaux.

4.6.1 Comparaison des facteurs principaux associés à différentes textures

Si nous souhaitons segmenter des images de texture en utilisant les facteurs principaux, il est nécessaire que ceux-ci soient discriminants pour les textures étudiées. Par ailleurs, les facteurs principaux ne jouent pas un rôle identique. Ceux associés aux énergies les plus fortes sont les plus représentatifs de la texture, mais aussi les plus stables en fonction de l'échelle (taille de l'échantillon servant au calcul), ce que nous allons étudier.

Sur l'exemple de la Figure 4.6, les facteurs principaux sont relativement dissimilaires aux grandes énergies. Ceci nous indique que, dans ce cas, les facteurs principaux peuvent être intéressants à utiliser. On remarque, par ailleurs, que les facteurs principaux associés aux plus faibles énergies sont relativement similaires, indépendamment des textures étudiées, ils ne sont donc pas discriminants. Un cas plus délicat où les facteurs principaux associés à la plus grande

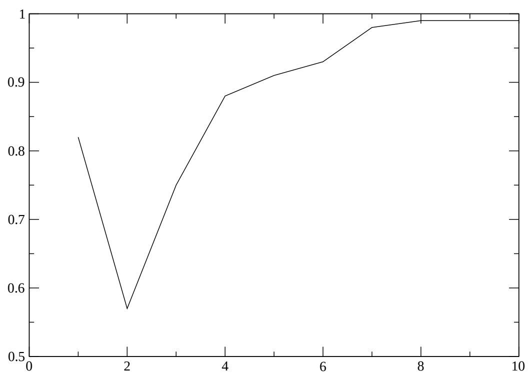

FIG. 4.6 – valeur absolue des produits scalaires des facteurs principaux des textures Asphalt et Misc, les abscisses correspondent au indices des facteurs principaux

énergie sont très proches est présenté en Figure 4.7. L'idée la plus naturelle pour segmenter de telles textures serait alors de considérer les espaces qui sont tels que les produits scalaires entre textures différentes sont les plus faibles en valeur absolue. Nous allons faire une étude détaillée de la stabilité des facteurs principaux en fonction de la taille du voisinage qui montrera que l'idée précédente n'est pas applicable.

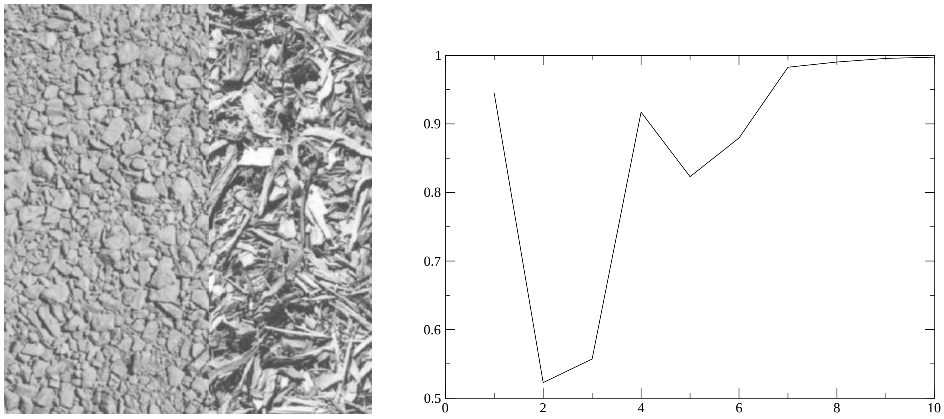

FIG. 4.7 – Image de gauche : image des textures Rock et Misc, graphe de droite : valeur absolue des produits scalaires des facteurs principaux, les abscisses correspondent ua indices des facteurs principaux

4.6.2 Stabilité des facteurs principaux et des valeurs propres en fonction de l'échelle

Dans cette section, nous considérons à nouveau une image I composée d'une seule texture et l'on suppose disposer en tout point de l'image du vecteur v, à $3J+1$ composantes, des coefficients de la décomposition sur le banc de filtres. Nous procédons alors à une étude plus approfondie des valeurs propres de la matrice de covariance du vecteur v que l'on note M, comme précédemment.

On envisage un découpage de l'image I en q sous images disjointes de taille égale. On note M_k la matrice de covariance associée à la sous-image k. La matrice de covariance M des coefficients de la décomposition sur le frame vérifie :

$$|I|M = |I_{k_0}| \sum_k M_k,$$

$|I_{k_0}|$ représentant la taille de n'importe laquelle des q sous-images. Alors, pour tout vecteur v, on a la relation :

$$|I| < Mw, w >= |I_{k_0}| \sum_k < M_k w, w >$$

De ceci, on tire immédiatement que :

$$|I|\lambda_1 \leq |I_{k_0}| \sum_{k=1}^{q} \quad \lambda_1^{(k)} \quad (1)$$

et :

$$\forall i \quad 1 \leq i \leq q \quad |I|\lambda_1 \geq |I_{k_0}|(\lambda_1^{(i)} + \sum_{k \neq i} \lambda_P^{(k)}) \quad (2)$$

où $\lambda_1 \geq \cdots \geq \lambda_P$ sont les valeurs propres de M.
Par ailleurs, par linéarité de la trace, on obtient :

$$|I| \sum_i \lambda_i = |I_{k_0}| \sum_{k=1}^{k=q} \sum_{i=1}^{i=P} \lambda_i^{(k)} \quad (3)$$

De (2), on déduit que si $|I|\lambda_1 = |I_{k_0}|\lambda_1^{(i)}$, alors les matrices S_k pour $k \neq i$ sont de rang au plus $P-1$, la plus petite valeur propre étant 0. En pratique, cela ne se produit jamais et l'on a toujours :

$$\forall i \quad 1 \leq i \leq q \quad |I|\lambda_1 > |I_{k_0}|\lambda_1^{(i)} \quad (4)$$

Supposons maintenant que l'on ait le même facteur principal associé à la ième valeur propre sur l'image et sur chacune des sous-images, alors on obtient :

$$|I|\lambda_i = |I_{k_0}| \sum_{k=1}^{q} \lambda_i^{(k)} \quad (5)$$

Supposons alors que $\lambda_i^{(k)} = \lambda_i^{(j)}$ pour $k \neq j$, alors il vient que $\lambda_i = \lambda_i^{(j)}$. Par ailleurs, si $\sum_p \lambda_p^{(k)} = \sum_p \lambda_p^{(j)}$ alors, on a la normalisation suivante :

$$\frac{\lambda_i}{\sum\limits_{p=1}^{p=P} \lambda_p} = \frac{|I_{k_0}|}{|I|} \frac{\sum\limits_{k=1}^{k=q} \lambda_i^{(k)}}{\sum\limits_{p=1}^{p=P} \lambda_p} = \frac{1}{q} \frac{\sum\limits_{k=1}^{k=q} \lambda_i^{(k)}}{\sum\limits_{j=1}^{j=P} \lambda_j^{(k)}} = \frac{\lambda_i^{(k)}}{\sum\limits_{j=1}^{j=P} \lambda_j^{(k)}}.$$

Inversement, supposons que l'on ait :

$$\frac{\lambda_i}{Tr(S)} = \frac{\lambda_i^{(1)}}{Tr(S_1)} = \frac{\lambda_i^{(2)}}{Tr(S_2)} = \cdots = \frac{\lambda_i^{(q)}}{Tr(S_q)}$$

alors, il vient :

$$\sum_{j=1}^{j=q} \lambda_i^{(j)} = \sum_{j=1} \frac{Tr(S_j)}{Tr(S)} \lambda_i = \frac{|I|}{|I_{k_0}|} \lambda_i.$$

L'égalité des valeurs propres n'implique pas forcément l'existence d'un facteur principal commun.

Nous allons déterminer pour quelle valeur propre la propriété (5) est une bonne approximation. Pour cela, il faut vérifier plusieurs choses. La première est la stabilité du facteur principal associé à une valeur propre sur différentes sous-images d'une même texture prises à la même échelle et la seconde est la stabilité du facteur principal pour une même texture mais calculé à des échelles différentes.

4.6.3 Stabilité des facteurs principaux en fonction de l'échelle

Dans cette partie, nous allons étudier la stabilité des facteurs principaux à une même échelle. Comme les facteurs principaux sont orthonormés, on peut mesurer leur stabilité relative en comparant les moyennes des carrés des produits scalaires mesurées sur des sous-images de I de même taille. Nous introduisons, donc, une définition de la stabilité des facteurs principaux à une échelle donnée :

Définition 1 *On dira que, pour une image de texture I, la stabilité du ième facteur principal est plus grande que celle du jème facteur principal, à l'échelle K, si, en considérant l'image découpée en sous-images de taille $K \times K$, on a :*

$$\sum_{k \neq p} < u_i^{(k)}, u_i^{(p)} >^2 \geq \sum_{k \neq p} < u_j^{(k)}, u_j^{(p)} >^2$$

où k et p désignent des sous-images de I, de taille $K \times K$

On peut ensuite définir la stabilité inconditionnelle, du ième facteur principal par rapport au jème, de la manière suivante.

Définition 2 *On dira que, pour une image de texture I, le ième facteur principal est inconditionnellement stable par rapport au jème, si quelque soit l'échelle K et, en considérant l'image découpée en sous-images de taille $K \times K$, on a :*

$$\sum_{k \neq p} < u_i^{(k)}, u_i^{(p)} >^2 \geq \sum_{k \neq p} < u_j^{(k)}, u_j^{(p)} >^2$$

où k et p désignent des sous-images, de taille $K \times K$, de I

Pour les textures étudiées, nous avons une propriété de stabilité inconditionnelle du premier facteur principal par rapport aux deuxième et troisième(voir Figure 4.8).

Pour mesurer cette stabilité nous divisons chaque image de texture que l'on souhaite tester en carrés disjoints de taille égale, puis l'on effectue une analyse en composantes principales du vecteur v sur chacune des sous-images. On calcule alors tous les carrés des produits scalaires entre facteurs principaux associés aux sous-images et correspondants aux mêmes indices de valeurs propres, et l'on considère leur valeur moyenne.

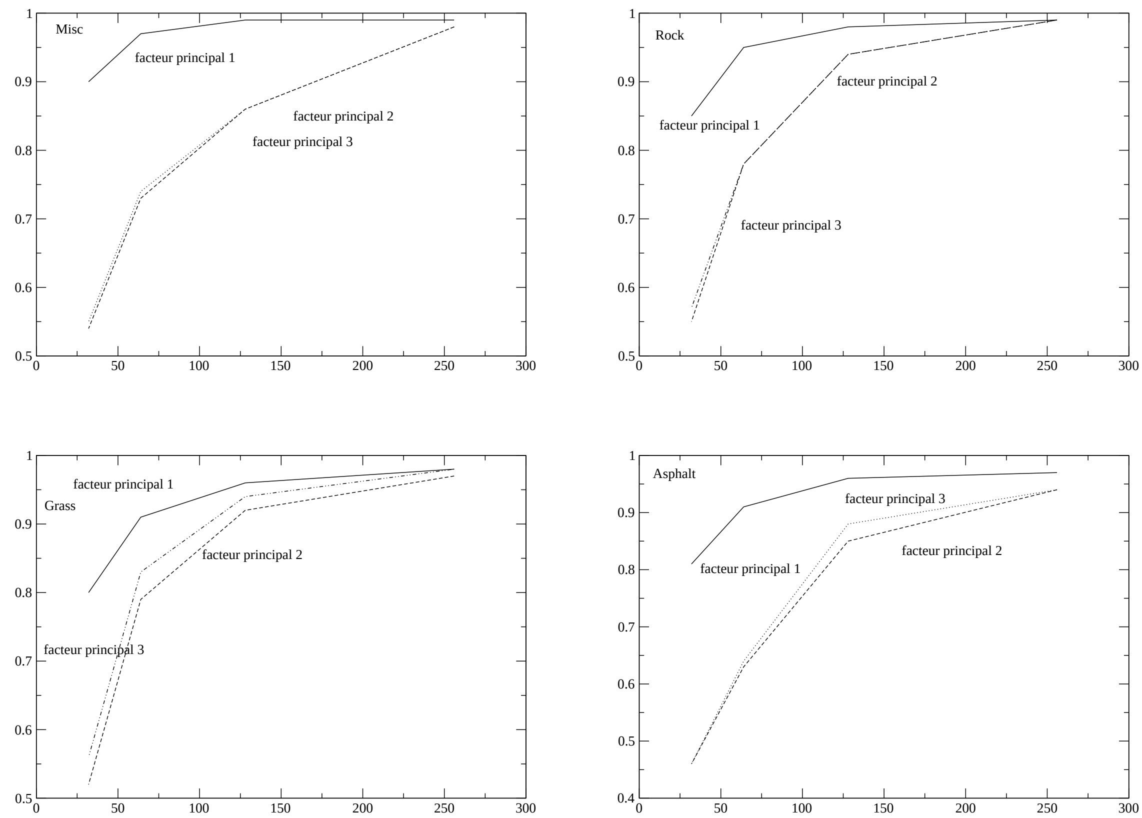

FIG. 4.8 – Valeur moyenne des carrés des produits scalaires des facteurs principaux, associés aux première, seconde et troisième valeurs propres, calculés sur des sous-images dont la taille est précisée en abscisse, pour les quatre textures de référence

Par ailleurs, Pour avoir la propriété (5), il faut vérifier que les facteurs principaux sont stables le long des échelles. Pour cela, on calcule les moyennes des valeurs absolues des produits scalaires entre les facteurs principaux calculés à l'échelle K, sur des sous-images disjointes, avec le facteur principal calculé sur l'image complète de texture. La propriété de stabilité des facteurs princi-

	32	64	128	256
fp1	0.95	0.98	0.99	1
fp2	0.75	0.92	0.98	1
fp3	0.45	0.65	0.86	1

	32	64	128	256
fp1	0.96	0.99	1	1
fp2	0.77	0.88	0.96	1
fp3	0.47	0.62	0.75	1

FIG. 4.9 – Produit scalaire moyen entre le facteur principal associé aux première, seconde et troisième valeur propre donnée, calculé à partir de toute l'image et les facteurs principaux correspondant aux mêmes valeurs propres mais calculés sur des sous-images, dont la taille est précisée en entrée du tableau ; à gauche pour la texture Misc et à droite pour la texture Rock

paux n'est à peu près vérifiée, pour les textures étudiées, que par le premier facteur principal et pour des échelles K suprérieures à 50 pixels. Dans nos simulations, nous ne considérerons donc qu'un seul facteur principal, étant donné la petitesse des voisinages sur lesquels nous devrons travailler, ce qui veut dire qu'à une texture correspondra une caractéristique. Chang [16] suggère de retenir 5 ou 6 caractéristiques des textures, ce qui montre que le nombre de caractéristiques à retenir est lié à leur nature.

De nombreuses études ont été menées sur le nombre de caractéristiques à conserver pour la segmentation des textures. Les deux conclusions indubitables sont que la précision de la segmentation n'est pas une fonction croissante du nombre des caractéristiques retenues [49], et que la détermination des caractéristiques pertinentes et leur nombre dépendent du modèle utilisé ([50],[80]). Nous proposons, dans la section suivante, des algorithmes de segmentation non supervisée fondés sur un découpage hiérarchique de l'image texturée, un regroupement en régions de l'image puis une segmentation finale

4.6.4 Algorithme de segmentation non supervisée en trois étapes

Dans cette partie, on construit des algorithmes de segmentation non supervisée d'images texturées que l'on testera sur l'image de la Figure 4.11.

Détection des textures par découpage hiérarchique (étape 1)

Nous pensons que le problème de la segmentation des textures est avant tout un problème d'échelle, et qu'un découpage hiérarchique de l'image (de type "coarse to fine") peut permettre une première reconnaissance des textures présentes dans l'image, puis qu'un second traitement permettra d'obtenir une segmentation correcte des images.

Les découpages hiérarchiques des images pour la segmentation des textures sont d'usage courant ([81],[92]). Les variations portent sur les critères retenus pour le découpage ; dans [92], celui-ci porte sur un paramètre de contraste et, dans [81], sur des caractéristiques locales d'histogrammes. Les approches hiérarchiques (qualifiées parfois d'approches multirésolution) permettent d'augmenter la précision des résultats, quelque soit les caractéristiques retenues pour la caractérisation des textures ([57],[69]). Expliquons alors le principe du découpage :

- On découpe initialement l'image en 2×2 sous-images.
- Puis, à nouveau, chaque sous-image en quatre sous-images, (I_1, I_2, I_3, I_4), et l'on estime les facteurs principaux sur chacune de ces sous-images. Notons $u_j^{I_p}$ le jème facteur principal de l'image I_p.

- On calcule les distances (pour $(p, p') \in \{1, \cdots, 4\}$, $p \neq p'$) :

$$\tilde{d}(u^{I_p}, u^{I_{p'}}) = 1 - \frac{1}{M} \sum_{j=1}^{j=M} | < u_j^{I_p}, u_j^{I_{p'}} > |$$

- alors, si :

$$\max_{I_p, I_{p'}} \tilde{d}(u^{I_p}, u^{I_{p'}}) < \alpha,$$

on est dans une même texture, sinon il faut à nouveau découper.
- Le découpage est arrêté, lorsque la taille des sous-images n'est plus fiable pour l'estimation des facteurs principaux.

L'avantage d'un tel découpage est qu'il compare des facteurs principaux calculés aux mêmes échelles. Le nombre M de facteurs principaux à conserver est fonction de leur stabilité le long des échelles. Un exemple de décomposition est présenté sur la Figure 4.10. On présente, en Fi-

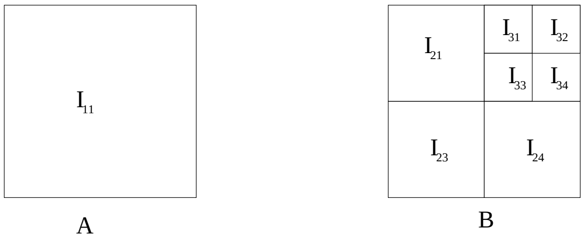

FIG. 4.10 – A: image initiale , B: exemple de découpage

gure 4.11, un exemple de découpage sur une image de textures. On constate que l'algorithme de découpage enregistre une différence angulaire plus importante du premier facteur principal à l'intérieur d'une même texture, que sur des régions contenant des frontières de textures. La texture Rock est plus instable vis-à-vis du critère retenu. L'intérêt d'un découpage plus important serait d'obtenir, après regroupement des régions, une meilleure approximation des bords. Cependant, c'est aussi un facteur d'instabilité car les facteurs principaux ne sont pas forcément stables pour des échelles petites (Figure 4.8). Nous regroupons, ensuite, les régions ainsi définies pour obtenir une première approximation des textures.

Regroupement des régions (étape 2)

Pour la procédure de regroupement, il est important de rappeler (cf Figure 4.9) que deux régions de taille différente peuvent avoir des facteurs principaux relativement différents, même si elles appartiennent à une même texture.

De ce fait, nous allons regrouper les régions adjacentes les plus proches, en commençant par les plus petites ; ce procédé a déjà été employé dans [81]. On définit donc le critère de regroupement suivant :

$$\min(\tilde{d}(u^I, u^{I'}) * \min(|I|, |I'|))$$

L'avantage de la multiplication par $\min(|I|, |I'|)$ est que l'on traite prioritairement les petites régions ; en même temps, on peut affecter une petite région à une plus grande, si les produits scalaires de leurs facteurs principaux sont suffisamment forts.

Le processus s'arrête lorsque le nombre de régions est égal au nombre de textures. Les régions ainsi obtenues sont des approximations des textures présentes dans l'image. On présente les résultats obtenus en Figure 4.12. L'intérêt de la méthode réside dans ce que l'approximation finale des textures est relativement stable en fonction du seuil. Cependant, un découpage trop important

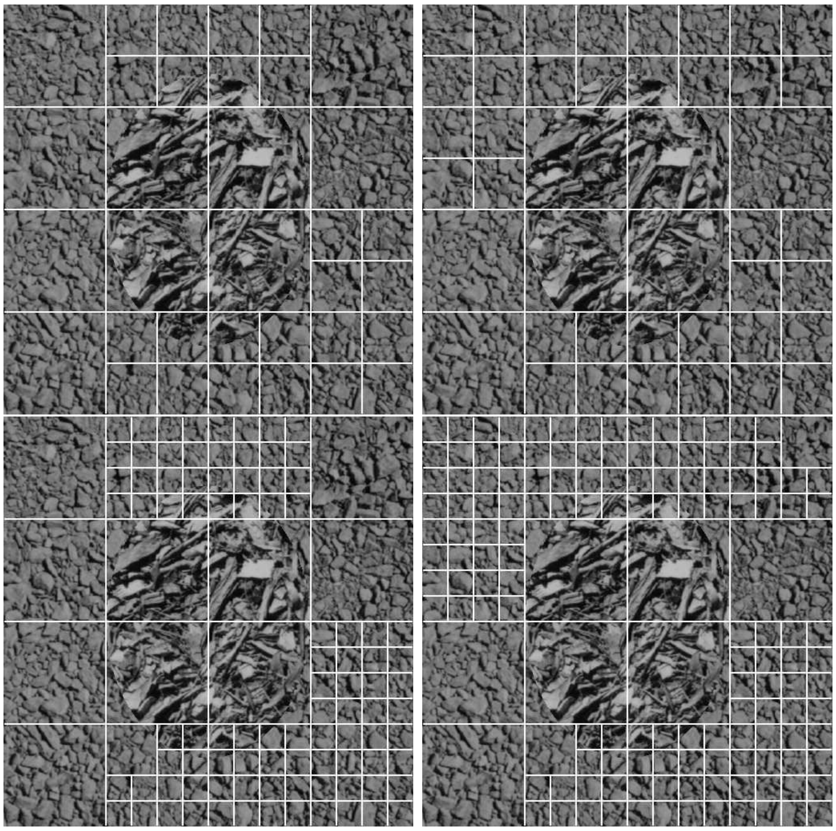

FIG. 4.11 – Découpage hiérarchique de l'image en utilisant un seul facteur principal, pour $\alpha = 0.03$ et $\alpha = 0.05$. En première ligne, on présente le découpage en s'arrêtant à des sous-images de taille 64 pixels, et en deuxième ligne 32 pixels

de l'image peut entraîner une mauvaise estimation des caractéristiques de textures. Nous sommes donc en désaccord sur ce point avec [81], qui explique que l'on a intérêt à décomposer davantage l'image de départ que pas assez. Ceci n'est vrai que pour des textures dont les caractéristiques sont stables aux échelles considérées. En effet, pour un découpage de l'image en carrés de taille 32 pixels, on obtient le regroupement de la Figure 4.13. Il apparaît donc qu'à des échelles trop petites, les caractéristiques ne suffisent plus pour regrouper convenablement les régions.

Segmentation finale (étape 3)

Segmentation en utilisant uniquement les facteurs principaux

Nous allons présenter des algorithmes utilisant uniquement les facteurs principaux pour la segmentation. Comme l'algorithme de regroupement assure la formation de textures connexes (d'un seul morceau), nous en tenons compte lors de cette dernière étape.

Les bords des textures peuvent être définis comme les pixels connectés à deux textures différentes, dans le sens qu'il existe, parmi les quatre pixels voisins les plus proches, deux pixels appartenant à des textures différentes.

On définit, ainsi, sur l'image obtenue après le regroupement en régions, une (ou des) frontière(s) sur l'image, que l'on va déplacer. Une des difficultés est de préserver le caractère continu de cette frontière.

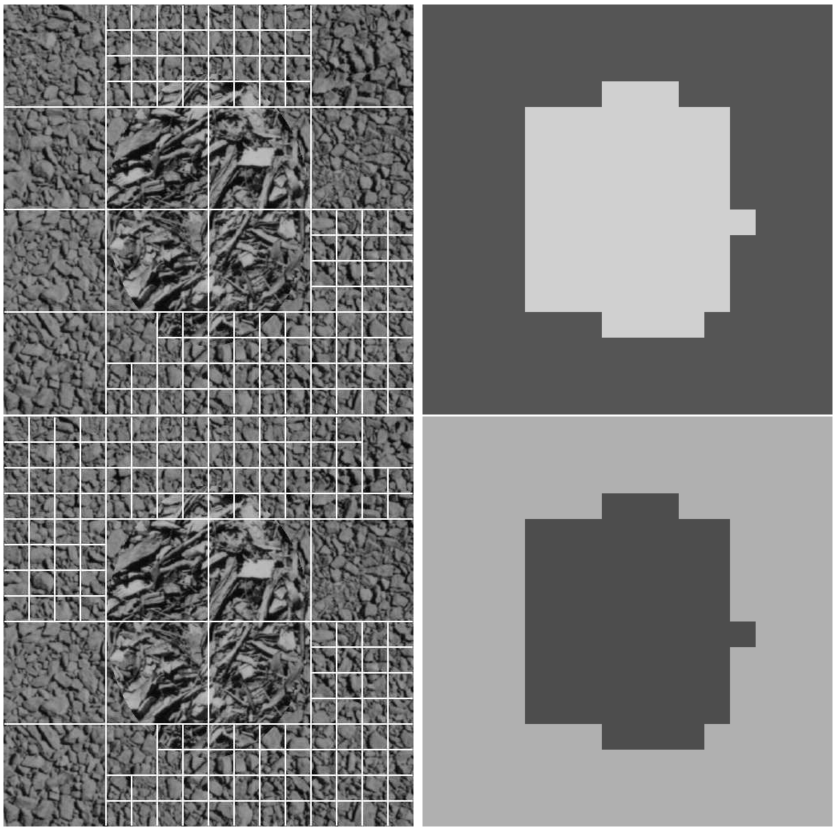

FIG. 4.12 – Première ligne : regroupement obtenu pour un seuil $\alpha = 0.05$, en utilisant un seul facteur principal. Deuxième ligne : idem avec $\alpha = 0.03$

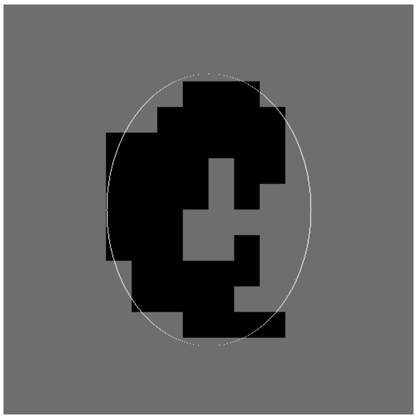

FIG. 4.13 – Regroupement des régions à partir d'un découpage de l'image en carrés de taille 32×32 pixels

Pour le regroupement des régions, l'algorithme n'utilisait qu'un seul facteur principal ; maintenant que l'on dispose d'une estimation plus précise, on peut se demander si on ne devrait pas utiliser plus de caractéristiques.
Par exemple, une technique simple consiste à calculer la somme des valeurs propres associées aux m premiers facteurs principaux et à diviser ensuite par la somme de toutes les valeurs propres (les valeurs propres sont ici toutes positives) [88]. Cette technique permet de retenir les facteurs principaux associées aux espaces de plus fortes énergies.
On présente les caractéristiques retenues pour différentes valeurs du seuil, qui correspond au pourcentage de l'énergie totale associée aux composantes principales contenue dans les sous-espaces que l'on conserve, pour les textures Rock et Misc, en Figure 4.14.

seuil	Misc	Rock
0.5	1	1
0.7	1,2	1,2
0.9	1,2,3	1,2,3,4

FIG. 4.14 – Espaces retenus pour avoir respectivement 50 % , 70 % et 90 % de l'énergie conservée

Pour la segmentation, nous considérons des voisinages de petites tailles (exemple 64×64 pixels), pour le calcul des facteurs principaux, pour lesquels les deuxième et troisième facteurs principaux ne sont pas suffisamment stables (cf Figure 4.9).

Il ne faut donc pas séléctionner les facteurs principaux à retenir pour la segmentation uniquement sur un critère énergétique. Nous présenterons des résultats utilisant uniquement le premier facteur principal. Cela est relativement gênant dans la mesure où, si les axes principaux sont très proches en direction, il y aura une grande incertitude sur la segmentation finale, si on n'utilise que ce paramètre. Notre souhait serait donc de mettre en évidence une caractéristique discriminante, lorsque l'angle ne suffit plus. L'idée qui vient alors à l'esprit est de comparer les énergies dans les sous-espaces propres, en étudiant les valeurs propres.

Segmentation en utilisant uniquement l'aspect énergétique des textures

L'utilisation des valeurs propres comme caractéristiques n'a de sens que si les facteurs principaux associés aux différentes textures sont colinéaires. Rappelons que l'égalité des valeurs propres n'entraine pas l'existence d'un facteur principal commun. Nous allons vérifier plusieurs choses dans nos simulations :

- Utiliser les valeurs propres associées à des facteurs principaux instables en fonction de l'échelle n'est pas intéressant pour la segmentation.
- La normalisation de chaque valeur propre retenue par la somme des valeurs propres est source d'erreur car cette somme est instable en fonction des échelles.

Cette étude permettra de conclure qu'il n'est pas nécessaire de conserver beaucoup d'information énergétique sur chacune des textures pour effectuer une segmentation.
Nous insisterons par ailleurs sur le fait qu'un paramètre peut s'avèrer discriminant à une échelle et source d'erreurs à une autre.

Dans de nombreux articles sur les textures, il n'est pas fait mention du problème d'échelle, qui est pourtant central pour leur segmentation, par exemple, dans [81], la segmentation finale se fait par comparaison de caractéristiques locales avec des caractéristiques globales ; or, les caractéristiques ne sont valables qu'à une échelle donnée, pour laquelle il est toujours indispensable de vérifier la stabilité de la caractéristique retenue.
Si on s'intéresse au problème difficile de la segmentation des macro-textures, pour lesquelles l'échelle du calcul des caractéristiques est primordiale, les algorithmes de segmentation du type de celui proposé dans [81] seront inefficaces.

Dans notre étude, il nous apparaît que les valeurs propres associées aux facteurs principaux sont caractéristiques intéressantes, lorsqu'elles sont liées à des facteurs principaux stables et que ceux-ci sont proches des facteurs principaux associés aux autres textures présentes dans l'image.
La stabilité est liée à l'énergie contenue dans le sous-espace considéré et l'on rejoint ici l'analyse faite par [16], qui proposait de ne conserver, de leur décomposition en ondelettes que les composantes associées aux plus fortes énergies à une échelle.
La différence majeure entre notre approche et celle proposée dans [16] réside dans la détermination des espaces que l'on considère stables.
En effet, dans [16], les espaces les plus stables pour caractériser des textures correspondent aux espaces situés à la racine de l'arbre associé à la décomposition en paquets d'ondelettes et les plus instables se situent vers les feuilles. Si l'on décompose une image sur un banc de filtres, tel que celui que l'on utilise, les espaces les plus stables ne correspondent pas aux espaces proches de la racine qui contiennent peu d'énergie comparativement aux feuilles.

4.6.5 Résultats

Segmentation en utilisant uniquement des facteurs principaux

Nous présentons, dans cette section, deux types d'algorithmes, la caractéristique retenue étant le facteur principal associé à la plus grande valeur propre. Nous considérons un premier algorithme qui, partant de la répartition texturale obtenue à la fin de l'étape 2, modifie les frontières des textures. On définit les frontières des textures comme l'ensemble des points qui ont, parmi leur quatre plus proches voisins, deux voisins appartenant à des textures différentes [81].
On calcule alors les facteurs principaux en chaque point des frontières ; ces points sont, ensuite, affectés à la texture de référence, dont les caractéristiques ont été calculées lors de l'étape 2 et en utilisant la distance décrite à l'étape 1, sous réserve que l'on ne crée pas de point isolé, i.e, que dans la nouvelle image segmentée, un point ne puisse avoir tous ses plus proches voisins appartenant à des textures différentes de celle à laquelle il est affecté. On recalcule alors les caractéristiques des textures de référence et on itère le procédé jusqu'à stabilisation de l'image segmentée.
Au cours de cette phase, peuvent apparaître des "trous" dans la segmentation obtenue, qui sont

dûs à une mauvaise estimation des facteurs principaux. Le fait de considérer des voisinages relativement grands permet de rémédier à ce problème.
On présente les résultats de segmentation à différentes étape du processus en Figure 4.15. Pour

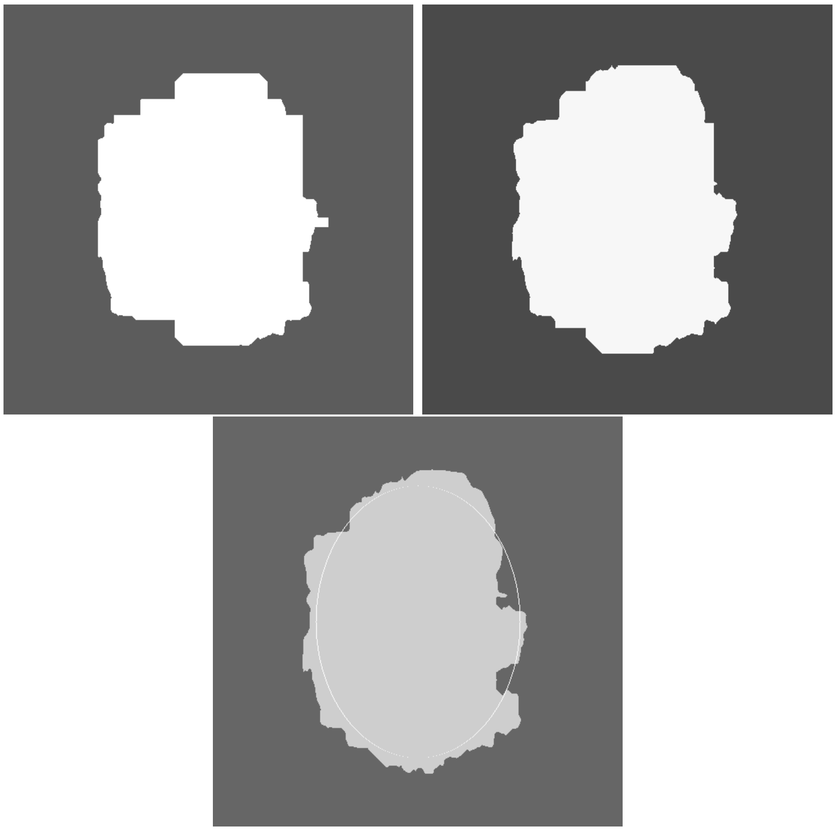

FIG. 4.15 – Résultats de l'algorithme au bout de 10, 20 et 30 itérations

limiter le coût en temps de calcul, il est possible de considérer un processus de segmentation un peu différent. On choisit un ensemble de points de référence disposés sur une grille cartésienne, en lesquels on calcule les facteurs principaux en se basant sur des voisinages de taille variable. Un point quelconque est ensuite affecté à la texture de référence la plus souvent reconnue au voisinage du point. Les résultats sont présentés en Figure 4.16.

Utilisation de critères énergétiques pour la segmentation

On entend par là que l'on va chercher à utiliser les valeurs propres dans le processus de segmentation.

- soit on considère une seule valeur propre (la première, associée à l'énergie la plus forte)
- soit plusieurs valeurs propres que l'on normalise par la somme des valeurs propres

Lorsque l'on utilise plusieurs valeurs propres, on peut déterminer le nombre de valeurs propres intéressantes à retenir en disant que l'on souhaite conserver un certain pourcentage α de l'énergie, i.e. on détermine la plus petite valeur de p telle que :

$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{k=1}^{3*J+1} \lambda_k} > \alpha.$$

Dans le premier cas, on note $p_{ref} = 1$ et $p_{ref} = p$, dans le second. On considère alors N points répartis sur une grille cartésienne, en lesquels on calcule les caractéristiques, puis on affecte

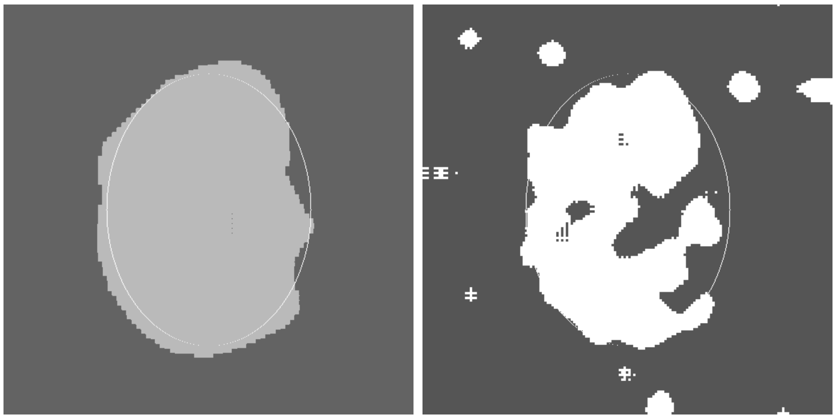

FIG. 4.16 – Résultats de segmentation en utilisant le premier facteur principal et en prenant un point sur 6 comme point de référence, les voisinages pour la recherche de la texture de référence sont des carrés de côté, à gauche, 64 pixels et, à droite, 32 pixels.

chacun des N points à la texture la plus proche. La distance utilisée lors de l'affectation est la suivante :

$$d(\beta, \beta_{ref}) = \sum_{k=1}^{k=p_{ref}} \frac{|\beta_k - \beta_k^{ref}|}{\beta_k + \beta_k^{ref}}$$

où β_k sont les caractéristiques retenues.
On affecte alors un point quelconque à la texture à laquelle les points de référence voisins ont été le plus souvent affectés.

On présente les résultats pour le second type de caractéristiques (les valeurs propres normalisées), en considérant une ou plusieurs caractéristiques, en utilisant des carrés de côté 64 pixels pour l'affectation des points de référence en Figure 4.17 et des carrés de taille 32 pixels en Figure 4.18.

FIG. 4.17 – Résultats de segmentation en considérant la première valeur propre normalisée et un point sur 6 comme point de référence; les voisinages pour la recherche des textures de référence sont des carrés de taille 64 pixels

On constate que l'augmentation du nombre de caractéristiques n'améliore pas la précision de la segmentation ; ceci est dû au fait que, dans notre cas, la première valeur propre est bien

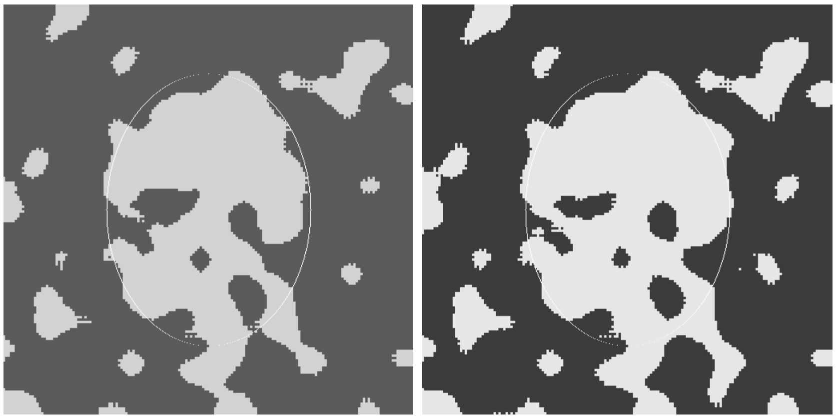

FIG. 4.18 – Résultats de segmentation en considérant la première valeur propre normalisée et un point sur 6 comme point de référence ; les voisinages pour la recherche des textures de référence sont des carrés de taille 32 pixels

discriminante pour chacune des deux textures, et que, par ailleurs, l'hypothèse de colinéarité des facteurs principaux n'est à peu près vérifiée que pour ceux associés à la plus grande valeur propre.
Par ailleurs, la segmentation n'empêche pas la formation de textures non connexes.
L'étude de la segmentation d'images en utilisant le premier type de caractéristiques (valeur propre seule), va nous permettre de donner un sens, en terme de valeurs propres, aux régions mal affectées lors de la segmentation utilisant les valeurs propres normalisées. En effet, si l'on utilise comme critère la première valeur propre sans aucune normalisation on obtient le résultat de segmentation de la Figure 4.19.

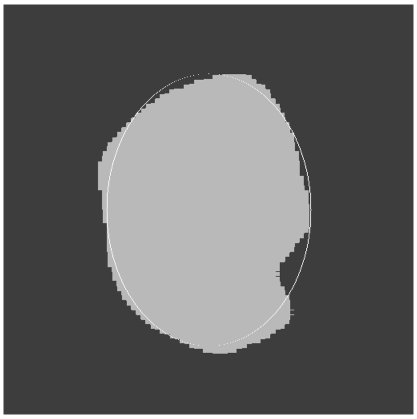

FIG. 4.19 – Segmentation en utilisant la première valeur propre sans normalisation. Les voisinages utilisés pour la recherche de la texture de référence sont des carrés de côté 64 pixels

Cette segmentation permet de mettre en évidence que les régions mal affectées, en utilisant la normalisation par la somme des valeurs propres, sont des régions qui ne vérifient pas la propriété que la somme des valeurs propres est constante à une même échelle. En fait, comme les résultats de segmentation sont corrects en utilisant uniquement la première valeur propre, on en déduit que la somme des $N-1$ dernières valeurs propres n'est pas constante.

4.6.6 Liens entre critère énergétique et facteurs principaux

Dans notre étude, nous avons remarqué que si l'on a un facteur principal commun à S et aux S_k associé à la première valeur propre, on a :

$$|I|\lambda_1 = |I_{k_0}| \sum_k \lambda_1^{(k)}$$

et si $\lambda_1^{(k)}$ est stable par rapport à k, alors les valeurs propres sont égales. Donc, l'étude des valeurs propres n'est intéressante que dans la mesure où il existe un facteur principal commun ; même si l'on a une valeur propre commune le facteur principal dépend des autres valeurs propres. On peut envisager de regrouper les informations concernant valeurs propres et facteurs principaux au sein d'une même distance que l'on définit par :

$$d((u,\lambda),(u^{ref},\lambda_{ref})) = (1 - |< u_1^{ref}, u_1 >|) \times \left(\frac{|\lambda_1 - \lambda_{1ref}|}{\lambda_1 + \lambda_{1ref}}\right)$$

Ainsi, on affecte un point à une texture, uniquement si le premier facteur principal et la première valeur propre coïncident. On visualise les résultats de segmentation en utilisant ce critère en Figure 4.20. Sur les exemples traités, les résultats obtenus ne sont pas meilleurs en utilisant le

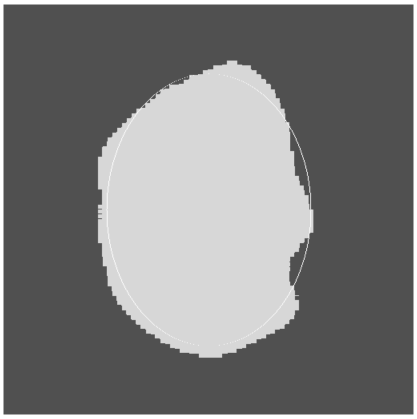

FIG. 4.20 – Segmentation en utilisant une distance incorporant le premier facteur principal et la valeur propre associée et en prenant un point sur 6 comme point de référence

premier facteur principal et la valeur propre associée. Bien que très proches pour les deux textures étudiées, les facteurs principaux associés à la première valeur propre suffisent pour effectuer une segmentation. Dans le cas où les facteurs principaux associés aux plus grandes valeurs propres sont rigoureusement colinéaires, l'étude des valeurs propres est nécessaire.

4.6.7 Intérêt des caractéristiques retenues

Le principal avantage de ces deux méthodes de segmentation est qu'elles ne font aucune hypothèse sur la nature statistique des données considérées.
Dans toutes les approches qui utilisent directement les coefficients, se pose le choix de la distance pour l'affectation des points , alors que, dans les méthodes proposées, on dispose d'une distance naturellement liée aux caractéristiques et ne faisant pas intervenir leurs ordres de grandeur.
Dans [95], une méthode fondée sur une décomposition en composantes principales a été utilisée,

mais la décomposition est dans ce cas globale, ce qui a plusieurs conséquences : si les textures testées contiennent des énergies importantes, il est possible que la première composante principale ne puisse pas bien discriminer les deux textures. En effet, c'est aussi celle de plus grande variance ; donc, si les moyennes sont proches la discrimination est difficile. Ensuite, l'analyse en composantes principales réalisée est liée à l'image de textures étudiée mais non spécifiquement à chacune des textures présentes dans l'image, contrairement à l'analyse locale qui permet d'extraire des caractéristiques intrinsèques à chaque texture.

Nous avons déjà développé deux méthodes de segmentation, sans que soit faite l'hypothèse gaussienne. Nous allons regarder une autre méthode classique utilisant les histogrammes et comparer les résultats avec les méthodes précédentes.

4.7 Segmentation non supervisée de textures à l'aide d'histogrammes

Dans cette section, on suppose que l'on dispose toujours, en tout point, du vecteur v à $3J+1$ composantes associé à la décomposition sur le banc de filtres de l'image I. Dans un premier temps, nous insistons sur la façon de construire les histogrammes. Puis nous étudions leur stabilité en fonction de la taille du voisinage servant à leur calcul. Après avoir fait une analyse critique des résultats obtenus, nous proposons un algorithme de segmentation non supervisée selon le même schéma que précédemment : découpage hiérarchique de l'image, regroupement en région et segmentation finale.

Une approche fondée sur l'utilisation d'histogrammes de coefficients issus de décomposition sur un banc de filtres a été utilisée par [43], mais en choisissant a priori les filtres à utiliser, dans ce cas des filtres de Gabor, et en ne précisant pas la construction pratique des histogrammes. L'approche par histogrammes présente l'avantage de préserver beaucoup plus d'information que les statistiques couramment utilisées (comme, par exemple, la variance des coefficients), son inconvénient principal étant que son estimation correcte nécessite beaucoup de points et un temps de calcul relativement long.
Nous présentons dans la section suivante une méthode de construction d'histogrammes.

4.7.1 Construction d'histogrammes à partir d'image de coefficients

On peut considérer que chaque composante de v est représentée par une image de coefficients. Soit $y[k,l]$ un coefficient d'une image associée à une composante et un voisinage V centré en $[k,l]$. Soit h un paramètre ; les points $y[k',l']$ de V vérifiant :

$$|y[k,l] - y[k',l']| < \frac{h}{2}$$

ont pour cardinal n_y, à partir duquel on définit une densité par :

$$f(y[k,l]) = \frac{n_y}{nh},$$

Où n est le nombre de points de V.

En pratique, cette construction est beaucoup trop coûteuse et nous choisissons a priori des points $y[k,l]$ sur l'image autour desquels on estime la densité.
Un moyen simple de déterminer des points convenables est de considérer l'écart-type empirique

de y sur l'image , $\sigma(I)$, puis de se restreindre à l'intervalle $[m - \alpha\sigma(I), m + \alpha\sigma(I)]$, où m est la moyenne de la composante, même si l'on sait que y n'est pas gaussienne. Les paramètres pour la construction des histogrammes sont donc h,α, le nombre de points $y[k,l]$ et leur position.
Plus h sera petit, meilleure sera la précision du calcul ; nous verrons, cependant, que considérer un paramètre h trop petit est pénalisant si le voisinage V est trop petit.

Une fois que l'on a déterminé un choix de paramètres pertinents pour le calcul de l'histogramme sur toute l'image, on peut en utilisant les mêmes paramètres définir des histogrammes locaux de coefficients.

4.7.2 Stabilité des histogrammes en fonction de l'échelle

Dans cette section, on considère que l'on a décomposé sur le banc de filtres une image de texture, qui donne naissance à un ensemble d'image de coefficients. On s'intéresse alors à une image, I, de coefficients de la décomposition, pour laquelle on calcule l'histogramme en utilisant k points, en fixant leur position et en donnant à m, α et h des valeurs particulières. On dispose donc d'un histogramme f caractéristique de l'image. En utilisant les mêmes paramètres on peut définir des histogrammes $\tilde{f}$ sur n'importe quelle sous-image de I. Nous menons une étude des paramètres de définition de l'histogramme en étudiant leurs effets sur la distance :

$$d(f,\tilde{f}) = \frac{1}{P}\sum_{i=1}^{i=P}\frac{|f(i)-\tilde{f}(i)|}{f(i)+\tilde{f}(i)}$$

Influence des paramètres sur la stabilité
Les images de coefficients à partir desquelles on calcule les histogrammes peuvent être les coefficients issus de la décomposition sur le banc de filtres ou ces derniers que l'on a décomposé en composantes principales.
Les autres degrés de liberté retenus sont le nombre de points de référence dans chaque intervalle $[m - \alpha\sigma(I), m + \alpha\sigma(I)]$, ainsi que leur position, ainsi que le choix de h.
Dans nos simulations, nous avons décidé de considérer uniquement des découpages de $[m - \alpha\sigma(I), m + \alpha\sigma(I)]$ en intervalles de taille égale, et d'estimer la densité de points sur chacun de ces intervalles, ce qui impose h et les points $y[k,l]$. Les résultats de l'analyse sont présentés en Figure 4.21 (pour l'analyse sans ACP), et en Figure 4.22 (pour l'analyse avec ACP)i
Pour les deux textures de référence étudiées plus haut, nous regardons comment évoluent les histogrammes des trois premières composantes, en calculant la distance moyenne d'un histogramme calculé sur une sous-image à l'histogramme de référence calculé sur l'image complète.

De cette étude, il ressort plusieurs choses. La première est que l'instabilité des histogrammes augmente avec la diminution de la taille des intervalles de calcul (augmentation du nombre des intervalles). Ensuite, les axes associés aux plus grandes énergies ne sont pas les plus stables, ce qui altère leur pouvoir discriminant.

4.7.3 Discussion de la pertinence de l'utilisation d'histogramme de coefficients pour la segmentation non supervisée d'images texturées

Nous avons essayé d'appliquer une procédure de segmentation non supervisée identique à celle présentée en début de chapitre, i.e, découpage hiérarchique en utilisant les histogrammes de coefficients, regroupement en régions en utilisant la distance entre histogrammes introduite à la

composante 1	256	128	64	32
2 intervalles	0.02	0.06	0.11	0.17
5 intervalles	0.05	0.12	0.24	0.39

composante 1	256	128	64	32
2 intervalles	0.05	0.07	0.08	0.11
5 intervalles	0.09	0.14	0.20	0.31

composante 2	256	128	64	32
2 intervalles	0.005	0.01	0.02	0.039
5 intervalles	0.02	0.06	0.11	0.20

composante 2	256	128	64	32
2 intervalles	0.0065	0.01	0.017	0.02
5 intervalles	0.02	0.05	0.08	0.17

composante 3	256	128	64	32
2 intervalles	0.006	0.017	0.02	0.04
5 intervalles	0.02	0.05	0.11	0.20

composante 3	256	128	64	32
2 intervalles	0.007	0.013	0.016	0.03
5 intervalles	0.03	0.06	0.09	0.17

FIG. 4.21 – Distance moyenne de l'histogramme calculé sur des sous-images à l'histogramme de référence calculé sur l'image complète, en fonction de la taille de la sous-image, pour la texture Misc en colonne de gauche et Rock en colonne de droite

composante 1	256	128	64	32
2 intervalles	0.02	0.06	0.10	0.16
5 intervalles	0.05	0.12	0.24	0.38

composante 1	256	128	64	32
2 intervalles	0.04	0.05	0.06	0.08
5 intervalles	0.08	0.12	0.18	0.28

composante 2	256	128	64	32
2 intervalles	0.001	0.03	0.06	0.08
5 intervalles	0.03	0.07	0.13	0.2

composante 2	256	128	64	32
2 intervalles	0.03	0.04	0.05	0.07
5 intervalles	0.05	0.09	0.1	0.17

composante 3	256	128	64	32
2 intervalles	0.005	0.013	0.023	0.04
5 intervalles	0.02	0.04	0.09	0.15

composante 3	256	128	64	32
2 intervalles	0.01	0.02	0.029	0.044
5 intervalles	0.02	0.004	0.069	0.13

FIG. 4.22 – Distance moyenne de l'histogramme calculé sur des sous-images à l'histogramme de référence calculé sur l'image complète, en fonction de la taille de la sous-image, pour la texture Misc en colonne de gauche et Rock en colonne de droite, les coefficients étant issus d'une analyse en composantes principales

section précédente, puis segmentation finale, mais les résultats de l'étape de regroupement sont toujours très médiocres.
La raison essentielle des mauvais résultats obtenus est le nombre très important de caractéristiques de texture générées, par cette approche par histogrammes.

Depuis le début de notre étude, nous avons délibéremment choisi d'analyser les textures par décomposition sur un banc de filtres, la construction d'histogrammes à partir des coefficients est trop instable pour être utiliser directement.

Nous allons proposer un nouvel algorithme qui n'utilise pas les mêmes caractéristiques pour les étapes de découpage hiérarchique de l'image et de regroupement en régions d'une part, et pour la segmentation finale d'autre part.

4.7.4 Histogrammes de niveaux de gris et découpage hiérarchique

En étudiant les textures précédentes, on remarque que les histogrammes de niveaux de gris sont relativement stables (Figure 4.23).
Ces histogrammes présentent l'avantage d'être faciles à calculer, et de fournir un faible nombre de

	256	128	64	32
2 intervalles	0.025	0.05	0.1	0.15
5 intervalles	0.028	0.06	0.12	0.2

	256	128	64	32
2 intervalles	0.04	0.06	0.07	0.1
5 intervalles	0.07	0.16	0.25	0.31

FIG. 4.23 – Distance moyenne de l'histogramme des niveaux de gris calculé sur des sous-images à l'histogramme des niveaux de gris calculé sur l'image complète, pour la texture Misc en colonne de droite et Rock en colonne de gauche

caractéristiques . Par ailleurs, on constate toujours que la multiplication du nombre d'intervalles est un facteur d'instabilité (cf Figure 4.23).

On considère maintenant l'image de deux textures que nous avons déjà étudié au début du chapitre. Nous allons voir comment procéder à un découpage hiérarchique puis à un regroupement en régions en utilisant comme caractéristiques de textures l'histogramme des niveaux de gris. Pour le découpage hiérarchique on applique rigoureusement la méthode de la section 4.4.4 et on obtient le regroupement de la Figure 4.24.

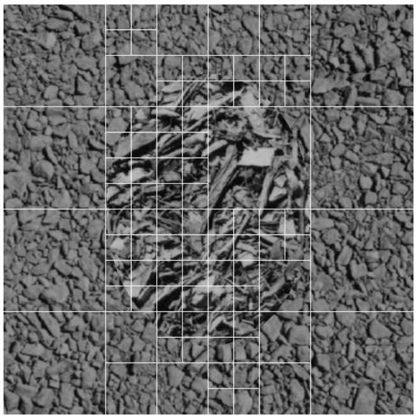

FIG. 4.24 – Découpage hiérarchique en utilisant les histogrammes des niveaux de gris, en effectuant un découpage de l'intervalle des niveaux de gris de l'image en 3 sous-intervalles, et en considérant un seuil de 0.2 pour l'arrêt du découpage

Si nous utilisons une procédure de regroupement analogue à celle utilisée dans le cas de la décomposition en utilisant les facteurs principaux, étant donné que les fluctuations des niveaux de gris sont trop importants sur des petits voisinages, le regroupement est inefficace.
Si l'on veut pouvoir estimer correctement les régions, il faut définir un nouveau processus de regroupement, en tenant compte des phénomènes d'échelle.
Quand une region est "petite" elle est regroupée à sa voisine la plus proche dont la taille est la plus proche de la sienne. Lorsque les régions ainsi regroupées atteignent une taille critique, pour laquelle les histogrammes de niveaux de gris sont jugés fiables, les regroupements se font en fonction des histogrammes et non de la taille des régions considérées.
On peut alors construire la distance suivante, qui tient compte de ces phénomènes d'échelle. Soit D une région, et soit $V(D)$ l'ensemble des régions voisines de D et , $taille_{max}$, la taille critique du voisinage au delà de laquelle on estime que les histogrammes de niveaux de gris sont des caractéristiques fiables. On peut alors formuler le problème de la manière suivante (T signifiant le taille) :

Si pour tout $D' \in V(D)$, $T(D') \geq taille_{max}$ et $T(D) \geq taille_{max}$

$$d(f,f') = {taille_{max}}^2 \frac{1}{P} \sum_{i=1}^{i=P} \frac{|f(i) - f'(i)|}{f(i) + f'(i)}$$

Sinon, si par exemple $T(D) \geq taille_{max}$ et $T(D') < taille_{max}$ ou inversement, si, par exemple, $T(D) < taille_{max}$ et $T(D') \geq taille_{max}$, nous avons :

$$d(f,f') = \min(T(D), T(D'))taille_{max} \frac{1}{P} \sum_{i=1}^{i=P} \frac{|f(i) - f'(i)|}{f(i) + f'(i)}$$

et, si $T(D) \leq taille_{max}$ et $T(D') \leq taille_{max}$:

$$d(f,f') = \min(T(D), T(D'))^2 \frac{1}{P} \sum_{i=1}^{i=P} \frac{|f(i) - f'(i)|}{f(i) + f'(i)}$$

Avec un tel algorithme, nous regroupons les zones de taille inférieure à $taille_{max}$ entre elles, puis les zones de taille inférieure à $taille_{max}$ restantes avec les zones de taille supérieure à $taille_{max}$ les plus proches, et, enfin, les zones de taille supérieure à $taille_{max}$. Les résultats du regroupement apparaissent en Figure 4.25.

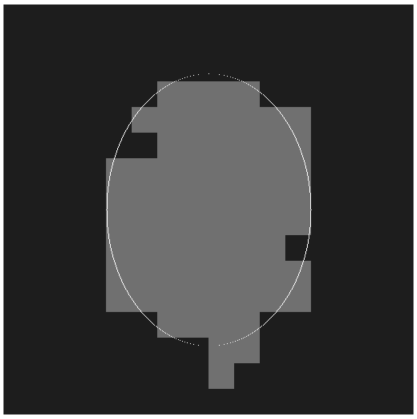

FIG. 4.25 – Résultat du regroupement pour un paramètre $taille_{max} = 64 \times 64$ pixels

On constate donc que la procédure de regoupement des régions doit être adaptée au type de caractéristiques retenues. La distance proposée est particulièrement bien adaptée au regroupement des régions, contenant des macro-textures dont les caractéristiques fluctuent beaucoup à petite échelle. Une fois que l'on a une estimation à peu près correcte des régions, il est alors possible de procéder à la segmentation finale.

4.7.5 Segmentation en utilisant des histogrammes de coefficients

Nous allons voir que les histogrammes de niveaux de gris, ne peuvent permettre une segmentation correcte des textures ; nous montrerons que l'on peut utiliser alors les histogrammes des coefficients de la décomposition sur le banc de filtres pour effectuer l'étape de segmentation.

Nous partons donc du regroupement en textures obtenu à la section précédente, et calculons les histogrammes de niveaux de gris de référence associés à chaque région ainsi définie et les histogrammes de niveaux de gris en des points situés sur une grille régulière de l'image. On affecte à ces derniers un label en fonction de l'histogramme de référence le plus proche. Les autres points de l'image sont ensuite affectés du label le plus souvent apparu dans le voisinage du point considéré. La Figure 4.26 montre les résultats de la segmentation en utilisant une densité de points variables.

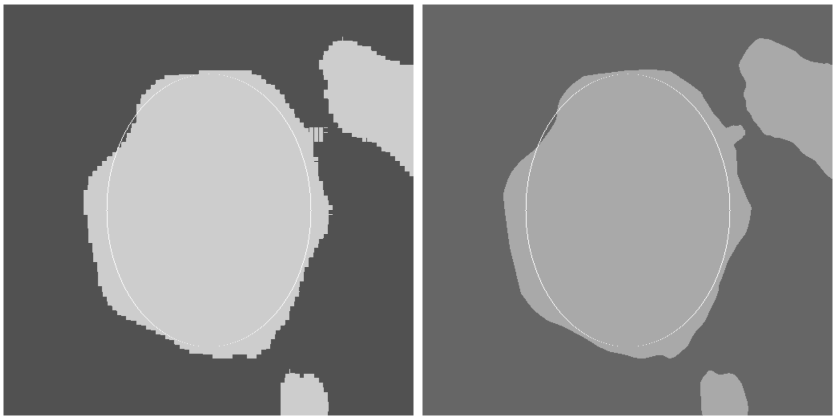

FIG. 4.26 – segmentation de l'image en utilisant des histogrammes de niveaux de gris calculés sur des sous-images de taille 64×64 pixels ; le nombre de sous-intervalles pour la construction des histogrammes est 3 sur l'image de gauche ; on considère un point sur 6 dans chaque direction comme point de référence pour l'image de gauche et tous les points dans l'image de droite

On remarque que, même si l'on considère tous les points, il demeure toujours principalement des régions mal affectées. L'utilisation d'histogrammes de niveaux de gris pour segmenter les macro-textures n'apparaît pas pertinent.
Nous allons montrer que l'utilisation des histogrammes issus des coefficients de la décomposition sur le banc de filtres permet la segmentation correcte des textures.

Etant donné que le nombre de caractéristiques issues des histogrammes des coefficients de la décomposition est nettement plus élevé que dans le cas des histogrammes de niveaux de gris précédents, nous allons réduire leur nombre. Pour cela, sur les régions obtenues après la phase de regroupement, nous effectuons une analyse en composantes principales, et nous décidons de ne conserver que les espaces contenant le plus d'énergie, en appliquant le critère utilisant les valeurs propres pour déterminer les espaces à conserver. Ayant calculé les histogrammes de référence sur les espaces contenant le plus d'énergie, pour chaque région, nous définissons la procédure d'affectation d'un point à une texture.

Pour savoir si un point appartient à une texture, le principe de l'algorithme est alors de projeter les vecteurs de coefficients en chaque point d'un voisinage du point considéré sur les bases des facteurs principaux associées à chaque région, puis de calculer les histogrammes des projections sur le voisinage. On compare, alors, les histogrammes obtenus à ceux des régions de référence, et l'on affecte le point à la région la plus proche.

Pour diminuer le temps de calcul, on utilise la même technique que précédemment en considérant un certain nombre de points de référence, on donne un label aux autres points en fonction du comportement des points de référence situés dans leur voisinage 4.27.

Dans cette étude, nous avons constaté un phénomène intéressant, à savoir qu'il n'est pas forcément judicieux d'utiliser les mêmes caractéristiques pour le découpage initial de l'image et pour la reconstruction. En effet, pour le découpage de l'image, qui permet d'avoir un aperçu

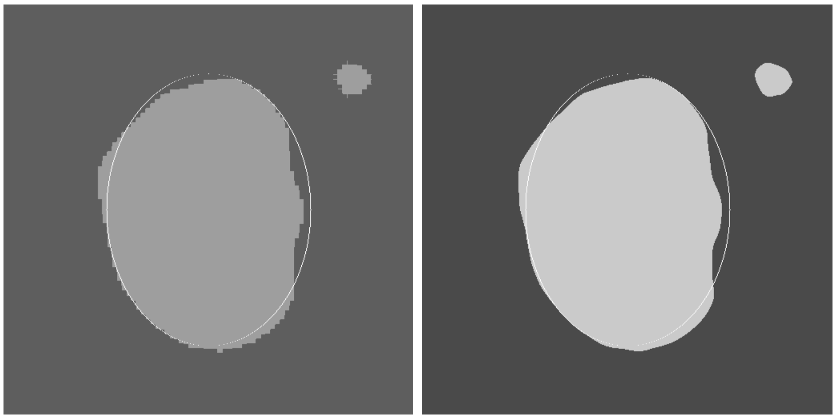

FIG. 4.27 – On considère les espaces propres tels que $\alpha = 0.9$, et l'on divise ensuite chaque intervalle en 3 sous-intervalles, l'image de gauche présente les résultats en prenant un point sur 6 comme point de référence, et celle de droite en considérant tous les points

grossier du contenu textural, nous avons vu que l'utilisation des coefficients de la décomposition sur le banc de filtres est peu fiable.

En revanche, lorsque l'on dispose d'une estimation des textures de l'image, la sélection des espaces contenant le plus d'énergie par le biais d'une analyse en composantes principales puis construction d'histogrammes associés permet d'obtenir une segmentation fiable.

Comme lors de l'étude que nous avions faite des facteurs principaux et des valeurs propres, nous constatons que sélectionner les espaces à conserver sur le critère des valeurs propres (cf section 4.4.5) n'est pas adapté.
En effet, on remarque toujours qu'une région de petite taille est mal affectée. Quelque soit la précision utilisée lors de l'affectation (i.e, le nombre de points de référence dans chacune des directions), cette zone demeure mal affectée.
Cette région correspond sensiblement à la région mal reconnue en considérant la normalisation par la somme des valeurs propres. Cela laisse supposer que cette mauvaise affectation est due au fait que les composantes sur certains espaces propres associés aux valeurs propres différentes de λ_1 fluctuent beaucoup. C'est ce que nous vérifions en Figure 4.28. Certaines composantes qui ont une signification en terme énergétique pour la texture peuvent ne pas être discriminantes pour la segmentation.

Nous allons juste brièvement présenter deux méthodes alternatives de segmentation avant de conclure.

4.8 Approche fractale pour la segmentation des textures

Soit S un ensemble borné de $\mathbb{R}^n$, soit $N(s)$ le nombre minimal de boules de rayon s nécessaire pour recouvrir S. Si S est un objet fractal, objet que l'on définit mathématiquement sous la forme d'ensemble auto-similaire, et si D est sa dimension, on a alors la relation :

$$N(s) \sim s^{-D},$$

ce que l'on réecrit sous la forme :

$$D = -\lim_{s \to 0} \frac{log(N(s))}{log(s)}$$

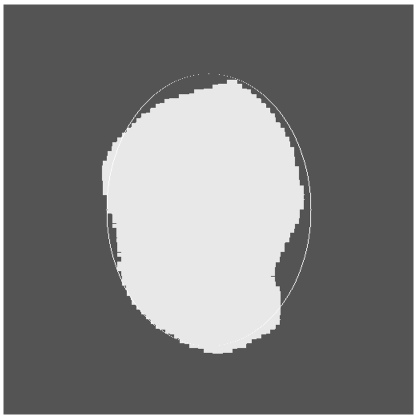

FIG. 4.28 – segmentation en utilisant uniquement l'histogramme associé à la valeur propre la plus grande

Dans [18], les caractéristiques de textures extraites sont les dimensions fractales de la texture, de l'image des niveaux de gris les plus hauts et les plus bas, et des images lissées dans le sens des colonnes et des lignes respectivement. La multiplication des caractéristiques est nécessaire dans la mesure où deux ensembles auto-similaires peuvent avoir la même dimension fractale. Un phénomène plus important est que les textures ne sont pas forcément des ensembles auto-similaires, et l'on peut améliorer les résultats en considérant les dimensions multi-fractales des ensembles [18] [77].
Une autre méthode consiste à estimer le spectre de singularités. Soit I une image ; pour chaque valeur de régularité Lipschitzienne α (voir chapitre suivant pour la définition de la régularité Lipschitzienne), on détermine la dimension fractale des ensembles sur lesquels l'image a une régularité α. On obtient, ainsi, une fonction $D(\alpha)$ qui caractérise les singularités présentes dans l'image, le principal obstacle à son utilisation pour la segmentation des textures étant un coût de calcul très élevé.
Une autre approche liée aux fractales consiste à considérer les textures comme des réalisations de processus auto-similaires. Pour plus de détails, voir [55].

4.8.1 Méthodes variationnelles

Les méthodes variationnelles sont des méthodes qui envisagent le problème de segmentation sous la forme d'une minimisation d'énergie dont la forme peut s'écrire :

$$E(u,K) = \int_{\Omega/\ K} (u-g)^2 dx + \lambda l(K),$$

où K est une frontière et, $l(K)$ la longueur de cette frontière. Il est à noter que, si l'on se donne une frontière fermée K, alors Ω est composé d'éléments connectés O_i et la minimisation globale revient à effectuer une minimisation sur chacun des éléments O_i, i.e, $u = \frac{1}{O_i}\int_{O_i} g(x)dx$. Donc, la minimisation équivaut à la recherche d'une frontière minimisant une énergie (pour plus de détails et des résultats de segmentation, voir [72]).

4.9 Conclusion

Nous avons étudié les textures du point de vue de leur structure et nous avons écrit de nouveaux algorithmes de segmentation non supervisée d'images texturées. Nous avons mis en évidence le rôle des valeurs propres et des facteurs principaux obtenus grâce à une analyse en composantes principales.
Cette étude nous a permis de montrer que la stabilité des caractéristiques en fonction des échelles est primordiale pour la segmentation. L'étude par histogrammes que nous avons faite ensuite est intéressante, dans la mesure où elle offre une autre caractérisation des textures, et que les résultats obtenus sont relativement similaires. Par ailleurs, cela permet de constater que les sous-espaces discriminants en utilisant les facteurs principaux et les valeurs propres le sont aussi lorsque l'on considère les histogrammes.

L'intérêt de l'étude porte aussi sur la constatation que l'on n'a pas besoin de connaître complétement le contenu énergétique des textures pour effectuer une segmentation, et sélectionner les paramètres en fonction de critères énergétiques n'est pas judicieux. Pour la segmentation à l'aide de facteurs principaux, sélectionner les caractéristiques sur un critère de stabilité le long des échelles nous est apparu pertinent. En revanche lorsque l'on utilise des histogrammmes, un critère de sélection pertinent reste à définir.

Dans la partie suivante, nous allons nous intéresser à l'étude de la régularité Lipschitzienne des textures, puis nous essaierons de comprendre la nature des bords des textures.

5

Etude de la régularité Lipschitzienne des textures et de leurs bords

Dans ce chapitre, on se propose d'étudier la régularité des textures, en faisant l'hypothèse que celles-ci sont des fonctions Lipschitziennes α, dont nous donnerons au préalable une définition. Nous verrons que cette hypothèse n'est pas valide pour caractériser les textures gaussiennes (i.e, les textures qui sont la réalisation d'un processus gaussien), mais peut en revanche être plus facilement utilisée pour la caractérisation de macro-textures contenant des bords significatifs ; nous verrons, néanmoins, que ce critère est plus instable que ceux précédemment utilisés.
Par ailleurs, nous construirons un algorithme de détermination des bords des textures, qui bien que partiel, donnera toutefois des informations intéressantes sur leur nature.

5.1 Définitions

Dans cette section, nous allons rappeler quelques définitions dont on se servira par la suite :

Définition 1 *Une fonction f est Lipschitzienne $\alpha \geq 0$ en ν, s'il existe $K > 0$, et un polynôme de degré $m = E(\alpha)$, où E désigne la partie entière, tel que :*

$$\forall t \in \mathbb{R} \quad |f(t) - P_\nu(t)| \leq K|t - \nu|^\alpha \quad (1)$$

Une fonction f est uniformément Lipschitzienne α sur $[a, b]$, si elle satisfait (1) pour tout $\nu \in [a, b]$, avec K indépendant de ν.
La régularité Lipschitzienne de f en ν ou sur $[a, b]$ est le sup des α tel que f est lipschitzienne α.

Remarques :
- Dans le cas où la fonction f est Lipschitzienne en ν avec $0 \leq \alpha < 1$, alors la condition (1) se simplifie en :
$$\forall t \in \mathbb{R}, |f(t) - f(\nu)| \leq K|t - \nu|^\alpha$$
- Dans le cas ou f est uniformément Lipschitzienne $\alpha > m$ alors f est de classe C^m.

On peut par ailleurs, définir la régularité Lipschitzienne des distributions de la manière suivante :

Définition 2 *Une distribution admet comme régularité Lipschitzienne α sur $[a, b]$ si sa primitive admet comme régularité Lipschitzienne $\alpha + 1$ sur $[a, b]$*

Enfin on définit la régularité Lipschitzienne d'une fonction f en dimension 2 par :

Définition 3 *Une fonction f est uniformément Lipschitzienne d'ordre $0 < \alpha < 1$ sur $[a,b] \times [c,d]$ si :*

$$\forall (x_1, y_1)(\nu_1, \nu_2) \in [a,b] \times [c,d] \frac{|f(x_1,x_2) - f(\nu_1,\nu_2)|}{\|(x_1-\nu_1, x_2-\nu_2)\|^\alpha} < C$$

5.2 Théorèmes de régularité et ondelettes

5.2.1 Théorème de régularité en dimension 1

Nous donnons ici un théorème caractérisant les fonctions uniformément Lipschitziennes en dimension 1.

Théorème 1 *Soit Ψ une ondelette de classe C^n, si $f \in L^2(\mathbb{R})$ est uniformément Lipschitzienne d'ordre $\alpha \leq n$ sur $[a,b]$ et, si l'on définit :*

$$Wf(u,s) = \int_{\mathbb{R}} f(t) \frac{1}{\sqrt{s}} \Psi(\frac{t-u}{s}) dt,$$

alors, il existe $A > 0$ tel que :

$$\forall (u,s) \in [a,b] \times \mathbb{R}^+, |Wf(u,s)| \leq A s^{\alpha + \frac{1}{2}}$$

Réciproquement, supposons que f soit bornée et que $Wf(u,s)$ satisfasse l'inégalité précédente pour une valeur $\alpha < n$, qui n'est pas entière, alors f est uniformément Lipschitzienne sur $[a+\epsilon, b-\epsilon]$, pour tout ϵ

5.2.2 Théorème de régularité en dimension 2

Nous allons voir comment s'étendent en dimension 2 les propriétés précédentes. Nous nous plaçons directement dans le cadre des échelles dyadiques $s = 2^j$ utilisées en traitement d'images.

Nous proposons ici deux approches de calcul de régularité Lipschitzienne. La première a été développée par Mallat [65], en utilisant des ondelettes splines et la seconde se place dans le cadre plus général d'ondelettes quelconques obtenus par analyse multirésolution.

- **Algorithme de Mallat :**
  L'étude de Mallat étant destinée à la caractérisation des bords des images, l'approche est voisine de celle de Canny [10].
  Le principe est que les bords d'une image f, définie comme une fonction de $\mathbb{R}^2$ dans $\mathbb{R}$, sont obtenus comme maxima de la norme du gradient de f, dans la direction de celui-ci. Mallat considère, dans un premier temps, la convolution de f avec θ appartenant à l'ensemble des fonctions lisses définies comme l'ensemble des fonctions d'intégrale égale à 1 sur $\mathbb{R}^2$ et qui tendent vers 0 à l'infini. On obtient alors une nouvelle fonction $\tilde{f} = f * \theta$. On cherche ensuite les bords de cette nouvelle image $\tilde{f}$. Utilisant l'algorithme de Canny, il détecte les bords en calculant les maxima de $\|\nabla \tilde{f}\|$, dans la direction de $\nabla \tilde{f}$, définis par :

  $$\nabla \tilde{f} = \begin{pmatrix} \frac{\partial f * \theta}{\partial x} \\ \frac{\partial f * \theta}{\partial y} \end{pmatrix} = \begin{pmatrix} f * \frac{\partial \theta}{\partial x} \\ f * \frac{\partial \theta}{\partial y} \end{pmatrix}$$

  où θ est une fonction lisse, donc $\Psi_1(x,y) = \frac{\partial \theta(x,y)}{\partial x}$ et $\Psi_2(x,y) = \frac{\partial \theta(x,y)}{\partial y}$ sont des ondelettes. Ainsi, les bords de l'image sont caractérisés par les maxima de $Mf(x,y,s) =$

$\sqrt{W^1f(x,y,s)^2 + W^2f(x,y,s)^2}$ dans la direction du vecteur $(W^1f(x,y,s), W^2f(x,y,s))$ où :

$$W^if(x,y,s) = \int\int f(u,v)\frac{1}{s}\Psi_i(\frac{x-u}{s}, \frac{y-v}{s})dudv$$

Soit $0 < \alpha < 1$, considérons des ondelettes Ψ_1 et Ψ_2 telles que celles-ci soient Lipschitziennes d'ordre supérieur strictement à α et telles que :

$$\int\int \|(x,y)\|^\alpha \Psi_i(x,y) < \infty \quad \text{pour } i \in \{1,2\}$$

alors on a l'équivalence : f est Lipschitzienne d'ordre α si et seulement si

$$Mf(x,y,2^j) \leq A2^{j(\alpha+1)} \quad \text{pour tout } j \in \mathbb{Z}$$

Mallat [64] a montré que ces résultats s'étendent au cas où f est une distribution. Nous allons vérifier numériquement cette propriété dans la section suivante.
Considérons une base d'ondelettes 2D obtenue par analyse multirésolution 1D, puis par produit tensoriel et l'on suppose, de plus, que l'ondelette Ψ et la fonction échelle Φ sont à support compact. Considérons les produits scalaires de f, à l'échelle j, avec les fonctions suivantes :

$$(2^{-j}\Psi(2^{-j}(x-k))\Phi(2^{-j}(y-p)), 2^{-j}\Phi(2^{-j}(x-k))\Psi(2^{-j}(y-p)), 2^{-j}\Psi(2^{-j}(x-k))\Psi(2^{-j}(y-p)))_{(p,k)\in\mathbb{Z}^2},$$

Si f est uniformément Lipschitzienne $0 < \alpha < 1$ sur $[a,b] \times [c,d]$, en posant :

$$Nf(k,p,2^j) =$$

$$\frac{1}{2^j}\sqrt{< f(x,y), \Psi(2^{-j}(x-k))\Phi(2^{-j}(y-p)) >^2 + < f(x,y), \Phi(2^{-j}(x-k))\Psi(2^{-j}(y-p)) >^2}$$

nous avons la propriété suivante :

$$Nf(k,p,2^j) \leq C2^{j(1+\alpha)}$$

Cette majoration permet de caractériser la régularité Lipschitzienne de f si $\Phi\Psi$ est suffisamment régulière (Lipschitzienne d'ordre strictement supérieur à α).

5.3 Régularité Lipschitzienne d'images tests

Nous allons étudier la régularité Lipschitzienne de deux types de singularités, un échelon, que l'on représente par une image à moitié blanche et à moitié noire et un Dirac, que l'on représente comme un point noir au milieu d'une image blanche. Ces singularités correspondent à une reégularité Lipschitzienne égale à -1 pour le Dirac et 0 pour la fonction échelon.

Pour cela, nous étudions le comportement des coefficients $Nf(x,y,2^j)$, en fonction de j, puis de $Mf(x,y,2^j)$.

- **Les ondelettes de Daubechies sur-échantillonnées :**
  Pour le calcul de $Nf(x,y,2^j)$, on utilise des ondelettes de Daubechies ainsi que les fonctions échelles associées. Le calcul de la régularité Lipschitzienne s'effectue en étudiant la décroissance du maximum des coefficients $Nf(x,y,2^j)$ en fonction des échelles, par une méthode de régression linéaire.

$$Nf(x,y,2^j) \leq A2^{j(1+\alpha)} \Rightarrow \log_2(Nf(x,y,2^j)) \leq j(1+\alpha)$$

Si l'on considère une échelle logarithmique, la décroissance du maximum des coefficients doit s'effectuer selon une droite de pente $1+\alpha$. Nous faisons l'hypothèse que que $\log_2(Nf(x,y,2^j))$ suit une trajectoire rectiligne aux trois premières échelles échelles dyadiques et nous considérons pour le calcul de α la droite de régression associée. La valeur trouvée n'est pas conforme à la théorie. En effet, une ondelette continue devrait suffire à l'estimation correcte des régularité dans les deux cas étudiés, i.e. D4 devrait donner de bons résultats (voir Figure 5.1). La raison d'un tel écart est liée au fait que $\log_2(Nf(x,y,2^j))$ ne suit pas une trajectoire rectiligne aux échelles considérées.

	D4	D8	D12	D20
régularité Lipschitzienne	-0.78	-0.79	-0.95	-0.97

	D4	D8	D12	D20
régularité Lipschitzienne	-0.03	-0.11	-0.02	-0.09

FIG. 5.1 – Calcul de la régularité Lipschitzienne pour un Dirac (en haut) et une fonction échelon (en bas), en utilisant les ondelettes de Daubechies (D4 indique deux moments nuls, D8 indique 4 moments nuls et ainsi de suite)

- **Les ondelettes splines :**
  On étudie ici le comportement des ondelettes utilisées dans [65], qui sont des ondelettes splines quadratiques à support compact, en appliquant rigoureusement l'algorithme proposé, notamment concernant les constantes de normalisation. Pour un Dirac, on obtient une régularité Lipschitzienne de -0.999124 et, pour un échelon, 0.000875. L'avantage de telles ondelettes est qu'elles permettent l'estimation correcte de la régularité Lipschitzienne des distributions, tout en ayant un support limité.

5.4 Régularité Lipschitzienne des textures

Dans cette section, nous allons étudier la régularité Lipschitzienne de différentes textures en utilisant les ondelettes splines quadratiques introduites par Mallat dans [65].
Nous voulons estimer la régularité Lipschitzienne uniforme des textures ainsi que la stabilité de cette régularité en fonction de la taille du voisinage de calcul considéré.

Pour effectuer une telle mesure, nous calculons les coefficients $Mf(k,p,2^j)$ en tout point de l'image de texture dont on souhaite calculer la régularité, pour plusieurs valeurs de j, puis, on découpe l'image de texture en sous-images disjointes carrées de taille égale. Sur chacune de ces sous-images, nous calculons la régularité Lipschitzienne uniforme en utilisant les coefficients $Mf(k,p,2^j)$. On calcule alors la moyenne des régularités trouvées ainsi que leur écart-type, pour une taille de sous-images donnée, et l'on compare les résultats avec la régularité obtenue sur toute l'image.

Ce type d'approche n'est intéressant que s'il permet de caractériser convenablement les textures, i.e, de voir si la régularité est à la fois un facteur stable en fonction de la taille de la sous-image considérée et discriminant par rapport à d'autres textures. Le tableau de la Figure 5.2 présente les résultats obtenus. Cette étude permet de constater que les textures qui se rapprochent le plus des textures gaussiennes (Asphalt et Grass) ne peuvent être caractérisées par leur régularité Lipschitzienne. Des outils statistiques simples permettent de mieux les caractériser.
Ce qui nous intéresse surtout est l'étude des macro-textures et, en particulier, les textures Rock

	512	128	64	32	16	8
Misc	0.29	(0.32,0.03)	(0.31,0.07)	(0.29,0.1)	(0.28,0.13)	(0.29,0.22)
Rock	0.28	(0.23,0.05)	(0.22,0.06)	(0.20,0.09)	(0.17,0.15)	(0.20,0.25)
Asphalt	0.07	(0.02,0.12)	(0.004,0.11)	(-0.015,0.14)	(-0.03,0.16)	(-0.02,0.22)
Grass	0.09	(0.06,0.06)	(0.06,0.07)	(0.06,0.10)	(0.03,0.13)	(0.01,0.19)

FIG. 5.2 – Régularité Lipschitzienne calculée sur chaque sous-image, en étudiant la décroissance des maxima en fonction des échelles, on représente la moyenne et l'écart-type empiriques des régularités Lipschitziennes trouvées pour différentes taille de sous-images (en entrée du tableau)

et Misc pour lesquelles nous avons des éléments de comparaison. On remarque que, pour ces textures, la régularité Lipschitzienne est très instable pour des tailles de sous-images inférieures à 64×64 pixels, la comparaison est, dans ces cas, difficile entre les textures Rock et Misc. Nous pensons que l'étude du spectre des singularités doit permettre, au prix d'un temps de calcul beaucoup plus long, une caractérisation plus fine des textures : c'est l'objet de l'analyse multifractale.

5.5 Etude des frontières des textures

Mis à part des cas très particuliers, les frontières de textures n'ont pas de régularité Lipschitzienne bien définie ; c'est pourquoi nous n'allons pas utiliser ce paramètre et privilégier des considérations géométriques. Le point de vue qui consiste à considérer la segmentation des textures par l'étude de leurs frontières à été adopté dans [107].
Le principe de l'algorithme développé dans cet article est de déterminer une localisation approximative des frontières des textures. A partir de ces frontières approximatives, on peut alors définir une zone frontière, dans laquelle on va chercher la vraie frontière, de la manière suivante :

$$B = \{x/d(x,p) < D \quad , p \in L \quad x \in I\},$$

où L est la frontière à l'échelle considérée et I l'image. La valeur $2 * D$ correspond au diamètre du voisinage utilisé pour le calcul.
L'étude servant à la localisation exacte du bord est fondée sur la détection de bords, que l'on classe en deux catégories : la première, les frontières des textures et la seconde, les bords intérieurs aux textures. Ces catégories sont déterminées en utilisant des propriétés du niveau de gris de l'image.
Nous allons voir comment les ondelettes permettent d'obtenir une approximation des bords, nous ne ferons pas la distinction entre bord et frontière. Pour cela, nous allons déterminer les maxima locaux du module de la transformée en ondelettes dans la direction du gradient de l'image filtrée, selon l'algorithme introduit par Mallat, et former les chaînes de maxima qui sont significatives, selon un critère que nous allons préciser.

5.5.1 Principe de l'algorithme

Nous allons donner une nouvelle définition des bords des images, mais avant cela précisons que les bords vont être cherchés parmi les maxima locaux de $Mf(x, y, 2^j)$ dans la direction du vecteur $Wf(x, y, 2^j) = (W^1f(x, y, 2^j), W^2f(x, y, 2^j))$.
En Figure 5.3, on représente une image de quatre textures [9], l'image des coefficients $Mf(k, l, 2)$ et enfin l'image des maxima dans la direction de $Wf(k, l, 2)$; les détails de construction seront

présisés dans la section "mise en place de l'algorithme".
Nous voyons déjà que la valeur des maxima ne peut pas caractériser seule les frontières d'une texture, ce qui importe le plus est l'organisation spatiale de ces maxima. La seconde étape de l'algorithme consiste à isoler les maxima appartenant à des chaînes de maxima, et, la dernière, à construire un algorithme de regroupement des chaînes.

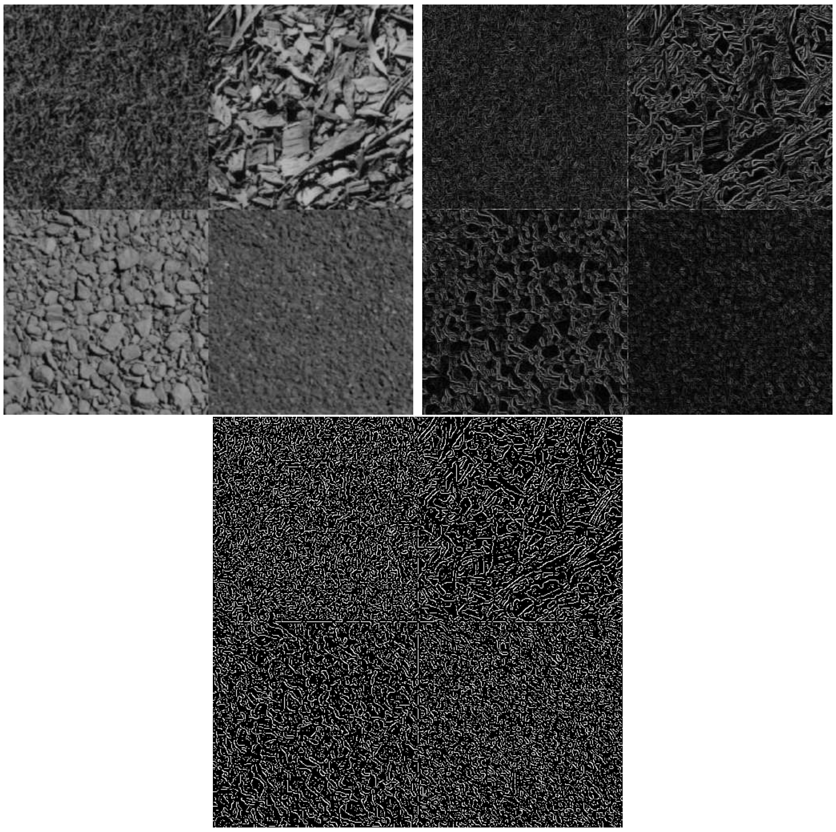

FIG. 5.3 – image en haut à gauche : image de textures, image en haut à droite : image des coefficients $Mf(k,l,1)$, image du bas : image des maxima de $Mf(k,l,1)$ dans la direction du vecteur $Wf(k,l,2^j)$

5.5.2 Construction des chaînes de maxima

Dans notre étude, nous nous limiterons à l'utilisation de la première échelle ($j = 1$) pour l'étude des bords des textures. En effet, les frontières de textures, sauf dans des cas triviaux, correspondent à des singularités difficilement détectable lorsque l'échelle augmente.
Nous allons comparer deux façons de construire des chaînes de maxima, l'une naturelle et implémentée dans [66] et, l'autre, nouvelle, mieux adaptée au traitement des bords des textures.
Première méthode (Classique [66])

- On construit une chaîne de maxima en partant d'un maximum local (k,l) et l'on regarde si, dans la direction orthogonale au vecteur $Wf(k,l,2)$, se trouve un maximum local. Supposons qu'il en existe un.
- Soit (k',l') ce maximum local. Si, partant de ce point, il existe un unique maximum local dans la direction orthogonale à $Wf(k',l',2)$, différent de (k,l) alors on prolonge la chaîne

par ce maximum. S'il n'existe pas d'autre maximum local dans la direction orthogonale au vecteur $Wf(k', l', 2)$, (k, l) mis à part, (k', l') est un bout de chaîne. Enfin, si l'on a deux maxima locaux distincts de (k, l), dans la direction du gradient, on prolonge la chaîne dans la direction induisant le plus grand rayon de courbure, de façon à induire le bord le plus régulier possible.

Cette méthode, efficace lorsque l'on a des bords disjoints, est difficile à appliquer lorsque les bords ne sont pas réguliers, i.e, lorsque le vecteur $Wf(k, l, 2)$ varie. Par ailleurs, le principal inconvénient est que l'on ne peut pas forcément parcourir les chaînes de maxima d'une extrémité à l'autre. Etant donnée une image, le regroupement des maxima locaux en chaînes de maxima n'est pas unique. Cela est dû à l'éventuelle irrégularité du bord ou à la jonction de bords, ce dernier cas se produisant très couramment au niveau des frontières des textures.

Seconde méthode (Méthode par symétrie)

Nous introduisons une relation de voisinage sur les maxima locaux voisins. Soient (k, l) et (k', l') deux maxima locaux voisins, on définit :

$$(k, l) \sim (k', l') \Leftrightarrow \left\{ \begin{array}{c} (k' - k, l' - l) \in vect(Wf(k, l, 2))^{\perp} \\ (k - k', l - l') \in vect(Wf(k', l', 2))^{\perp} \end{array} \right.$$

On définit ensuite les chaînes de maxima, que l'on assimile aux bords des textures, en considérant l'ensemble des couples de maxima locaux liés par la relation précédente. Si un point appartient à deux couples, c'est un point intérieur à une chaîne, sinon s'il appartient à un seul couple, c'est un bout de chaîne.

L'avantage d'une telle construction est que, partant d'un bout de chaîne de maxima, si l'on souhaite remonter cette chaîne, on retombe sur l'extrémité de départ. Un maximum local ne peut appartenir qu'à au plus une chaîne.

Les bords ayant été déterminés, nous écrivons un algorithme dont le but est de retrouver les portions manquantes des bords. La section suivante récapitule les différentes étapes et détaille le procédé de regroupement des bords.

5.5.3 Mise en place de l'algorithme

La première étape est de déterminer les maxima des coefficients $Mf(k, l, 2)$ dans la direction du vecteur $Wf = (W^1 f, W^2 f)$. Appelons $\theta(k, l)$ l'angle entre $Wf(k, l, 2)$ et le vecteur horizontal orienté vers la droite (voir Figure 5.4), cet angle appartenant à $[0, 2\pi]$. Alors, la recherche du maximum dans la direction de $Wf(k, l, 2)$ se fait de la manière suivante :

- si $\theta(k, l) \in [0, \arctan(\sqrt{2} - 1)] \cup [\pi - \arctan(\sqrt{2} - 1), \pi] \cup [\pi, \pi + \arctan(\sqrt{2} - 1)] \cup [2\pi - \arctan(\sqrt{2} - 1), 2\pi]$, alors comparaison avec la valeur aux points $(k, l - 1)$ et $(k, l + 1)$
- si $\theta(k, l) \in [\arctan(\sqrt{2} - 1), \arctan(\sqrt{2} + 1)] \cup [\pi + \arctan(\sqrt{2} - 1), \pi + \arctan(\sqrt{2} + 1)]$, alors comparaison avec la valeur aux points $(k - 1, l + 1)$ et $(k + 1, l - 1)$
- si $\theta(k, l) \in [\pi - \arctan(\sqrt{2}+1), \pi - \arctan(\sqrt{2}-1)] \cup [2\pi - \arctan(\sqrt{2}+1), 2\pi - \arctan(\sqrt{2}-1)]$, alors comparaison avec la valeur aux points $(k + 1, l + 1)$ et $(k - 1, l - 1)$
- si $\theta(k, l) \in [\arctan(\sqrt{2}+1), \pi - \arctan(\sqrt{2}+1)] \cup [\pi + \arctan(\sqrt{2}+1), 2\pi - \arctan(\sqrt{2}+1)]$, alors comparaison avec la valeur aux points $(k - 1, l)$ et $(k + 1, l)$.

Les bornes des intervalles sont trouvées en comparant les produits scalaires entre le vecteur $(\cos(\theta(k, l)), \sin(\theta(k, l)))$, et les vecteurs directeurs associés aux directions du centre du pixel considéré aux pixels voisins. Sur la Figure 5.4, on trouve dans chacun des pixels voisins de (k, l) les vecteurs directeurs correspondant aux directions du centre du pixel (k, l) au centre du pixel considéré.

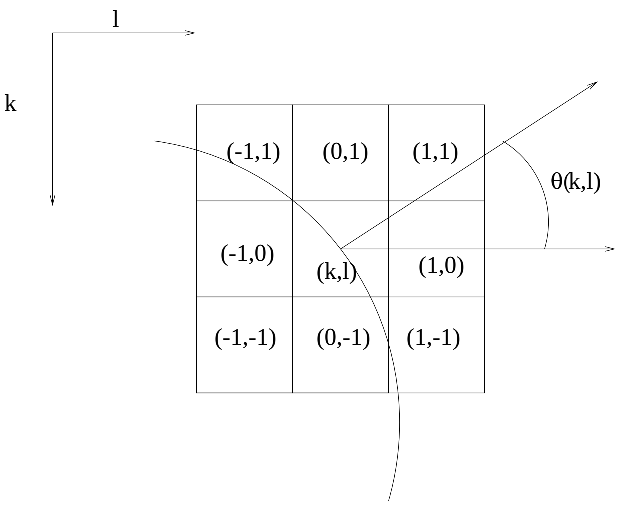

Fig. 5.4 – On représente la direction des axes sur l'image, le pixel (k,l) et ses voisins, la direction $\theta(k,l)$ et la direction des centres des pixels voisins de (k,l), vus depuis le centre de celui-ci

Pour la recherche des bords, on effectue juste une rotation de $\frac{\pi}{2}$, des directions de recherche précédente, i.e :

- si $\theta(k,l) \in [0, \arctan(\sqrt{2}-1)] \cup [\pi - \arctan(\sqrt{2}-1), \pi] \cup [\pi, \pi + \arctan(\sqrt{2}-1)] \cup [2\pi - \arctan(\sqrt{2}-1), 2\pi]$, alors comparaison avec la valeur aux points $(k+1,l)$ et $(k-1,l)$
- si $\theta(k,l) \in [\arctan(\sqrt{2}-1), \arctan(\sqrt{2}+1)] \cup [\pi + \arctan(\sqrt{2}-1), \pi + \arctan(\sqrt{2}+1)]$, alors comparaison avec la valeur aux points $(k+1,l+1)$ et $(k-1,l-1)$
- si $\theta(k,l) \in [\pi - \arctan(\sqrt{2}+1), \pi - \arctan(\sqrt{2}-1)] \cup [2\pi - \arctan(\sqrt{2}+1), 2\pi - \arctan(\sqrt{2}-1)]$, alors comparaison avec la valeur aux points $(k-1,l+1)$ et $(k+1,l-1)$
- si $\theta(k,l) \in [\arctan(\sqrt{2}+1), \pi - \arctan(\sqrt{2}+1)] \cup [\pi + \arctan(\sqrt{2}+1), 2\pi - \arctan(\sqrt{2}+1)]$, alors comparaison avec la valeur aux points $(k,l-1)$ et $(k,l+1)$.

On détermine ensuite les bords en utilisant la méthode par symétrie, présentée dans la section précédente. En Figure 5.5, on représente les bords de longueur supérieure à 5 pixels.

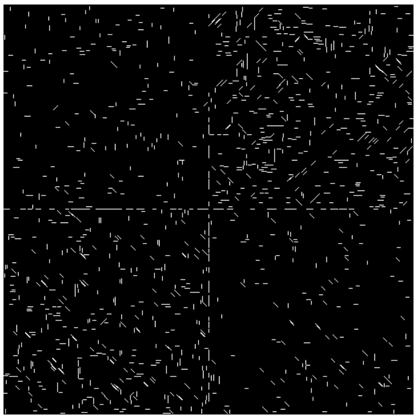

Fig. 5.5 – image des bords de longueur supérieure à cinq pixels de l'image de texture étudiée

Comme les bords générés sont discontinus, il est nécessaire de les regrouper pour essayer de reconstruire les frontières des textures, ainsi que les bords intérieurs aux textures. Comme nous n'avons pas construit de modèles permettant de distinguer un bord intérieur à une texture d'une frontière, les résultats obtenus sur le regroupement seront peu lisibles, cependant nous pourrons en déduire une comparaison des différents algorithmes de reconstruction, que nous allons présenter maintenant, en nous limitant aux résultats de reconstruction sur les frontières des textures.

Nous présentons successivement trois méthodes différentes de calcul de la direction de prolongation des bords. La première cherche à prolonger chaque bord dans la direction orthogonale au vecteur $Wf(k,l,2)$, où (k,l) est une extrémité d'un bord. La seconde cherche à prolonger chaque bord dans la direction du gradient moyen, dont on précisera la définition , et enfin la troisième propose une méthode de prolongation dans la direction du bord, calculée sans l'utilisation d'ondelettes. Nous comparerons les résultats des trois méthodes en termes de reconstruction des frontières des textures.

Les trois méthodes ne diffèrent que par la définition des directions de prolongation. Le principe de l'algorithme, dans chacun des trois cas, est de regarder à l'extrémité de chaque bord i quel est le bord j dont l'extrémité est la plus proche, dans un voisinage de taille donnée de l'extrémité que l'on considère, en utilisant la fonction coût :

$$W_{ij} = 2 - \cos(\theta_{ji}) - \cos(\theta_{ij})$$

dont on cherche le minimum. La signification des angles θ_{ij} et θ_{ji} se trouve en Figure 5.6. Pour donner un sens à cette fonction coût, il faut préciser que les bords que l'on cherche à reconstruire sont de classe C^1. Les différentes méthodes portent donc sur la façon d'évaluer θ_{ij} et θ_{ji}.

<u>Première méthode (direction orthogonale au gradient) :</u>

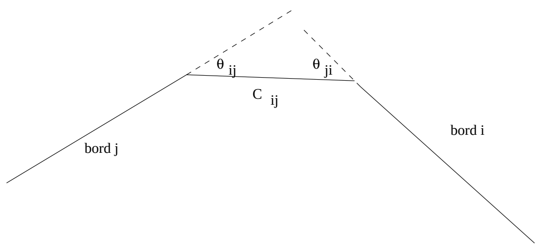

FIG. 5.6 – Visualisation des angles entre le bord et la direction de prolongation utilisée pour reconstruire le bord

La procédure consiste à considérer l'orthogonal au vecteur Wf calculé à l'extrémité du bord B_i, puis, de rechercher dans un voisinage de cet extrémité le minimum de W_{ij}. On détermine ainsi le bord B_j et l'on calcule les angles θ_{ij} et θ_{ji}.

<u>Seconde méthode (direction du gradient moyen) :</u>

Au lieu de prolonger dans la direction orthogonale au gradient aux extrémités, nous allons prolonger dans la direction orthogonale au gradient moyen. Ce gradient moyen est calculé sur le bord B_i à l'aide de :

$$D1 = \frac{1}{|B_i|} \sum_{(k,l)\in B_i} |W^1 f(k,l,2)|$$

$$D2 = \frac{1}{|B_i|} \sum_{(k,l)\in B_i} |W^2 f(k,l,2)|$$

Si, sur le bord B_i, les coefficients $W^1 f$ et $W^2 f$ sont majoritairement de même signe, alors la direction du gradient moyen est donnée par :

$$\nabla B_i = \arctan(\frac{D2}{D1})$$

Si, sur le bord B_i, les coefficients $W^1 f$ et $W^2 f$ sont majoritairement de signe opposé, alors la direction du gradient moyen est donnée par :

$$\nabla B_i = -\arctan(\frac{D2}{D1})$$

Donc, dans le premier cas, la direction orthogonale au gradient est donnée par :

$$\nabla {B_i}^{\perp} = \frac{\pi}{2} + \arctan(\frac{D2}{D1})$$

et, sinon, par :

$$\nabla {B_i}^{\perp} = \frac{\pi}{2} - \arctan(\frac{D2}{D1})$$

On recherche alors le minimum de W_{ij} en utilisant la direction de prolongation ainsi définie, et l'on calcule, alors, θ_{ij} et θ_{ji}.

Troisième méthode(direction du bord) :

Dans ce paragraphe, on détermine la direction de prolongation en utilisant directement les directions moyennes des bords déterminés par la méthode par symétrie. La direction de prolongation des bords est donc déterminée par :

$$\nabla {B_i}^{\perp} = \arctan\left(\frac{ordo(E(B_i)_1) - ordo(E(B_i)_2)}{absc(E(B_i)_1) - absc(E(B_i)_2)}\right),$$

où $E(B_i)_1$ et $E(B_i)_2$ représentent les extrémités du bord B_i, $absc$ et $ordo$ signifiant abscisse et ordonnée.

Pour chacune des trois méthodes, on vient de déterminer quel était le plus proche voisin de chaque extrémité. On regroupe alors les bords en considérant d'abord les bords les plus longs identifiés par la méthode par symétrie puis en imposant que le coût de la reconstruction soit inférieur à un certain seuil. Par ailleurs, la recherche des bords voisins s'effectue sur plusieurs tailles de voisinage en commençant par reconstruire les bords en considèrant une échelle minimale puis, lorsque l'algorithme s'arrête, en regardant les reconstructions possibles pour une taille de voisinage plus importante. Les résultats de la reconstruction sur l'image des quatre textures présentée au début du chapitre sont visibles pour les trois méthodes respectivement en Figure 5.7, en Figure 5.8 et en Figure 5.9. Pour chacune des trois méthodes seuls les bords de longueur supérieure à 5 pixels ont été considérés.

5.6 Analyse des résultats

Cette étude a permis de mettre en évidence les problèmes de définition des bords des images à partir des coefficients d'ondelettes, et nous avons introduit une définition non ambigüe.

Ensuite, nous avons comparé plusieurs méthodes de regroupement des bords et, il apparaît au niveau des frontières de textures que prolonger dans la direction orthogonale au vecteur $Wf(k,l,2)$ est beaucoup trop instable. Cela montre aussi clairement que des résultats, qui sont vrais si l'on considère les bords d'une image supposée être une fonction continue, se trouve

démentis lorsque l'on considère l'image d'un point de vue discret.
La méthode de prolongation dans la direction du bord donne les meilleurs résultats au niveau des frontières des textures.

Un développement futur consistera à voir comment obtenir des frontières fermées de textures et de comparer les résultats obtenus avec l'analyse des textures faite au chapitre précédent.

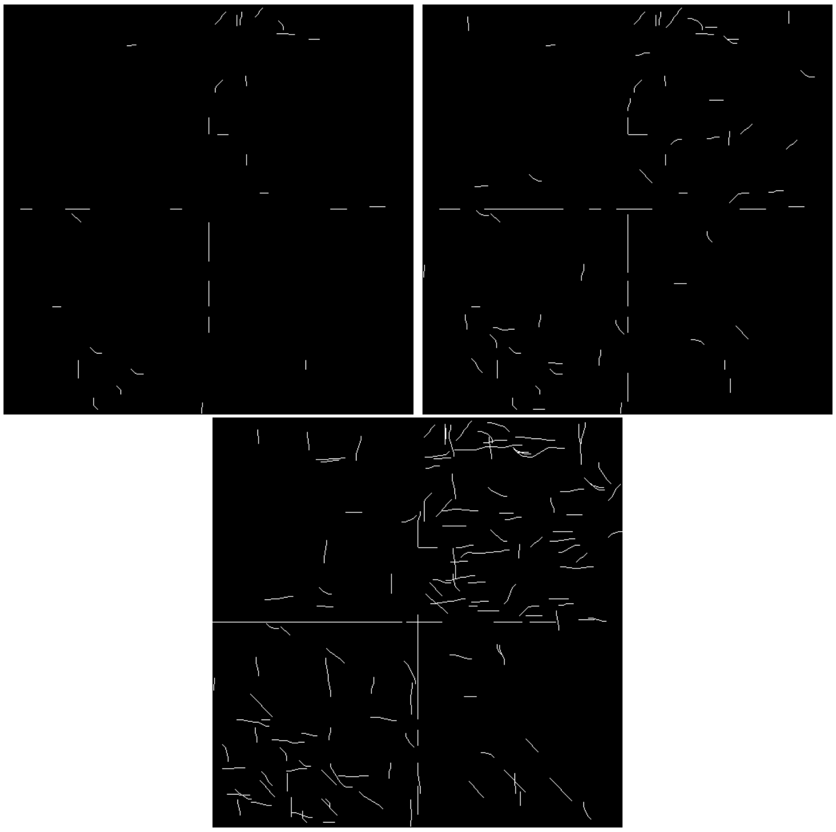

FIG. 5.7 – image en haut à gauche : regroupement des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon 4 pixels, image en haut à droite : regroupement des bords obtenus, en supposant le cout inférieur à 0.1 et en considérant des voisinages de rayon inférieur à 10 pixels, image du bas : regroupement des bords obtenus, en supposant le cout inférieur à 0.1 et en considérant des voisinages de rayon inférieur à 20 pixels.

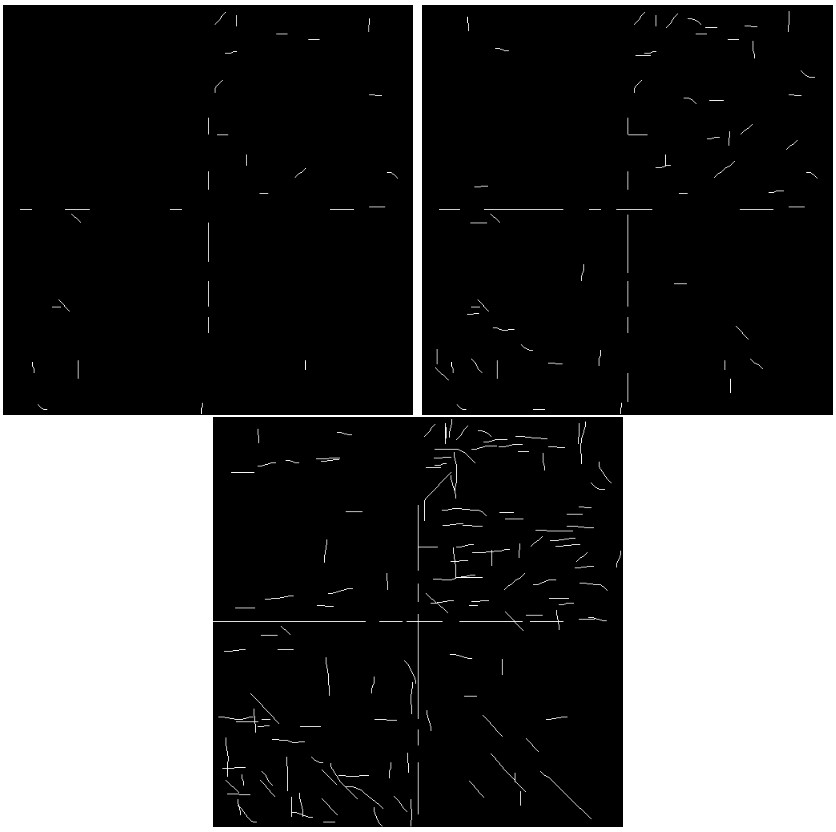

FIG. 5.8 – image en haut à gauche : regroupement en utilisant la direction du gradient moyen des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon 4 pixels, image en haut à droite : regroupement en utilisant la direction du gradient moyen des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon inférieur à 10 pixels, image du bas : regroupement en utilisant la direction du gradient moyen des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon inférieur à 20 pixels

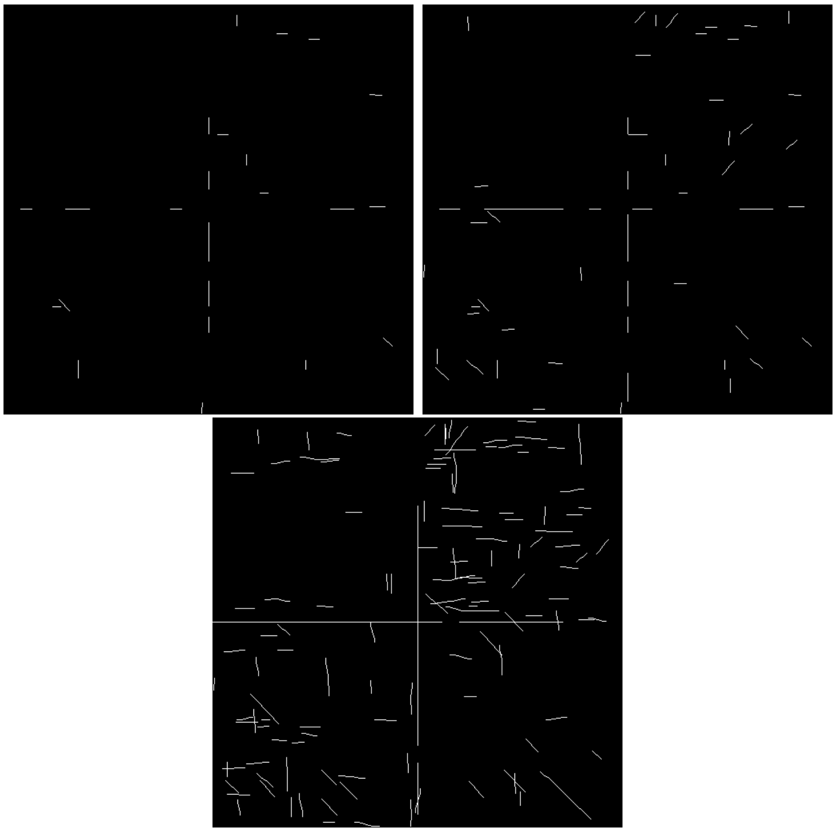

FIG. 5.9 – image en haut à gauche : regroupement en utilisant la direction des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon 4 pixels, image en haut à droite : regroupement en utilisant la direction des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon inférieur à 10 pixels, image du bas : regroupement en utilisant la direction des bords obtenus, en supposant le coût inférieur à 0.1 et en considérant des voisinages de rayon inférieur à 20 pixels

Deuxième partie

Analyse de singularités

1

Détection de singularités basée sur une analyse multi-échelles

Dans ce chapitre, nous allons étudier comment détecter et suivre des singularités à différentes échelles au moyen de la transformée de Berkner [4], que nous modifierons et pour laquelle nous donnerons des propriétés nouvelles. L'intérêt de cette méthode, pour le suivi des singularités le long des échelles, apparaîtra clairement lorsque l'on chaînera les maxima d'une échelle à l'autre. En effet, cette méthode permettra un chaînage automatique alors que les méthodes habituelles [66], utilisant des ondelettes, sont fondées sur des décompositions dyadiques pour lesquels le chaînage inter-échelles est compliqué.
Nous donnerons deux applications intéressantes de ce schéma de décomposition au débruitage automatique des signaux puis à l'analyse des signaux EMG (électromyogramme) au chapitre suivant.

1.1 La transformée de Berkner

Dans cette section, on introduit une transformée basée sur les coefficients binomiaux ; cette transformée est une variante de celle introduite par Berkner [4], et nous allons faire ressortir des propriétés qui n'avaient pas été mises en évidence dans [4]. Comme il a été démontré dans [73], nous avons une propriété d'approximation des lois normales par les coefficients binomiaux. Soit :

$$\rho_N(j) = \frac{1}{2^{N+1}} C_N^j,$$

alors, on a l'approximation :

$$\rho_N(j) \approx \frac{1}{\sqrt{2\pi}} \frac{1}{\frac{\sqrt{N+1}}{2}} e^{-\frac{1}{2}((j-(\frac{N+1}{2}))/(\sqrt{N+1}/2))^2}$$

Par ailleurs, la rième dérivée d'une gaussienne peut-être approchée par :

$$\rho_N^r(j) = \begin{cases} \sum\limits_{l=0}^{r} (-1)^l C_r^l \rho_N(j-l) & \text{si } 0 \leq j \leq N+r \\ 0 \text{ si } j \notin [0, N+r] \end{cases}$$

Cette suite permet d'approcher la dérivée d'une gaussienne au sens :

$$\rho_N^1(j) \approx \frac{1}{\sqrt{2\pi}} \frac{1}{\frac{N+1}{4}} \frac{(j - \frac{N+1}{2})}{\sqrt{N+1}/2} e^{-\frac{1}{2}((j-(\frac{N+1}{2}))/(\sqrt{N+1}/2))^2}$$

D'après la formule de décomposition des coefficients binômiaux ("triangle de Pascal"), les coefficients à une échelle s'obtiennent facilement en fonction de l'échelle précédente selon la formule de récurrence suivante :

$$\rho^r_{N+1}(j) = \frac{1}{2}(\rho^r_N(j) + \rho^r_N(j-1))$$

On démontre, alors, que :

$$\sum_j \rho^r_N(j) = 0$$

1.2 Algorithme de décomposition d'un signal

Nous proposons ici une adaptation de l'algorithme développé dans [4]. On définit les coefficients $c^r_{N,k}$ comme la convolution d'un signal f avec la séquence $\tilde{\rho}^r_N(j) = \rho^r_N(j + \frac{N+1}{2})$. Dans le cas $r = 1$, on constate, en Figure 1.1, que la convergence quand $N \leftrightarrow +\infty$ vers la dérivée de la gaussienne est très rapide, étant en $O(N^{-\frac{1}{2}})$ [4].
On dispose de la formule de décomposition suivante :

$$c^r_{N,k} = \sum_{j\in\mathbb{Z}} \tilde{\rho}^r_N(j) f(k+j) \quad (1)$$

De par la définition de $\tilde{\rho}$, on obtient les relations entre échelles suivantes :

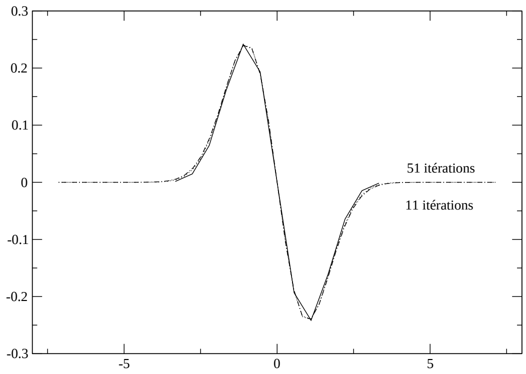

FIG. 1.1 – valeurs de $(\frac{2j}{\sqrt{N+1}}, \frac{\sqrt{N+1}}{2}\tilde{\rho}^r_N(j))$ pour $N = 11$ et $N = 51$

Propriété 1 *Si $N + r$ est pair, on a :*

$$c^r_{N,k} = \frac{1}{2}(c^r_{N-1,k} + c^r_{N-1,k-1})$$

Si $N + r$ est impair, on a :

$$c^r_{N,k} = \frac{1}{2}(c^r_{N-1,k} + c^r_{N-1,k+1})$$

Démonstration :
En effet, comme :

$$c^r_{N,k} = \sum_{j\in\mathbb{Z}} \tilde{\rho}^r_N(j) f(k+j),$$

dans le cas où $N+r$ est pair :

$$\tilde{\rho}_N^r(j) = \rho_N^r(j+\frac{N+r}{2}) = \frac{1}{2}(\rho_{N-1}^r(j-1+\frac{N+r}{2})+\rho_{N-1}^r(j+\frac{N+r}{2})) = \frac{1}{2}(\tilde{\rho}_{N-1}^r(j)+\tilde{\rho}_{N-1}^r(j+1))$$

d'où, l'on déduit :

$$c_{N,k}^r = \frac{1}{2}\sum_{j\in\mathbb{Z}}(\tilde{\rho}_{N-1}^r(j)+\tilde{\rho}_{N-1}^r(j+1))f(j+k) = \frac{1}{2}(c_{N-1,k}^r + c_{N-1,k-1}^r)$$

De la même façon, on obtient le résultat pour $N+r$ impair. ■

1.3 Algorithme de détection des fortes variations d'un signal

Nous allons voir, dans un premier temps, comment détecter les fortes variations d'un signal f en étudiant les extrema de la décomposition obtenue grâce à la formule (1) de la section précédente avec $r=1$. Ensuite, nous regarderons comment suivre les chaînes d'extrema le long des échelles, i.e. lorsque N augmente. Nous parlerons d'échelle pour désigner N, même si d'après la convergence de $\rho_N^1(j)$, on peut seulement considérer qu'asymptotiquement $\frac{\sqrt{N+1}}{2}$ est une échelle.

1.3.1 Détection des fortes variations et suivi le long des échelles

Pour $N=0$, en considérant $r=1$, on peut détecter les fortes variations locales de f, en étudiant les extrema locaux des coefficients. Nous donnons ici une définition des fortes variations de f :

Définition 1 *on dit que f admet une forte variation au point k si :*
$f(k)-f(k+1)>0$ *:*
$f(k)-f(k+1)\geq(>)f(k+1)-f(k+2)$ *et* $f(k)-f(k+1)>(\geq)f(k-1)-f(k)$
une des deux inégalités étant stricte
ou $f(k)-f(k+1)<0$ *:*
$f(k)-f(k+1)\leq(<)f(k+1)-f(k+2)$ *et* $f(k)-f(k+1)<(\leq)f(k-1)-f(k)$
une des deux inégalités étant stricte

Cette notion de fortes variations permet de prendre en compte la plupart des singularités présentes dans l'image. Si l'on étudie cette définition du point de vue des coefficients, nous avons la caractérisation suivante des fortes variations de f :

$$c_{0,k}^1 > 0 \text{ et } c_{0,k-1}^1 \leq (<) c_{0,k}^1 > (\geq) c_{0,k+1}^1$$
$$c_{0,k}^1 < 0 \text{ et } c_{0,k-1}^1 > (\geq) c_{0,k}^1 \leq< c_{0,k+1}^1$$

Donc, les fortes variations du signal f correspondent aux extrema des coefficients de la décomposition.

On peut alors suivre les extrema locaux le long des échelles. Donnons, avant cela, la définition :

Définition 2 $c_{N,k}^1$ *est un extremum local à l'échelle N, si :*

$$\begin{aligned} & c_{N,k}^1 > 0 \ \textit{et}\ c_{N,k-1}^1 < c_{N,k}^1 \geq c_{N,k+1}^1 \\ \textit{ou}\ & c_{N,k}^1 > 0 \ \textit{et}\ c_{N,k-1}^1 \leq c_{N,k}^1 > c_{N,k+1}^1 \\ \textit{ou}\ & c_{N,k}^1 < 0 \ \textit{et}\ c_{N,k-1}^1 > c_{N,k}^1 \leq c_{N,k+1}^1 \\ \textit{ou}\ & c_{N,k}^1 < 0 \ \textit{et}\ c_{N,k-1}^1 \geq c_{N,k}^1 < c_{N,k+1}^1 \end{aligned}$$

Les deux premières lignes correspondent à la définition des maxima, les deux dernières lignes à celle des minima. Avec la décomposition des coefficients présentée à la section précédente, nous montrons que l'on peut connaître l'évolution des extrema lorsque N augmente (Propriété 2) et, retrouver, à partir d'un extrema à n'importe quelle échelle, à quel extremum, à l'échelle $N = 0$, il est relié (Propriété 3). On a donc les propriétés suivantes :

Propriété 2 *Si $c^1_{N-1,k}$ est un extremum local, les extrema éventuels à l'échelle N sont à chercher dans $[k-1,k]$ si $N+1$ impair et $[k,k+1]$ sinon.*

Démonstration :
premier cas : $N+1$ pair
Les coefficients faisant intervenir l'extremum local $c^1_{N-1,k}$ sont :

$$\begin{aligned} c^1_{N,k} &= \tfrac{1}{2}(c^1_{N-1,k} + c^1_{N-1,k-1}) \\ c^1_{N,k+1} &= \tfrac{1}{2}(c^1_{N-1,k+1} + c^1_{N-1,k}) \end{aligned}$$

L'extremum se propage donc, si l'un de ces deux coefficients reste extremum à l'échelle N.
deuxième cas : $N+1$ impair
Les coefficients faisant intervenir l'extremum local $c^1_{N-1,k}$ sont :

$$\begin{aligned} c^1_{N,k} &= \tfrac{1}{2}(c^1_{N-1,k} + c^1_{N-1,k+1}) \\ c^1_{N,k-1} &= \tfrac{1}{2}(c^1_{N-1,k-1} + c^1_{N-1,k}), \end{aligned}$$

d'où la conclusion.■
Inversement, il est important de constater que l'on ne crée pas d'extrema, lorsque l'échelle augmente.

Propriété 3 *Supposons que l'on ait un extremum en k pour l'échelle N, alors, si $N+1$ est pair, on avait obligatoirement un extremum de même nature en k ou en $k-1$ à l'échelle $N-1$ si $N+1$ est pair et en k ou en $k+1$ si $N+1$ est impair.*

Démonstration :
Par exemple, prenons le cas où $N+1$ est pair. Si l'on a un maximum local en k à l'échelle N, c'est que l'on avait un maximum local en k ou en $k-1$ à l'échelle $N-1$.
En effet, supposons que l'on ait un maximum local en k, alors, par définition du coefficient $c^1_{N,k}$ cela équivaut à :

$$c^1_{N-1,k-1} \geq c^1_{N-1,k+1} \text{ et } c^1_{N-1,k-2} < c^1_{N-1,k} \ (1)$$

ou :

$$c^1_{N-1,k-1} > c^1_{N-1,k+1} \text{ et } c^1_{N-1,k-2} \leq c^1_{N-1,k} \ (2)$$

et

$$c^1_{N,k} > 0$$

On a alors deux cas.
Si $c^1_{N-1,k} \leq 0$, alors comme $c^1_{N,k} > 0$, on a $c^1_{N-1,k-1} > 0$. De ceci, on tire que $c^1_{N-1,k} < c^1_{N-1,k-1} \geq c^1_{N-1,k-2}$, donc $c^1_{N-1,k-1}$ est un maximum local.
Si $c^1_{N-1,k} > 0$, alors :
si $c^1_{N-1,k-1} > c^1_{N-1,k}$, alors $c^1_{N-1,k-1}$ est un maximum local.
Si $c^1_{N-1,k-1} \leq c^1_{N-1,k}$, alors $c^1_{N-1,k}$ est un maximum local, si la condition (2) s'applique, et si la condition (1) s'applique, le seul cas où $c^1_{N-1,k}$ n'est pas maximum local est lorsque $c^1_{N-1,k-1} = c^1_{N-1,k} = c^1_{N-1,k+1}$ et, comme $c^1_{N-1,k-2} < c^1_{N-1,k}$, $c^1_{N-1,k-1}$ est un maximum local.

Regardons ensuite ce qu'il en est lorsque $N+1$ est impair. Supposons que l'on ait un maximum local en k et montrons qu'alors il y avait forcément un maximum local en k ou en $k+1$ à l'échelle $N-1$. Le fait que l'on ait un maximum local à l'échelle N se traduit par :

$$c^1_{N-1,k+1} \geq c^1_{N-1,k-1} \text{ et } c^1_{N-1,k+2} < c^1_{N-1,k} \ (1)$$

$$c^1_{N-1,k+1} > c^1_{N-1,k-1} \text{ et } c^1_{N-1,k+2} \leq c^1_{N-1,k} \ (2)$$

et

$$c^1_{N,k} > 0$$

Si $c^1_{N-1,k} \leq 0$, alors comme $c^1_{N,k} > 0$, on a $c^1_{N-1,k+1} > 0$. De ceci, on tire que $c^1_{N-1,k} < c^1_{N-1,k+1} > c^1_{N-1,k+2}$, donc $c^1_{N-1,k+1}$ est un maximum local.
Si $c^1_{N-1,k} > 0$, alors :
si $c^1_{N-1,k+1} > c^1_{N-1,k}$, alors $c^1_{N-1,k+1}$ est un maximum local. Si $c^1_{N-1,k+1} \leq c^1_{N-1,k}$, alors $c^1_{N-1,k}$ est un maximum local, si la condition (2) s'applique et, si la condition (1) s'applique, le seul cas où $c_{N-1,k}$ n'est pas un maximum local correspond à $c^1_{N-1,k} = c^1_{N-1,k+1} = c^1_{N-1,k+2}$, mais alors $c^1_{N-1,k} > c^1_{N-1,k+2}$ et on obtient que $c^1_{N-1,k+1}$ est un maximum local.

Un raisonnement analogue pourrait être mené pour les minima locaux.■

1.4 Chaînage des extrema le long des échelles

Le chaînage des maxima de la transformée en ondelettes le long des échelles permet de caractériser la régularité des bords présents dans l'image [65]. Le principal problème concerne le fait que le chaînage est rendu délicat lorsque la discrétisation en échelle n'est pas assez fine ; cela est, typiquement, le cas avec la décomposition en ondelettes dyadiques .
L'avantage de la présente méthode est qu'elle permet un chaînage automatique des extrema. Nous verrons comment ce chaînage permet un débruitage automatique signaux sans utiliser d'hypothèse sur la nature du bruit.

Pour suivre les extrema depuis les petites échelles vers les grandes, il suffit de regarder les échelles successivement. Cependant, deux cas particuliers sont à considérer. On peut avoir deux extrema locaux voisins à l'échelle $N-1$, alors que l'on a un seul extremum à l'échelle N. Dans ce cas, il faut alors choisir l'un des deux extrema pour prolonger la chaîne (cela se produit pour $c^1_{N-1,k-2} < c^1_{N-1,k-1} = c^1_{N-1,k} > c^1_{N-1,k+1}$ pour les maxima locaux, avec $N+1$ pair, les inégalités étant inversées si l'on considère des minima locaux).

Un autre cas peut se produire, qui est le dédoublement d'un extremum. Cette situation se produit, par exemple, dans l'étude des maxima, lorsque $N+1$ est pair, si $c^1_{N-1,k+1} = c^1_{N-1,k-1}$, $c^1_{N-1,k+2} < c^1_{N-1,k}$ et $c^1_{N-1,k-2} < c^1_{N-1,k}$. Dans ce cas, il faut aussi choisir de ne garder qu'un des deux extrema. Donc, avec un tel algorithme, il peut y avoir création d'extrema, mais il est toujours possible de relier ce nouvel extremum à un extremum à l'échelle $N = 0$, d'après la propriété 3. En théorie, des situations semblables ne devraient pas pouvoir se produire car, si l'on travaille sur des réels, avec les erreurs d'approximation commises, on ne peut pas avoir une stricte égalité de deux coefficients. Néanmoins, comme on travaille initialement sur des valeurs entières, de tels cas se produisent et il faut les traiter.

1.4.1 Algorithme de chaînage

A partir des remarques faites au paragraphe précédent, on peut construire un algorithme simple de chaînage d'extrema d'une échelle à l'autre.

Pour faciliter les calculs, nous chaînons les extrema en partant des échelles les plus fines vers les échelles les plus grossières. Cela évite la mémorisation d'une matrice creuse très grande. Le principe est le suivant : on détermine les extrema locaux à la première échelle, puis on regarde si ces extrema se propagent à l'échelle suivante, en utilisant la propriété 2. Quatre cas peuvent se produire :

- la chaîne issue de l'échelle $N-1$ se prolonge en un maximum unique à l'échelle N
- la chaîne issue de l'échelle $N-1$ disparaît
- Deux chaînes à l'échelle $N-1$ se rejoignent
- Une chaîne à l'échelle $N-1$ bifurque en deux chaînes

Les deux derniers cas sont dûs à des artefacts de calculs et, si l'on travaillait avec des réels, nous n'aurions pas à les prendre en compte. La stratégie adoptée est, lorsque deux chaînes se rencontrent, de faire s'arrêter l'une des deux à l'échelle correspondante, et lorsqu'une chaîne bifurque, de ne garder qu'une des branches issues de la bifurcation.

Le principe de l'algorithme de débruitage consiste alors à remarquer que les extrema les plus significatifs doivent perdurer le plus longtemps en fonction de N. On mémorise donc la longueur de chaque chaîne et l'on décide de ne garder que les lignes de maxima dont la longueur est supérieure à un certain seuil, comme on le voit en Figure 1.2.
Pour éviter de remonter chacune des chaînes pour retrouver, alors, la localisation initiale de chaque extremum, on garde celle-ci comme paramètre supplémentaire lors de l'étape de chaînage. On peut alors définir un bord ou une singularité par l'échelle jusqu'à laquelle l'extremum des

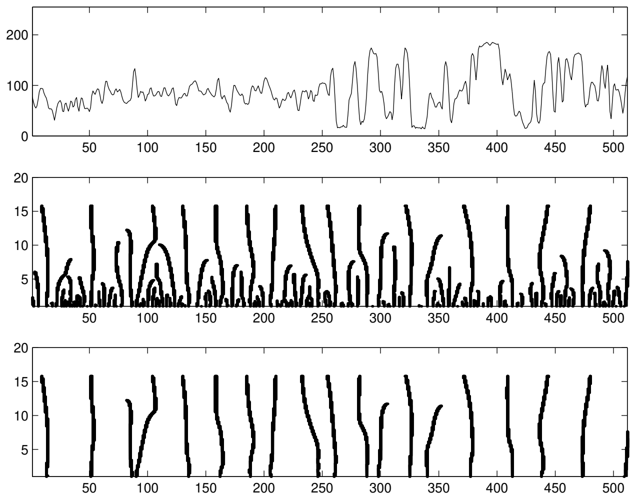

FIG. 1.2 – premier graphe : le signal, second graphe : les chaînes d'extrema locaux en fonction des échelles, troisième graphe : les chaînes d'extrema locaux qui perdurent au delà de l'échelle $N = 500$, en ordonnée on a $\frac{\sqrt{N+1}}{2}$

coefficients correspondants perdure. Cependant, celle-ci dépend beaucoup du voisinage de l'objet ou de la singularité que l'on désire isoler. Nous allons étudier cela sur des exemples, en deux dimensions.

1.5 Algorithme 2D et application à la détection de bords

Contrairement à l'étude faite par Mallat dans [64], où les bords sont repérés par un maximum des coefficients de la décomposition sur deux ondelettes caractérisant les détails verticaux et horizontaux dans la direction définie par le gradient associé à ces deux composantes, on décide de considérer ici la définition suivante de bords :

Définition 3 *On dit que l'on a un bord dans l'image au point* (k, l) *, s'il y a une forte variation, en ce point, dans la direction horizontale ou s'il y a une forte variation, en ce point, dans la direction verticale.*

En effectuant une décomposition selon les lignes et une décomposition selon les colonnes de l'image selon le schéma introduit précédemment, les bords sont repérés par les points où existe un extremum dans la direction horizontale pour la décomposition de l'image selon les lignes , ou un extremum dans la direction verticale pour la décomposition selon les colonnes.

Le principe de l'algorithme est alors d'étudier séparément les extrema selon les lignes et selon les colonnes. L'avantage principal est que le suivi d'extrema est automatique contrairement à l'approche basée sur la détection des maxima dans la direction du gradient où demeure une incertitude, due au fait que la direction doit être adaptée à la topologie discrète de l'image . Il sera cependant intéressant d'essayer de faire le lien avec l'approche classique de Canny [10], adaptée par Mallat au cadre des ondelettes.

Avec notre méthode, on peut déterminer comment se propage un extremum d'une échelle à l'autre. On peut aussi savoir précisément où se trouvaient initialement n'importe quel extremum (mis à part pour quelques cas pathologiques mis en évidence précédemment). On présente, en Figure 1.3, la décomposition selon les lignes et la décomposition selon les colonnes d'une image d'ellipse bruitée.

En théorie, un extremum se prolonge en au plus un extremum d'une échelle à l'autre. Cependant,

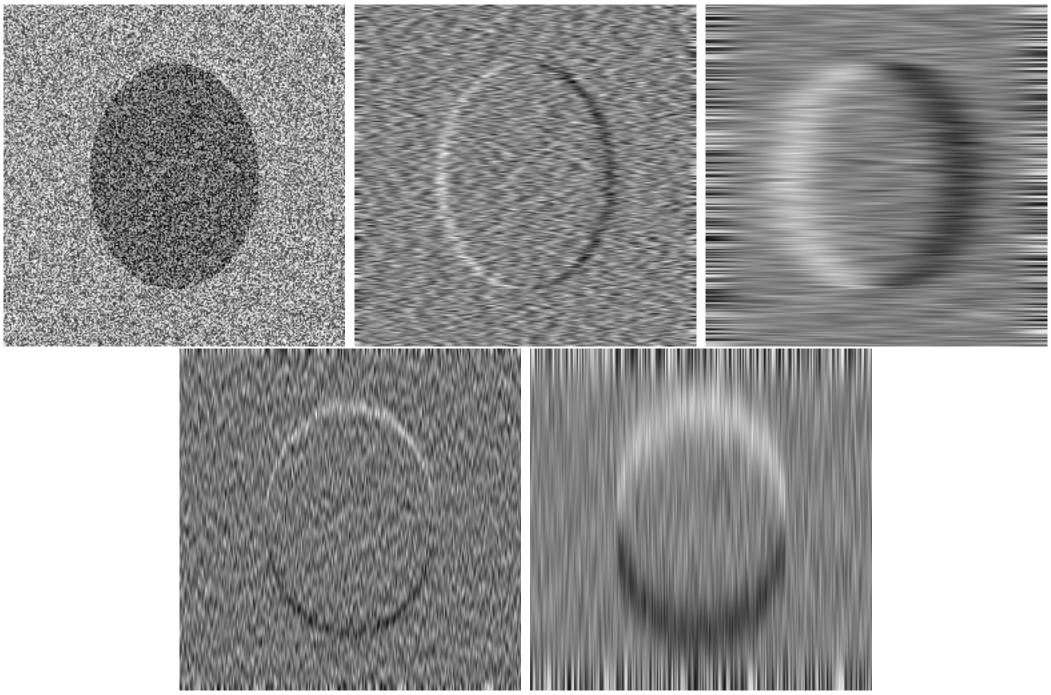

FIG. 1.3 – première ligne : image bruitée, transformation selon les lignes à l'echelle $N = 100$ et jusqu'à l'échelle $N = 1000$, puis sur la deuxième ligne : tranformation sur les colonnes pour $N = 100$ et $N = 1000$

nous avons vu que, numériquement, on pouvait, dans certains cas, avoir un dédoublement d'extremum d'une échelle à l'autre et, dans ce cas, l'algorithme ne garde qu'un des extrema créés. Ainsi, tous les extrema à une échelle n'appartiennent pas forcément à une ligne d'extrema, cependant nous vérifions que ceux-ci sont très peu nombreux. Dans le tableau de la Figure 1.4, on a calculé, sur l'image précédente d'ellipse bruitée, le nombre d'extrema à une échelle et, parmi ceux-ci, ceux qui appartiennent à une ligne d'extrema. La conclusion est que les cas de bifurcations de lignes d'extrema sont très rares.

	N=4	N=100	N=200	N=1000
nb extrema	27513	8842	4746	2362
ligne extrema	27434	8837	4742	2359

	N=4	N=100	N=200	N=1000
nb extrema	27730	8858	4812	2387
ligne extrema	27466	8849	4805	2380

FIG. 1.4 – Extrema de la décomposition selon les lignes n'appartenant pas à une chaîne de maxima ou y appartenant, en fonction de la profondeur de décomposition (premier tableau), étude identique sur les colonnes dans le second tableau.

On regarde, alors, quels sont les extrema qui subsistent à une échelle donnée , i.e, s'il existe une ligne d'extrema correspondant à cet extremum dans la décomposition selon les lignes ou dans la décomposition selon les colonnes.
On représente, sur les images de la Figure 1.5, les origines des chaînes d'extrema qui subsistent dans la décomposition de l'image, soit selon les les lignes, soit selon les colonnes pour différentes échelles. Pour $N = 1000$, on voit que le bord significatif de l'ellipse apparaît bien. Cette méthode

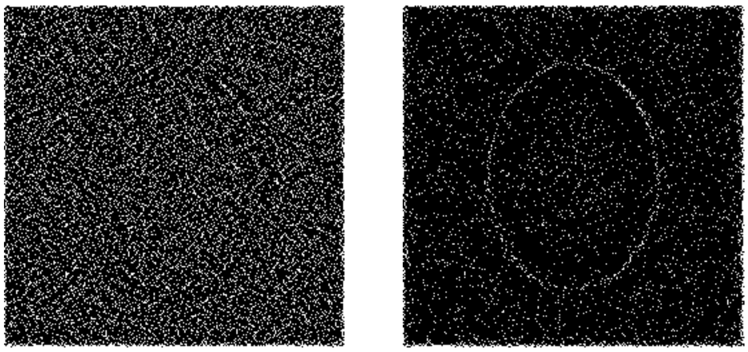

FIG. 1.5 – Extrema selon les lignes ou les colonnes qui perdurent jusqu'à l'échelle $N = 100$ et $N = 1000$.

permet une détection efficace de bords d'images très bruitées, puisque les bords significatifs d'une image disparaissent pour des échelles relativement petites. Nous allons maintenant étudier la signification des extrema locaux obtenus en regard de la méthode proposée par Canny, et relier la décomposition présentée avec la décomposition en ondelettes dyadiques .

1.6 Adaptation de la méthode de Canny à la décomposition de Berkner pour la détection de bords

La définition des bords dans une image est un problème compliqué. Tout le monde s'accorde à dire que les gradients de niveaux de gris sont primordiaux pour la détection des bords, les gradients significatifs du niveau de gris des images étant, pour de nombreux auteurs, ce qui doit être reconnu, puis préservé lors d'une étape de segmentation ([12],[21]). On peut alors définir les bords des images comme les extrema du niveau de gris dans la direction définie par les gradients de niveau de gris [10], ce que nous avons étudié en détail au chapitre précédent.

Mallat [65] a, par la suite, réutilisé cette approche pour définir les bords des images à partir des coefficients d'ondelettes. Dans son étude, il définit deux ondelettes Ψ_1 et Ψ_2 (qui sont les dérivées directionnelles de splines quadratiques), à partir desquelles il construit un frame d'ondelettes (cf chapitre précédent). Il définit ,alors, les produits scalaires d'une image f par ces deux ondelettes, pour, ensuite, en déduire la définition des bords suivante :

Définition 4 *Un point fait partie d'un bord d'une image si, en ce point,* $\sqrt{< f, \Psi_1 >^2 + < f, \Psi_2 >^2}$ *est maximum dans la direction du vecteur* $(< f, \Psi_1 >, < f, \Psi_2 >)$ *au point considéré.*

Dans l'étude menée par Mallat, les ondelettes détectent, l'une les bords horizontaux de l'image et l'autre les bords verticaux. On peut donc faire une analogie avec notre décomposition et regarder les extrema de $\sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2}$, où $c^l_{N,k}$ est le coefficient de la décomposition selon les lignes et $c^c_{N,p}$ est le coefficient de la décomposition selon les colonnes.
Supposons que le vecteur gradient à l'échelle N soit donné par :

$$\nabla f(N, [k,p]) \quad = \quad \begin{pmatrix} c^c_{N,p+1} - c^c_{N,p-1} \\ c^l_{N,k+1} - c^l_{N,k-1} \end{pmatrix}$$

Nous avons alors démontré le théorème suivant :

Propriété 4 *Si* $\sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2}$ *est un maximum local (respectivement minimum), dans la direction du vecteur* $\nabla f(N, [k,p])$ *et si* $c^l_{N,k} > 0$ *(respectivement* < 0*) et* $c^c_{N,p} > 0$ *(respectivement* < 0*), alors* $c^l_{N,k}$ *ou* $c^c_{N,p}$ *est un maximum local (respectivement minimum).*

<u>Démonstration:</u>
Supposons que $\sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2}$ soit maximum dans la direction de ∇f et que $c^l_{N,k} > 0$ et $c^c_{N,p} > 0$.
Regardons alors les différentes éventualités, correspondants au cas où l'on a :

$$\begin{cases} c^l_{N,k+1} - c^l_{N,k-1} > 0 \\ c^c_{N,p+1} - c^c_{N,p-1} > 0, \end{cases} \qquad (1)$$

ou au cas ou l'on a :

$$\begin{cases} c^l_{N,k+1} - c^l_{N,k-1} < 0 \\ c^c_{N,p+1} - c^c_{N,p-1} < 0 \end{cases} \qquad (2)$$

Trois situations peuvent alors se produire, correspondant aux différentes orientations de ∇f.
<u>Premier cas</u> :

$$\sqrt{{c^l_{N,k-1}}^2 + {c^c_{N,p-1}}^2} < \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2} > \sqrt{{c^l_{N,k+1}}^2 + {c^c_{N,p+1}}^2}$$

Supposons la condition (1) vérifiée :
alors, si ${c^l_{N,k}}^2 < {c^l_{N,k+1}}^2 \Rightarrow {c^c_{N,p}}^2 > {c^c_{N,p+1}}^2$,
on obtient alors que $c^c_{N,p}$ est un maximum local.
Alors, si ${c^l_{N,k}}^2 > {c^l_{N,k+1}}^2$, on obtient que $c^l_{N,k}$ est un maximum local.
Supposons la condition (2) vérifiée :
alors, si ${c^l_{N,k}}^2 < {c^l_{N,k-1}}^2 \Rightarrow {c^c_{N,p}}^2 > {c^c_{N,p-1}}^2$, on obtient que $c^c_{N,p}$ est un maximum local.
alors, si ${c^l_{N,k}}^2 > {c^l_{N,k-1}}^2$, on obtient que $c^l_{N,k}$ est un maximum local.

Deuxième cas :

$$\sqrt{{c^l_{N,k-1}}^2 + {c^c_{N,p}}^2} < \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2} > \sqrt{{c^l_{N,k+1}}^2 + {c^c_{N,p}}^2}$$

on obtient que $c^l_{N,k}$ est un maximum local.
Troisième cas :

$$\sqrt{{c^l_{N,k}}^2 + {c^c_{N,p-1}}^2} < \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2} > \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p+1}}^2}$$

on obtient que $c^c_{N,p}$ est un maximum local.
Etudions maintenant les autres cas :

$$\left\{ \begin{array}{l} c^l_{N,k+1} - c^l_{N,k-1} < 0 \\ c^c_{N,p+1} - c^c_{N,p-1} > 0 \end{array} \right. \qquad (1)$$

ou :

$$\left\{ \begin{array}{l} c^l_{N,k+1} - c^l_{N,k-1} > 0 \\ c^c_{N,p+1} - c^c_{N,p-1} < 0 \end{array} \right. \qquad (2)$$

De même que précédemment, trois situations peuvent survenir. On suppose toujours que $\sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2}$ est maximum dans la direction du gradient et que $c^l_{N,k} > 0$ et $c^c_{N,p} > 0$.
Premier cas :

$$\sqrt{{c^l_{N,k-1}}^2 + {c^c_{N,p+1}}^2} < \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2} > \sqrt{{c^l_{N,k+1}}^2 + {c^c_{N,p-1}}^2}$$

Si la condition (1) est remplie :
si $c^l_{N,k} > c^l_{N,k-1}$, on obtient que $c^l_{N,k}$ est un maximum local.
Si $c^l_{N,k} < c^l_{N,k-1}$, on obtient que ${c^c_{N,p}}^2 > {c^c_{N,p+1}}^2$ et, donc, on obtient que $c^c_{N,p}$ est un maximum local.
Si la condition (2) est remplie :
si $c^l_{N,k} > c^l_{N,k+1}$ alors $c^l_{N,k}$ est un maximum local.
Inversement, si $c^l_{N,k} < c^l_{N,k+1}$ alors $c^c_{N,p} > c^c_{N,p-1}$, donc $c^c_{N,p}$ est un maximum local.
Deuxième cas :

$$\sqrt{{c^l_{N,k}}^2 + {c^c_{N,p+1}}^2} < \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2} > \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p-1}}^2}$$

on obtient que $c^c_{N,p}$ est un maximum local.
Troisième cas :

$$\sqrt{{c^l_{N,k+1}}^2 + {c^c_{N,p}}^2} \leq \sqrt{{c^l_{N,k}}^2 + {c^c_{N,p}}^2} \geq \sqrt{{c^l_{N,k-1}}^2 + {c^c_{N,p}}^2}$$

on obtient immédiatement que $c^l_{N,k}$ est un maximum local.
Une étude identique pourrait être menée en considérant les minima locaux.■

1.7 Formule de reconstruction

Jusqu'à présent, nous n'avons pas exhibé de formule de reconstruction de la fonction f, ces formules ne figurent pas dans la littérature et rendent la décomposition de Berkner plus intéressante. Nous verrons en quoi l'approche proposée diffère des approches classiques de décomposition qui utilisent des analyses multirésolution. La décomposition semble, en apparence, proche de celle de Sweldens [91], connue sous le nom de "lifting scheme", mais cette ressemblance cache une différence majeure que nous discuterons dans la section suivante. En effet, tout signal f discret peut se décomposer sous la forme :

$$f(k) = \frac{1}{2}(f(k) + f(k+1)) + \frac{1}{2}(f(k) - f(k+1)) = d^1_{0,k} + c^1_{0,k}$$

On définit ensuite, pour N positif :
si $N+1$ pair:

$$c^1_{N,k} = \frac{1}{2}(c^1_{N-1,k} + c^1_{N-1,k-1})$$

$$d^1_{N,k} = \frac{1}{2}(d^1_{N-1,k} + d^1_{N-1,k-1})$$

si $N+1$ impair:

$$c^1_{N,k} = \frac{1}{2}(c^1_{N-1,k} + c^1_{N-1,k+1})$$

$$d^1_{N,k} = \frac{1}{2}(d^1_{N-1,k} + d^1_{N-1,k+1})$$

Alors, pour $N = 1$, on a :

$$d^1_{1,k} + c^1_{1,k} = \frac{1}{2}(d^1_{0,k} + d^1_{0,k-1}) + \frac{1}{2}(c^1_{0,k} + c^1_{0,k-1})$$

$$= \frac{1}{4}(f(k)+f(k+1)+f(k-1)+f(k)+f(k)-f(k+1)+f(k-1)-f(k)) = \frac{1}{2}(f(k)+f(k-1)) = d^1_{0,k-1}$$

De même, pour N = 2, on montre facilement que :

$$d^1_{2,k} + c^1_{2,k} = d^1_{1,k}$$

Puis, par une récurrence immédiate, on montre que :

$$d^1_{N,k} + c^1_{N,k} = d^1_{N-1,k} \text{ si } N+1 \text{ impair}$$

$$d^1_{N,k} + c^1_{N,k} = d^1_{N-1,k-1} \text{ si } N+1 \text{ pair}$$

On obtient, donc, la formule de reconstruction suivante pour f (à la profondeur N) :

$$f(k) = d^1_{N,k+\frac{N+1}{2}} + \sum_{l=0}^{N} c^1_{l,k+\frac{l+1}{2}}$$

Les coefficients $d^1_{N,k+\frac{N+1}{2}}$ correspondent à la convolution du signal f avec une gaussienne dilatée par un facteur $\frac{\sqrt{N+1}}{2}$. En effet, d_0 correspond à la convolution du signal avec $\rho_0(j)$, puis les formules qui permettent de passer d'une échelle à l'autre sont les mêmes que celles introduites en section 6.2. On visualise, à ce propos, en Figure 1.6 , la convergence de la séquence $\frac{\sqrt{N+1}}{2}\tilde{\rho}_N(j)$ vers la gaussienne $\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{j}{\sqrt{N+1}/2})^2}$ pour différentes valeurs de N. Cette étude permet de déduire que l'on projette sur des espaces d'approximation et de détails.

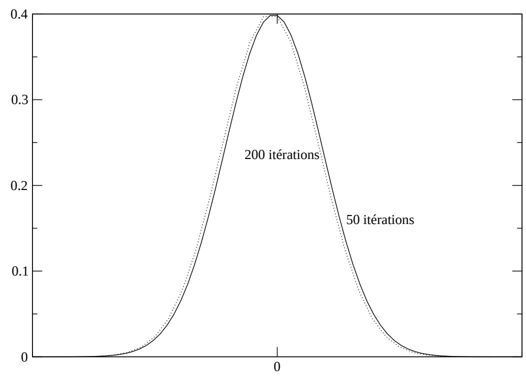

FIG. 1.6 – Convergence de la séquence $\frac{\sqrt{N+1}}{2}\tilde{\rho}_N(j)$ vers la gaussienne $\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{j}{\sqrt{N+1}/2})^2}$ pour $N = 50$ et $N = 200$

1.8 Comparaison avec les décompositions usuelles

Une propriété originale de la décomposition précédente est que le calcul des détails à l'échelle N, les coefficients $c^1_{N,k}$, utilise l'approximation à l'échelle N et à l'échelle $N-1$. Tous les schémas proposés par Sweldens [91], utilisent pour le calcul des détails à l'échelle N uniquement les approximations à l'échelle $N-1$. Prenons l'exemple de la base de Haar ; supposons le signal $d_{k,J}$ initial de longueur 2^J, alors on peut définir le schéma suivant :

$$\left\{ \begin{array}{lcl} d_{k,l-1} & = & \frac{1}{2}(d_{2k,l} + d_{2k+1,l}) \\ c_{k,l-1} & = & d_{2k,l} - d_{2k+1,l} \end{array} \right.$$

On décompose, dans ce cas le signal, sur deux sous-espaces orthogonaux à chaque itération de l'algorithme ; il n'existe donc pas de relation donnant les coefficients de détails à une échelle en fonction des coefficients de détails à l'échelle précédente. On ne peut donc pas suivre les maxima d'une échelle à l'autre. Cette remarque peut-être étendue à toutes les décompositions sur des bases orthonormées d'ondelettes.

1.9 Sous-espaces associés à la décomposition et propriétés de conservation

1.9.1 Décomposition sur des sous-espaces

Posons $g_0 = \{0.5, -0.5\}$, et $h_0 = \{0.5, 0.5\}$, alors les filtres utilisés peuvent être vus comme :

$$\forall N \geq 1, \quad g_N(j) = h * g_{N-1}(j)$$

et :

$$\forall N \geq 1, \quad h_N(j) = h * h_{N-1}(j)$$

Considérons alors les espaces engendrés par h_N et g_N :

$$W_N = \{g_N[n-l]\}_{l\in\mathbb{Z}}$$

et

$$V_N = \{h_N[n-l]\}_{l\in\mathbb{Z}}$$

On obtient, donc, que $W_N \subset V_{N-1}$ et $V_N \subset V_{N-1}$, d'après la commutativité de la convolution. La formule de reconstruction du signal ne supposant aucune hypothèse sur celui-ci, on peut écrire que :

$$S = V_N + \sum_{k=0}^{N} W_k$$

où S est l'ensemble des suites de $\mathbb{Z}$.
Les sous-espaces en question ne sont bien sûr pas orthogonaux.

1.9.2 Propriété de conservation de la transformation

Nous vérifions, dans cette partie, les propriétés usuelles que l'on souhaite retrouver sur l'approximation et sur les détails, afin de faire une dernière analogie avec les bases d'ondelettes.
On démontre, facilement, que :

$$\sum_k c^1_{0,k} = 0$$

et

$$\sum_k d_{0,k} = \sum_k f(k)$$

Puis, par récurrence, on obtient que :

$$\sum_k c_{N,k} = 0$$

et que :

$$\sum_k d_{N,k} = \sum_k f(k)$$

2

Application de la décomposition de Berkner à la détection d'évenements dans les signaux EMG

2.1 Description des signaux et intérêt de la décomposition

Dans cette partie, nous allons montrer comment la décomposition de Berkner permet de caractériser la nature des changements dans des signaux EMG (électromyogramme). Nous appliquerons une méthode fondée sur cette décomposition pour mesurer le temps de latence réflexe du muscle génioglosse. Après une brève présentation des signaux EMG, nous insisterons sur les possibilités qu'offrent une étude de tels signaux à l'aide d'un algorithme de détection d'extrema. Nous comparerons à ce sujet les résultats obtenus avec d'autres méthodes existantes.

2.1.1 Mesure de la latence réflexe de l'activité du muscle génioglosse

Le génioglosse est un muscle extrinsèque de la langue. C'est un muscle dilatateur du pharynx qui a de fortes implications dans le phénomène d'apnée du sommeil. L'application brutale d'une pression négative au niveau pharyngé est à l'origine d'une augmentation d'activité électromyographique de ce muscle. Cette réponse rapide (temps de latence compris entre 50 ms et 100 ms) apparaît comme réflexe [43].

On cherche ici à mesurer avec précision le temps de latence réflexe du muscle génioglosse suite à l'application d'un stimulus en pression. Le signal EMG mesuré est filtré analogiquement entre 20 Hz et 400 Hz puis échantillonné à la fréquence de 1000 Hz.

2.1.2 Spécificité des signaux

La technique utilisée pour mesurer l'activité électromyographique (EMG) du génioglosse s'effectue au niveau de la langue par voie de surface. Cette technique est peu invasive mais présente le désavantage d'être globale et peu précise. Cela se traduit par une absence de spécificité fréquentielle des signaux avant la stimulation par pression (activité tonique) et, aprés celle-ci (activité phasique, réflexe). Or, c'est la période de transition entre ces deux parties du signal qu'il est intéressant de chercher à déterminer. En Figure 2.1, on représente un signal EMG et le signal de pression. En Figure 2.2, on représente les réponses fréquentielles des deux signaux considérés.

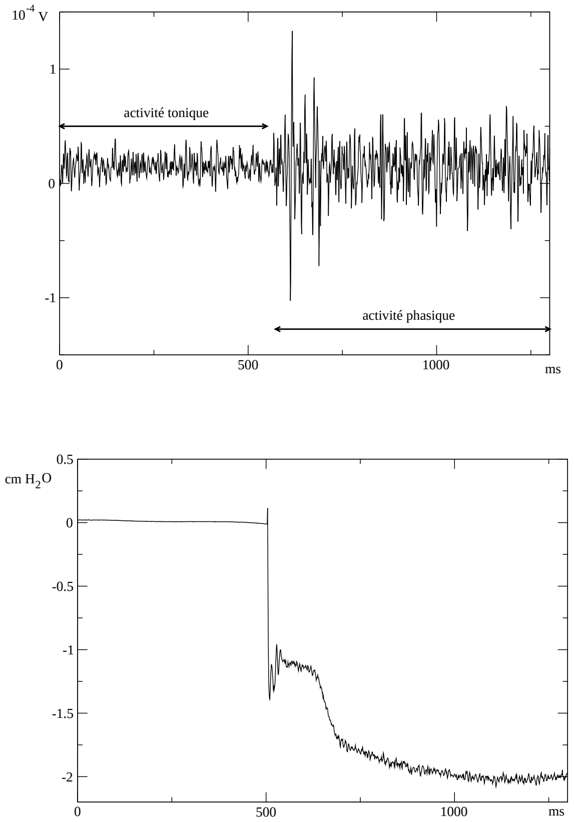

Fig. 2.1 – Exemple de signal EMG du génioglosse (en haut), soumis à un signal de pression (en bas)

2.1.3 Présentation des différentes méthodes de détermination du temps de latence

Nous présentons ici trois techniques différentes de détection de la transition entre les phases d'activité tonique et phasique.

Technique rudimentaire basé sur l'EMG seul :

Le principe de la technique utilisée est de faire la moyenne du signal sur des fenêtres dont la taille varie entre 10 et 100 ms, ces temps étant choisis en rapport avec l'ordre de grandeur du phénomène que l'on cherche à détecter.

Le changement de phase est déterminé en calculant l'écart-type des coefficients du signal sur la partie pré stimulus (σ_{ref}) puis en déterminant la première fois où la moyenne dépasse deux fois σ_{ref} dans la partie post stimulus [45].

Utilisation des ondelettes :

Le principe de l'étude est de regarder si la projection sur certains sous-espaces de détails associés à une décomposition en ondelettes ne peuvent pas discriminer correctement les phases tonique et réflexe.

Les problèmes qui se posent alors sont ceux du choix de la base d'ondelettes pour la décomposition, puis, du choix des échelles à retenir.

Dans [56], le choix de l'ondelette est abordée de manière empirique et les espaces à conserver sont déterminés en fonction de l'énergie des coefficients de la décomposition sur ces espaces. Dans cet

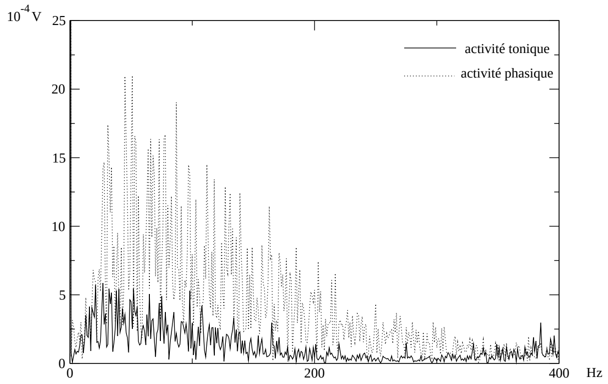

FIG. 2.2 – Analyse en fréquence des phases tonique et phasique du signal

article, portant sur l'analyse d'EMG utérin, l'étude en fréquence des signaux est pertinente dans la mesure où la spécifité fréquentielle des différentes composantes du signal étudié est facile à mettre en évidence.

2.2 Algorithme de détection d'extrema et changement de phase

Le problème ne se pose pas, comme dans [56], en terme de choix d'échelles à conserver dans une décomposition particulière en ondelettes car il n'y a pas d'échelle caractéristique du signal. Dans l'étude que nous allons mener, nous allons montrer que la détection des changements dans la nature du signal enregistré peut s'étudier grâce au changement de nature des extrema locaux des coefficients de la décomposition de Berkner. Nous faisons l'hypothèse que le comportement intéressant que l'on souhaite mettre en évidence se caractérise par l'apparition d'extrema locaux de plus forte amplitude et dont la durée de vie le long des échelles est sensiblement plus importante ; cette hypothèse est vérifiée sur les signaux que nous avons étudiés et l'on en donne une illustration en Figure 2.3 . Ceci est justifié par le fait que le début de l'activité réflexe du génioglosse se caractérise par une activité électrique plus importante. L'idée de notre étude se rapproche un peu de celle faite par Rosenfeld [86], qui avait tenté d'analyser des images en étudiant des propriétés sur les extrema des niveaux de gris (ou en une dimension, les extrema du signal).

2.2.1 Décomposition du signal

On considère un signal EMG que l'on filtre en utilisant l'algorithme décrit au chapitre précédent. Les extrema locaux disparaissent progressivement lorsque la profondeur de décomposition N augmente. Leur disparition plus ou moins rapide caractérise l'isolement d'un extremum par rapport aux extrema voisins. Par ailleurs, l'amplitude des extrema locaux est aussi un paramètre important du signal. Nous allons construire un modèle qui prend en compte les paramètres d'amplitude des extrema et de persistence des extrema le long des échelles, pour détecter les changements d'activité.
La profondeur de décomposition est déterminée par un paramètre qui indique le nombre minimal d'extrema à conserver dans l'approximation à l'échelle N par rapport au nombre d'extrema dans le signal initial.

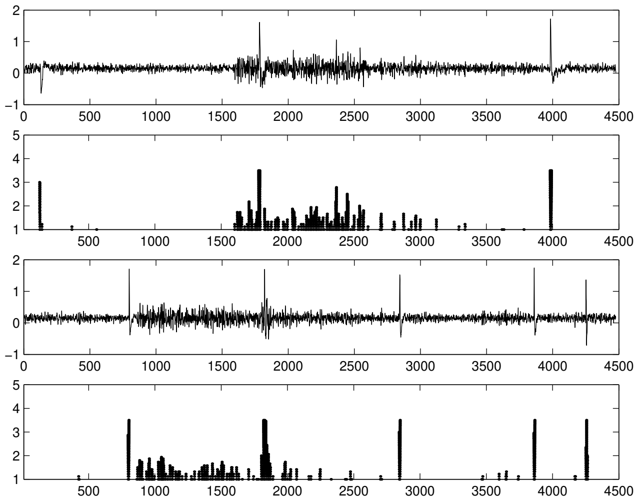

FIG. 2.3 – Première et troisième ligne : signaux EMG ; deuxième ligne et quatrième ligne : chaîne d'extrema se terminant lorsque le maxima correspondant devient infèrieur à 0.05, le nombre d'échelle considérée est $N = 50$

On ne considère, dans le signal initial, que les extrema ainsi déterminés, on note $E(N)$ cet ensemble. On définit, ensuite :

$$m(x) = \text{ médiane du vecteur x}$$

Sur des voisinages de taille variable, on calcule la médiane des valeurs des extrema qui appartiennent à $E(N)$. C'est à dire qu'en tout point i du signal on détermine la médiane du vecteur formé des coefficients de l'ensemble :

$$D(i) = \{c^1_{0,k}, k \in E(N), k \in V(i)\}$$

Le filtrage médian est choisi dans la mesure où le signal peut contenir des valeurs aberrantes qui faussent le calcul (en particulier, le mouvement des électrodes). Le principe de détection du début de l'activité réflexe est alors identique aux méthodes présentées précédemment. On calcule l'écart-type du paramètre retenu sur la phase pré stimulus puis l'on considère comme début d'activité réflexe le point où pour la première fois le paramètre dépasse deux fois cet écart-type. Les voisinages utilisés pour le calcul de la médiane sont de taille variable, de l'ordre de 10 ms à 100 ms.

2.2.2 Résultats et justification de la méthode

Dans cette section, on cherche à déterminer, pour différents signaux, à l'aide du critère D, le début de l'activité réflexe. Les paramètres de notre modèle sont la taille du voisinage et la proportion d'extrema conservés.
Ce dernier paramètre est lié à la profondeur de décomposition en ondelettes. En effet, lorsque l'on décompose un signal sur une base d'ondelettes, le nombre d'extrema locaux diminue lorsque l'échelle augmente ; l'avantage de la méthode est qu'elle offre une discrétisation très fine des échelles. Si l'on décompose le signal sur une base d'ondelettes se pose le problème du choix des espaces à conserver ; la discrétisation fine des échelles permet de contourner ce problème.
Nous allons maintenant regarder, sur des signaux réels, comment influent les différents paramètres

sur la détermination du début de l'activité réflexe. Etudions en effet, les signaux de la Figure 2.4, qui sont enregistrés de façon synchrones par deux paires d'électrodes très voisines, et regardons l'influence de la taille du voisinage V et de la proportion d' extrema que l'on conserve sur le calcul du temps de réaction du muscle.

Le principe est de calculer sur la partie prestimulus, l'ecart-type et la moyenne de D, puis de déterminer la première fois que celui-ci s'écarte de trois fois l'écart-type dans la partie poststimulus. Ce point sert alors au calul du temps de latence. On présente en Figure 2.5, les temps de latence observés avec le modèle utilisé, pour différents voisinages et des proportions d'extrema conservées différentes. Les signaux étudiés correspondant à deux enregistrements d'un même événement, les temps de latence mesurés doivent être les mêmes. Par ailleurs, avec la méthode utilisée, on constate qu'un voisinage trop petit entraine une mauvaise estimation des paramètres, du fait que le nombre d'extrema alors présent dans le voisinage est trop faible pour que la médiane contienne une information pertinente. La fenêtre la plus petite permettant d'avoir un résultat fiable est de l'ordre de 20 ms. Il apparaît que le temps de latence se stabilise lorsque l'on diminue le nombre d'extrema que l'on souhaite conserver. L'étude du temps de latence pour une taille de voisinage donné va nous permettre de déterminer l'échelle la mieux adaptée à notre problème, correspondant à des temps de latence les plus similaires pour les deux électrodes ; on retient alors comme temps de latence, la moyenne des temps de latence mesuré à cette échelle. En Figure 2.6, on regarde comment le temps de latence évolue en fonction du nombre d'extrema conservés, pour les deux enregistrements. Cette étude nous permet de conclure que le temps de latence est de 63 ou 64 ms. Nous constatons, donc, que faire porter l'étude sur plusieurs échelles permet d'augmenter la précision du résultat.

2.3 Conclusion

Cette étude nous a permis de mettre en évidence l'intérêt de l'analyse multi-échelles, et de son couplage avec une approche multicapteurs, pour l'étude du temps de réaction proposé. Le schéma proposé permet une détermination automatique du temps de latence, ce qui n'est pas le cas dans le cas de l'utilisation de méthodes par ondelettes où l'on doit déterminer dans un premier temps l'ondelette qui convient le mieux puis le sous-espace de détails le mieux adapté au calcul du temps de latence. Par ailleurs, ces techniques pourraient être appliquées à d'autres types de signaux, l'étude étant très générale. Enfin, notre modèle semble mieux discriminer l'activité tonique de l'activité phasique dans la mesure où nous avons considéré que le changement d'activité s'effectuait uniquement lorsque le critère retenu s'écarte de trois fois l'écart-type calculé sur la phase pré stimulus. Si l'on suppose que le critère retenu est gaussien sur chacune des deux phases, la détection du changement de phase est plus robuste que dans les méthodes proposées dans [45]. Nous avons enfin mis en évidence que la taille du voisinage est un paramètre essentiel de l'analyse et on s'attache toujours à utiliser le plus petit voisinage permettant d'avoir une estimation correcte du critère.

Cette technique ouvre des possibilités pour les méthodes multicapteurs et notamment pour l'électrophysiologie haute résolution par réseaux d'électrodes.

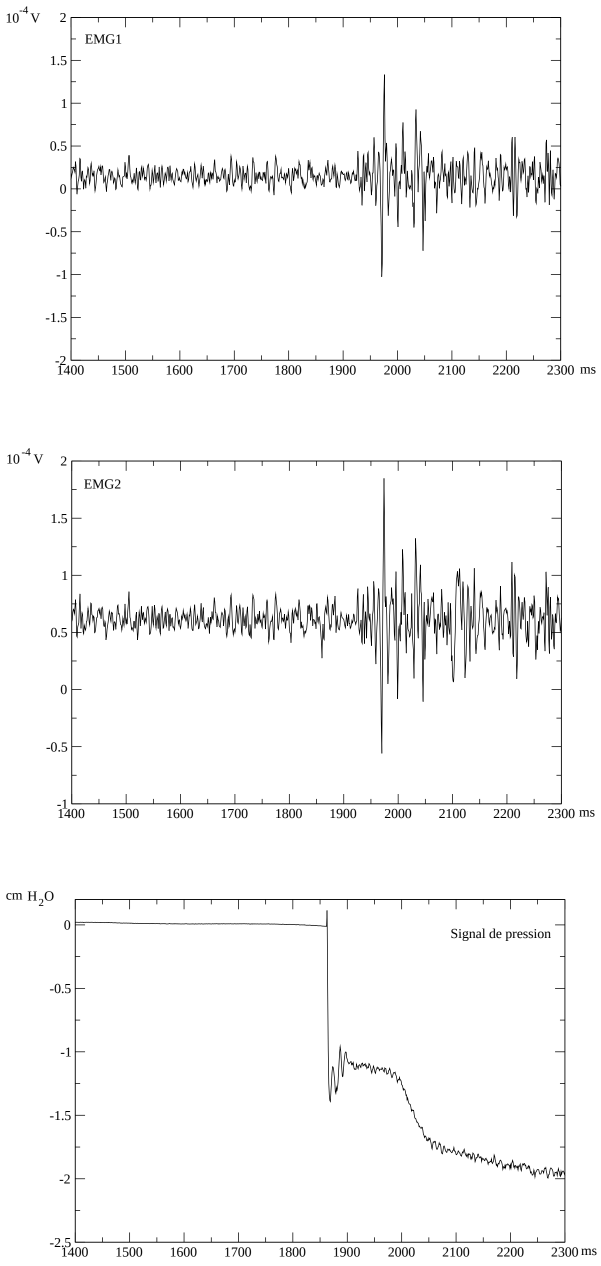

FIG. 2.4 – Signaux EMG étudiés (2 premiers graphes) et le signal de pression (dernier graphe)

EMG1:

taille du voisinage	41	21	11	5
temps de latence	61	63	69	71

taille du voisinage	41	21	11	5
temps de latence	61	63	63	66

taille du voisinage	41	21	11	5
temps de latence	71	68	68	64

EMG2:

taille du voisinage	41	21	11	5
temps de latence	62	61	2	69

taille du voisinage	41	21	11	5
temps de latence	60	61	2	64

taille du voisinage	41	21	11	5
temps de latence	68	65	1	62

FIG. 2.5 – Temps de latence mesuré pour différentes valeurs du voisinage et en conservant respectivement 30 % , 50 % et 100 % des maxima

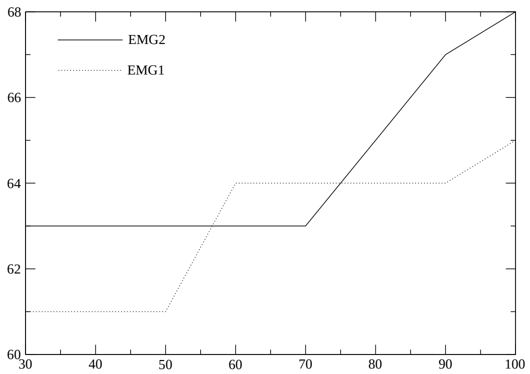

FIG. 2.6 – Détermination du temps de latence (en ordonnée), pour un pourcentage d'extrema que l'on conserve variable (en abscisse)

Troisième partie

Synchronisation d'oscillateurs neuronaux

Dans cette partie, nous présentons un travail indépendant qui porte sur l'étude d'oscillateurs neuronaux. L'objectif de l'article est, partant d'un modèle d'oscillateurs neuronaux, connus sous le nom d'oscillateurs de Wilson-Cowan, dont l'évolution est régie par deux équations différentielles, de voir si les comportements observés ne sont pas reproductibles avec un modèle d'oscillateurs dont l'évolution est discrète en temps.

Après avoir mis en évidence les différences entre les deux types de systèmes, nous regardons comment se comportent les oscillateurs lorsqu'on les connectent entre eux. Des propriétés intéressantes sont déduites en considérant des connexions synaptiques particulières pour lesquelles nous montrons que le réseau d'oscillateurs connectés se comportent comme un seul oscillateur, i.e. on observe une synchronisation des oscillateurs.
Nous présentons ici l'article publié dans Neural Networks [94].

Synchronization and Desynchronization of neural oscillators

Arnaud Tonnelier,Sylvain Meignen,Holger Bosch and Jacques Demongeot

Abstract

We have used continuous and discrete-time versions of a neural oscillator model to analyze how various types of synaptic connections between oscillators affect synchronization and desynchronization phenomena. First, we present a synthesis of the mathematical properties of both neural oscillator versions. Then, we show that the choice of parameters leads to a relationship between the two versions. Finally, we achieve the coupling of two oscillators in order to study how synaptic connections affect the phase lag. With this in mind, we state some of the results for the continuous-time model. The second part of this paper deals with the behavior of neural networks comprising connected oscillators, wich involves looking at the conditions for desynchronization of a totally synchronized oscillator net. Such a study has been carried out both for a fully and for a sparsely connected network. This leads to the observation that some architectures enable proper desynchronization when the size of the network is large. While searching for the conditions for desynchronization, we have discovered that a macroscopic description of the network is sometimes possible. To conclude , we discuss the advantages and the limitations of this macroscopic approach.

Keywords: Dynamical system, Neural oscillators, Synchronization, Desynchronization.

0.4 Introduction

Oscillatory electrical activities have been uncovered in several brain regions, such as the visual cortex, the olfactory bulb, studied by Li and Hopfield (1989), and the hippocampus (Gluck, 1996). Recordings carried out by Gray et al.(1989) on the first visual cortical area of anesthesized cats have shown that visual stimuli can generate brief, coherent electrical activity. These biological results are consistent with the theoretical investigations of Von der Malsburg (1981), which gave rise to the labelling hypothesis, i.e., cell assemblies are established through oscillations, and labeled by a phase. In accordance with this hypothesis, the electrical activity might be explained

by the fact that visual stimuli provoke brief phase locking of separate regions of the cortical area. Our purpose has therefore been to modelize this phenomenon by synchronization of neural oscillators followed by a fast desynchronization.
In order to do this, we were confronted with a choice between the use of continuous or discrete time for the oscillator network. Many people find discrete-time nets to be conceptually simpler, and more appropriate for dealing with discrete or symbolic data. On the other hand, continuous-time nets seem more natural in many models and applications. Since there is no biological evidence concerning the kind of network to be used, we will try to explore the labelling hypothesis through both continuous and discrete time approaches. To this end, we will use a common type of continuous-time dynamic describing additive nets:

$$\frac{dx_i}{dt} = -c_i x_i + \sum_{j=1}^{n} W_{ij} g_j(x_j), (i = 1, ..., n)$$

Here, g_j is the activation function of unit j; its value is the output of unit $j.c_i$ is a relaxation coefficient, W_{ij} is the weight of the connection line from unit j to unit i and $x_i(t)$ is the activity of unit i. Several investigations have been carried out concerning the synchronization aspect of the labelling hypothesis, using special values of the parameters leading to specific architecture. For a study of a ring network, we may refer to Atiya and Baldi (1989). A more systematic mathematical study is available in Hoppensteadt and Izhikevitch(1997). For discrete-time networks, we will use a general additive net of the form:

$$y_i(t+1) = g_i(\sum_{j=1}^{n} W_{ij} y_j), (i = 1, ..., n)$$

Here, y_i is the output of unit i. This model was first introduced by Hopfield (1982). Synchronization for such systems poses a difficult problem, from a mathematical point of view, and, as far as we know, little has been done to pursue this goal. However, synchronization can also be achieved artificially by the addition of inputs, as we will be explained later.
One major aim of our paper is to study desynchronization of additive nets after their synchronization for specific kinds of synaptic weights. To start with, we describe, in section 2, the basic unit of the network, the neural oscillator. In the following section, we study the mathematical properties of the oscillator, in order to illustrate the differences between continuous and discrete time approaches. In contrast, section 4 focuses on the possible relationship between the two descriptions. In the final section, we analyze the coupling of two oscillators, in order to obtain an insight into the desynchronization of more complex neural nets.

0.5 Models and Equations

This section is devoted to the presentation of a single oscillator. This comprises two neurons, one excitatory and one inhibitory as depicted in the following interaction diagram:

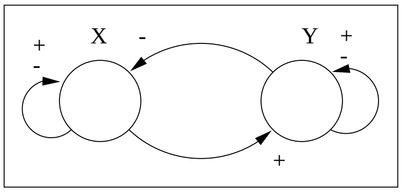

Figure 1 : Representation of two coupled oscillators, signs indicate whether the action in the direction of the arrow is inhibitory or excitatory.

X (or Y) can be either the activity of the excitatory (or inhibitory) neuron, or its output. >From now on, $U_x(t)$ (or $U_y(t)$) will denote the output of the excitatory (or inhibitory) neuron and $x(t)$ (or $y(t)$) will represent its activity. Such a modelization was first carried out by Wilson and Cowan (1972), in order to analyze the interaction between a population of inhibitory neurons and a population of excitatory neurons. Furthermore, this kind of interaction diagram has been widely used to interpret many regulated biological systems, particularly in genetic (Thomas and D'Ari 1989). In this paper, we consider two kinds of evolutional behavior. First, we describe a continuous-time evolution for U_x and U_y, given by:

$$\left\{ \begin{array}{rcl} \frac{dU_x(t)}{dt} & = & -\frac{U_x(t)}{\tau} + \tanh(\lambda U_x(t)) - \tanh(\lambda U_y(t)) \\ \frac{dU_y(t)}{dt} & = & -\frac{U_y(t)}{\tau} + \tanh(\lambda U_y(t)) + \tanh(\lambda U_x(t)) \end{array} \right. \quad (1)$$

where τ is a characteristic time constant and $\lambda > 0$ is the amplification gain. Such a system is a simple case of additive nets, as described in the introduction, in which synaptic connections are equal to 1 or -1. Furthermore, the system allows oscillatory behavior for $U_x(t)$ and $U_y(t)$.
Secondly, we also emphasize discrete-time systems owing to the fact that such systems take delays into account naturally (Chapeau-Blondeau and Chauvet, 1992). We therefore study the following Hopfield model for the evolution of $x(t)$ and $y(t)$:

$$\left\{ \begin{array}{rcl} x(t+1) & = & g(a_1 x(t) + a_2 y(t)) \\ y(t+1) & = & g(a_3 x(t) + a_4 y(t)) \end{array} \right.$$

where $g(y) = \tanh(y)$ and a_1, a_2, a_3 and a_4 are the synaptic weights. The above system is obviously a particular case of the discrete-time evolution system described in the introduction. In accordance with the symmetrical properties of the continuous-time version, but also for the sake of simplicity, we consider that $a_3 = -a_2$ and $a_4 = a_1$. Despite the restrictive nature of this choice, it may prove useful when comparing the two versions of the oscillator. Furthermore, in the present work, the study of neural networks comprising oscillators will show the relevance of this simplification. Thus, $x(t)$ and $y(t)$ are solutions of:

$$\left\{ \begin{array}{rcl} x(t+1) & = & \tanh(\alpha x(t) - \beta y(t)) \\ y(t+1) & = & \tanh(\beta x(t) + \alpha y(t)) \end{array} \right. \quad (2)$$

0.6 Oscillator properties

This section is divided into two parts; the first is devoted to the discrete-time version and the second involves its continuous counterpart. Let us first look at a behavioral analysis of the discrete-time system (2), when α and β are varied. Secondly, we will discuss classical results involving the bifurcation study and period calculation for the continuous version, which are also available in Atiya and Baldi (1989). In addition, we will try to shed light on the existence of fixed points. Finally, we will undertake an investigation of isochrons, which make it easier to understand the synchronization process.

0.6.1 Discrete-time oscillator

We have carried out a study of system (2), looking at the following points:
- In property 1, we seek the conditions that dictate whether the fixed point $(0,0)$ is repulsive or attractive.
- In property 2, we set supplementary conditions for the existence of other fixed points, and we then discuss their stability.
- In property 3, we calculate an approximation of the period.

Property 1 *The origin is a local attractive fixed point if and only if:*

$$\alpha^2 + \beta^2 < 1$$

Proof: From dynamical systems theory, it follows that near the origin, the behavior of system (2) is equivalent to:

$$\begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = Jac(0,0) \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

where $Jac(0,0)$ is the Jacobian matrix at the origin.

Therefore, the origin is attractive if and only if the spectral radius of the Jacobian matrix at the origin is less than 1, which reads:

$$\rho(Jac(0,0)) = \sqrt{\alpha^2 + \beta^2} < 1 \qquad \blacksquare$$

When $\alpha^2 + \beta^2 > 1$, the trajectories may tend towards a limit cycle, as shown in Figure 2 (as the rotation angle of trajectories around the cycle is not a rational number, the cycle comprises an infinite number of points). When the dynamic is of this sort, we speak of an oscillator with pacemaker activity, or a pacemaker oscillator.

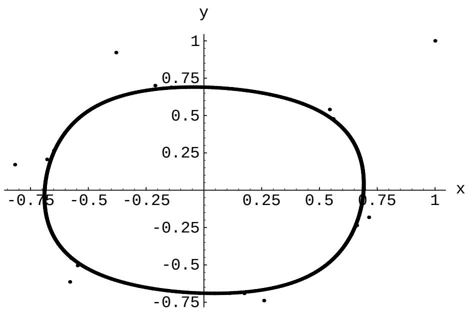

Figure 2: Trajectories in the (x,y) plane for values of α and β so that $\alpha^2+\beta^2 > 1$ ($\alpha = 0.6$ and $\beta = 1$) in order to obtain a limit cycle.

>From now on, we will refer to secondary Hopf bifurcation as the change in stability of the origin. Beyond the secondary Hopf bifurcation, the dynamic may also have fixed points:

Property 2 *The system can have either one, five or nine fixed points.*

Proof: A fixed point (x, y) satisfies the equilibrium system (ES):

$$\left\{ \begin{array}{lcl} x & = & \tanh(\alpha x - \beta y) \\ y & = & \tanh(\beta x + \alpha y) \end{array} \right.$$

hence, it also gives:

$$\left\{ \begin{array}{lclcl} y & = & \frac{1}{\beta}(\alpha x - \operatorname{argtanh}(x)) & = & f(x) \\ x & = & \frac{1}{\beta}(-\alpha y + \operatorname{argtanh}(y)) & = & -f(y) \end{array} \right.$$

With such a system, three cases may occur: we can have one solution, as depicted in figure 3(A), five solutions, when the two curves are tangential, and, possibly, nine solutions, as shown in figure 3(B):

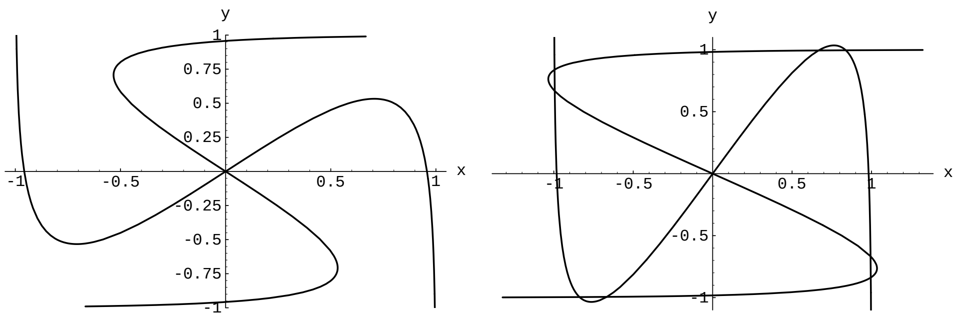

Figure 3: Representation of the nullclines of the discrete model in the (x, y) plane. Their intersections represent the fixed points of the system. In figure (A), the parameters are chosen so that there is only one fixed point, whereas, in figure (B), they are chosen so as to obtain nine fixed points.

Furthermore, we have:

$$f'(x) = \frac{1}{\beta}(\alpha - \frac{1}{1 - x^2})$$

>From this we can infer that:

1. if $0 \leq \alpha \leq 1$, f is a decreasing function, and thus $f \circ f$ is increasing, then $x + f \circ f(x) = 0$ has only one solution, $x = 0$, which implies that the only fixed point is (0,0)
2. if $\alpha > 1$

Since $f(x)$ is increasing for x in $[0, \sqrt{1 - \frac{1}{\alpha}}]$, and decreasing for x in $[\sqrt{1 - \frac{1}{\alpha}}, 1[$ we may deduce that:

- When the nullclines are tangential for some value of x where $x > 0$, we have a single fixed point (x, y) where $x > 0$ and $y > 0$. Furthermore, $x < \sqrt{1 - \frac{1}{\alpha}}$. The system has five fixed points because of the symmetry.
- When the nullclines cross for a value of $x > 0$, they then cross twice. We thus have two fixed points in the region $x > 0$ and $y > 0$, leading to nine fixed points because of the symmetry.

Let us assume that there are some fixed points. Of these fixed points, one has coordinates (x_1, y_1) that satisfy $y_1 > x_1$. As x_1 must be positive, we may deduce that $\alpha > \beta \quad \blacksquare$.
The above demonstration shows that, if $\beta \geq \alpha$ or $\alpha \leq 1$, we have no fixed point apart from the origin. Therefore, if α and β also satisfy $\alpha^2 + \beta^2 > 1$, we are in a cycle configuration. This also explains the dynamic of the oscillator in terms of bifurcation. Indeed, there is a first bifurcation corresponding to the change in stability of the origin. Another bifurcation occurs when a fixed point appears in each quadrant. Numerically, we have observed that it corresponds to a saddle-node on a limit cycle. This point will be developed in a further work.
Let us return to a limit cycle attractor configuration in order to state a simple property concerning the period of the oscillations.

Property 3 *Near the secondary Hopf bifurcation the period T is given by the formula:*

$$T = \frac{2\pi}{\theta}$$

$$\textit{where } \alpha + i\beta = r\ e^{i\theta}$$

Proof: It should be remembered that the evolution of the discrete-time oscillator at the origin and near the secondary Hopf bifurcation is governed by:

$$\begin{cases} x(t+1) &= \alpha x(t) - \beta y(t) \\ y(t+1) &= \beta x(t) + \alpha y(t) \end{cases}$$

So, let z be:

$$z(t) = x(t) + iy(t)$$

The evolution equation for z is therefore:

$$z(t+1) = (\alpha + i\beta) z(t) = r\ e^{i\theta} z(t) \blacksquare$$

It is obvious that this formula can only be used if T is really close to an integer.

Remark: The choice of fixed connection strengths is not biologically relevant. A possible improvement would be to introduce uncertainty in the parameters, i.e., to replace the classical iteration by an iteration of the following kind (studied by Demongeot et al. (1997)):

$$\begin{cases} x(t+1) &= \tanh_\epsilon(\alpha x(t) - \beta y(t)) \\ y(t+1) &= \tanh_\epsilon(\beta x(t) + \alpha y(t)) \end{cases}$$

where $\tanh_\epsilon(u)$ is defined by: $\exists \eta \in [-\epsilon, \epsilon] \ / \ \tanh_\epsilon(u) = \frac{1 - e^{2(1+\eta)u}}{1 + e^{2(1+\eta)u}}$
A possible description of the uncertainty involves making a random choice $\eta \in [-\epsilon, \epsilon]$ at each iteration. This allows us, at each step, to compute the classical iteration with $\alpha = \alpha_\eta = (1+\eta)\alpha$

and $\beta = \beta_\eta = (1+\eta)\beta$. Figure 4 shows this iteration for two values of ϵ.

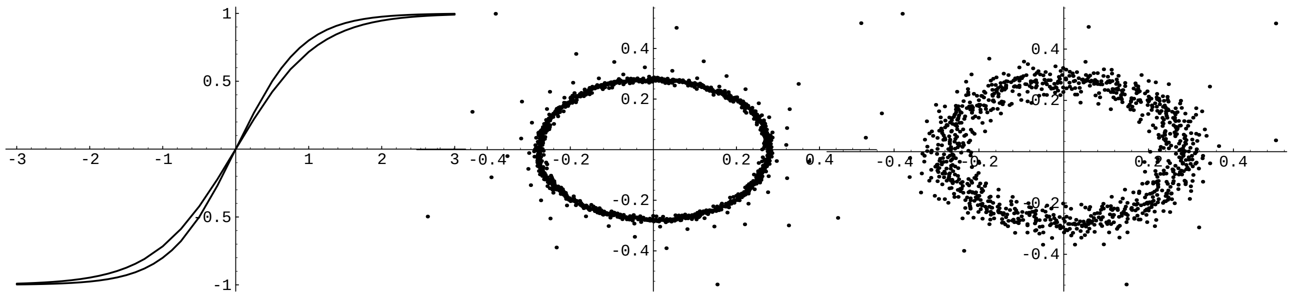

Figure 4: Figure (A) shows a sigmoïdal transfer function delimited by the two maps $\tanh((1-\epsilon)u)$ *and* $\tanh((1+\epsilon)u)$*. Figure (B) shows the trajectories for a compact set valued iteration defined by a random value* η *in* $[-\epsilon, \epsilon]$, $\epsilon = 0.01$, *and :*

$$\begin{cases} x(t+1) & = & \tanh((1+\eta)(\alpha x(t) - \beta y(t))) \\ y(t+1) & = & \tanh((1+\eta)(\beta x(t) + \alpha x(t))) \end{cases}$$

Figure (C) concerns the case for $\epsilon = 0.05$.

These phase diagrams show the structural stability of the iterations. The relevance of such a desynchronization model will be discussed at the end of the paper.

0.6.2 Continuous-time oscillator

Here is a reminder of the evolution of $(U_x(t), U_y(t))$:

$$\begin{cases} \frac{dU_x(t)}{dt} & = & -\frac{U_x(t)}{\tau} + \tanh(\lambda U_x(t)) - \tanh(\lambda U_y(t)) \\ \frac{dU_y(t)}{dt} & = & -\frac{U_y(t)}{\tau} + \tanh(\lambda U_y(t)) + \tanh(\lambda U_x(t)) \end{cases}$$

As with the discrete-time model, we first focus on the Hopf bifurcation characterization and period calculation. We then discuss the existence of fixed points. The part devoted to isochrons is of major importance in the understanding of the synchronization of oscillators.

The Hopf bifurcation

The Hopf bifurcation characterizes the change in dynamic for $(U_x(t), U_y(t))$. The evolution system changes from a single global attractor, the origin, to a cycle attractor. A mathematical criterion has been found out for this transition:

- If $\lambda\tau \leq 1$, the origin is the only stationary point. The origin is stable and all the trajectories converge to the origin (no limit cycles).
- If $\lambda\tau > 1$, the origin is unstable and all trajectories converge to a limit cycle attractor.

Proof for this theorem may be found in Atiya and Baldi (1989).

Period

In the case of small oscillations ($\lambda\tau$ close to 1), the period, T, can be approximated using the imaginary part of the eigenvalues of $Jac(0,0)$, where $Jac(0,0)$ is the Jacobian matrix of the system at the origin. This yields $T \approx \frac{2\pi}{\lambda} \approx 2\pi\tau$. This approximation is based on a linearization around the origin. As we move further away from the bifurcation point, $\lambda\tau = 1$, the first approximation of the period remains fairly good, whereas the second deteriorates more drastically.

Nullclines

For all values of λ and τ, the origin is the only fixed point. Indeed, as the nullclines are determined by the following equations:

$$\begin{cases} U_y &= -U_x + 2\tau \tanh(\lambda U_x) \\ U_x &= U_y - 2\tau \tanh(\lambda U_y) \end{cases}$$

we can show that $(0,0)$ is the only solution of this system.
Using the following change in variables:

$$\begin{cases} U_x &= \tau(x+y) \\ U_y &= \tau(x-y) \end{cases}$$

we seek the solutions of the system:

$$\begin{cases} y &= -x + \mu \text{ argtanh } (x) \\ x &= y - \mu \text{ argtanh } (y) \end{cases}$$

where $\mu = \frac{1}{\lambda\tau}$, $-1 < x < 1$ and $-1 < y < 1$.

- <u>if $\mu \geq 1$</u>
  If $x > 0$, then $y > 0$, from the first equation. But if $y > 0$, then $x < 0$, from the second equation which leads to a contradiction. A similar result can be derived for $x < 0$, and therefore $x = y = 0$ is the only fixed point.
- <u>if $\mu < 1$</u>
  We show that every half curve is located in a different region of the plane defined by the tangents at the origin ($y = (\mu - 1)x$ and $x = (1-\mu)y$), i.e., the half curve defined by:

$$\begin{cases} y = -x + \mu \text{ argtanh } (x) \\ x \in]0,1[\\ y \in]-1,1[\end{cases}$$

  satisfies $y - (\mu-1)x > 0$ and $x - (1-\mu)y > 0$. Similar results may be obtained by rotating the half curve under consideration through $k\frac{\pi}{2}$.
  Firstly, $y - (\mu-1)x > 0$ since $x \to -x + \mu \text{ argtanh}(x)$, $x \in [0,1[$ is convex. Secondly, the study of $g(x) = x - (1-\mu)y = (2-\mu)x - \mu(1-\mu) \text{ argtanh}(x)$ leads to $g(x) \geq 0$, $\forall x \in [0, x_1]$. We can then show that $y(x_1) = \frac{x_1}{1-\mu} > 1$, since $g(1-\mu) > 0$. Furthermore, studying the variations of $y(x)$, it is found to be increasing on $[x_1, 1[$, which gives $y(x) > 1$ if $x \geq x_1$. The half curve under consideration does not exist on $[x_1, 1[$, and therefore $g(x) > 0$. This ends the demonstration.

Isochrons

The use of isochrons to interpret biological phenomena was introduced by Winfree (1974). The first mathematical study of them dates back to Guckenheimer (1975). Isochron properties have been used to study nervous systems by both Pham Dinh et al. (1983), and by Ermentrout and Kopell (1994). For our study, we only need a general definition, which may be stated thus: if two initial conditions are part of the same isochron, then the solutions have identical asymptotic behavior.
Some isochrons of the continuous-time oscillator are shown in Figure 5.

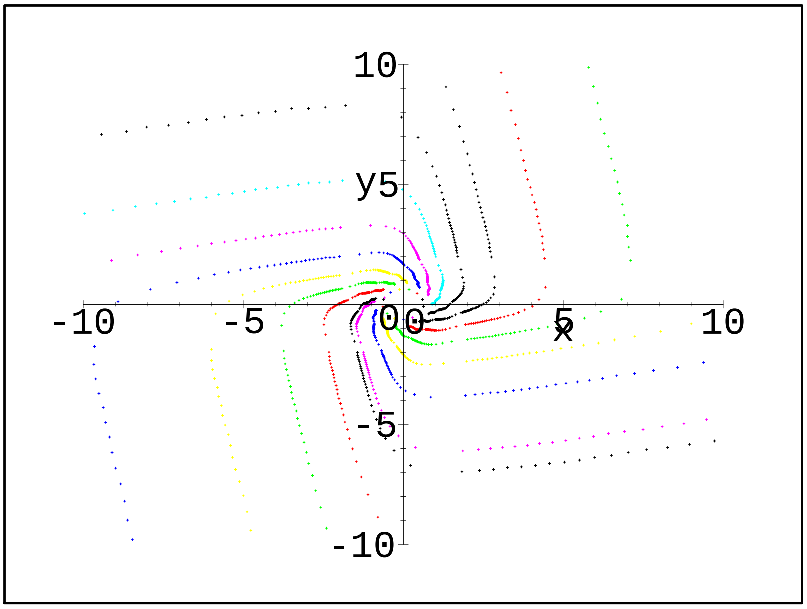

Figure 5: Representation of some isochrons in the (U_x, U_y) plane when $\lambda\tau > 1$.

Analysis of isochrons Isochrons of the neural oscillator follow a regular pattern resembling rectangular spirals. The main reasons for this shape are the symmetry of the system and its linearity for large values of U_x and U_y:

Property 4 *For $|U_x|, |U_y| > \frac{2}{\lambda}$, the system can be approximated using the following linear systems:*

$$\begin{array}{ccccc}
U_x > \frac{2}{\lambda} & , & U_y > \frac{2}{\lambda} & : & \frac{dU_x}{dt} \approx -\frac{U_x}{\tau} & , & \frac{dU_y}{dt} \approx -\frac{U_y}{\tau} + 2 \\
U_x > \frac{2}{\lambda} & , & U_y < -\frac{2}{\lambda} & : & \frac{dU_x}{dt} \approx -\frac{U_x}{\tau} + 2 & , & \frac{dU_y}{dt} \approx -\frac{U_y}{\tau} \\
U_x < -\frac{2}{\lambda} & , & U_y > \frac{2}{\lambda} & : & \frac{dU_x}{dt} \approx -\frac{U_x}{\tau} - 2 & , & \frac{dU_y}{dt} \approx -\frac{U_y}{\tau} \\
U_x < -\frac{2}{\lambda} & , & U_y < -\frac{2}{\lambda} & : & \frac{dU_x}{dt} \approx -\frac{U_x}{\tau} & , & \frac{dU_y}{dt} \approx -\frac{U_y}{\tau} - 2
\end{array}$$

Proof: For $|U_x|, |U_y| > \frac{2}{\lambda}$; $|\tanh(\lambda U_x)| \approx |\tanh(\lambda U_y)| \approx 1$. Hence, we obtain the above approximations.■
Since the four systems are identical through rotations of $\frac{\pi}{2}$, it is sufficient to study the isochrons in the first quadrant. Figure 6 shows two graphs that will make it easier to understand the

following demonstration:

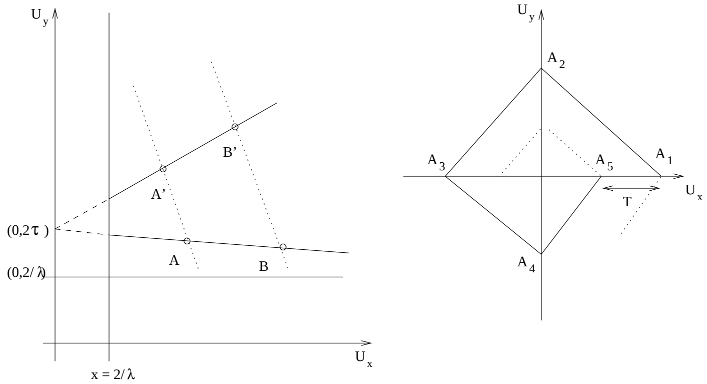

Figure 6: Figure(A) shows two portions, AA' and BB', of isochrons far beyond the point $(\frac{2}{\lambda}, \frac{2}{\lambda})$; thus the isochrons are drawn as lines. A and A' represent two points located on the same isochron, as do B and B'. Details of the geometrical shape are given in the demonstration. Figure(B) shows the drawing of an isochron far beyond $(\frac{2}{\lambda}, \frac{2}{\lambda})$ during a complete period.

Demonstration of the shape of the isochrons

For $U_x > \frac{2}{\lambda}$ and $U_y > \frac{2}{\lambda}$

Trajectories in the phase space follow: $U_y = kU_x + 2\tau (k \in \mathbb{R})$. Let us assume that part of an isochron is segment AA'. Let us then consider another isochron crossing the trajectories from A and from A' at B and B', respectively. We would like to show that (AA') is parallel to (BB'). Let us denote a1,a1',b1 and b1' as the respective abscissae of A,A',B and B'. Using the evolution equation of U_x, and given that the system requires the same time to go from A to A' as to go from B to B', it follows that:

ln(b1)-ln(a1)=ln(b1')-ln(a1') which is equivalent to $\frac{b1}{a1} = \frac{b1'}{a1'} \Rightarrow$ (BB')//(AA')

Thus, all isochrons are parallel in the four quadrants. We may now investigate how the different parts of the same isochron join on the axes. As the function $F(U_x, U_y) = (-\frac{U_x}{\tau} + \tanh \lambda U_x - \tanh \lambda U_y, -\frac{U_y}{\tau} + \tanh \lambda U_x + \tanh \lambda U_y)$ is continuous, the isochrons are also continuous, and therefore the straight segments in the different quadrants of the same isochron join on the axes. Since two points on the same trajectory separated by one time period are part of the same isochron, the resulting pattern forms a rectangular spiral.

For $|U_x| < \frac{2}{\lambda}$ and $|U_y| < \frac{2}{\lambda}$

Here, the mathematical study is much more complicated, but it has been proved by Guckenheimer (1975) that isochrons tend towards the cycle perpendicularly.

Slope of the spiral Since the linear parts of the isochrons are parallel in each quadrant, the slope of the isochrons is well defined. If the oscillator isochrons can be assumed to be a perfect rectangular spiral, as in Figure 6(B), we obtain:

Property 5 *The slope of the isochrons can be approximated by*

$$m = e^{\frac{T}{4\tau}}$$

where T is the period of the oscillator.

Proof: On the one hand, the distance between A_1 and A_5, in Figure 6(B), is given by the length of the trajectory covered during one period. As the system can be assumed to be linear, if $\|A_i\|$ denotes the distance from the origin to the point A_i, we obtain $T = \tau(\ln(\|A_5\|) - \ln(\|A_1\|))$, hence:

$$e^{\frac{T}{\tau}} = \frac{\|A_5\|}{\|A_1\|}$$

On the other hand, the above fraction can be expressed as

$$\frac{\|A_5\|}{\|A_1\|} = m^4$$

Where m is the slope of the isochrons when $U_x > 0$ and $U_y > 0$. The result of the property is a direct consequence of the above equalities.■

0.7 Relationship between the discrete-time and the continuous-time versions

In this part, we discuss the relationships between the parameters of the two versions. We therefore try to switch from the continuous model to the discrete one by making appropriate approximations. Using an Euler method with a time step equal to δ, the continuous model integrates as follows:

$$\begin{cases} U_x(t+\delta) &= (1-\frac{\delta}{\tau})U_x(t) + \delta(\tanh(\lambda U_x(t)) - \tanh(\lambda U_y(t))) \\ U_y(t+\delta) &= (1-\frac{\delta}{\tau})U_y(t) + \delta(\tanh(\lambda U_y(t)) + \tanh(\lambda U_x(t))) \end{cases}$$

We study this system close to the Hopf bifurcation point. It is well known that the size of the cycle shrinks as the values of the parameters approach the bifurcation point. Therefore, with an appropriate value of τ, it is possible to get a cycle, for which the diameter is small compared with $\frac{2}{\lambda}$. Thus, we can make the assumption that the respective values of U_x and U_y are asymptotically small compared to $\frac{2}{\lambda}$. This leads to the following simplification:

$$\begin{cases} U_x(t+\delta) &= (1+\delta\lambda(1-\frac{1}{\lambda\tau}))U_x(t) - \delta\lambda U_y(t) \\ U_y(t+\delta) &= (1+\delta\lambda(1-\frac{1}{\lambda\tau}))U_y(t) + \delta\lambda U_x(t) \end{cases}$$

Now, let $n\delta = 1$ and $\gamma = 1 + \delta\lambda(1-\frac{1}{\lambda\tau}) + i\delta\lambda = 1 + \frac{\lambda}{n}(1-\frac{1}{\lambda\tau}) + i\frac{\lambda}{n}$. Then:

$$\begin{cases} U_x(t+1) &= \mathrm{Re}(\gamma^n)U_x(t) - \mathrm{Im}(\gamma^n)U_y(t) \\ U_y(t+1) &= \mathrm{Im}(\gamma^n)U_x(t) + \mathrm{Re}(\gamma^n)U_y(t) \end{cases}$$

where Re and Im stand for the real part and the imaginary part.
Let us now return to the discrete-time system discussed in section 2, in which we use tanh as the synaptic transfer function. If we also hypothesize that x and y are sufficiently small, as $U_x(t) = x(t) - y(t)$ and $U_y(t) = x(t) + y(t)$, it follows:

$$\begin{cases} U_x(t+1) &= \alpha U_x(t) - \beta U_y(t) \\ U_y(t+1) &= \beta U_x(t) + \alpha U_y(t) \end{cases}$$

We may therefore deduce a relationship between paramaters for the discrete-time and continuous-time versions of the oscillator, which allows them to behave similarly:

$$\begin{aligned} \alpha &= \lim_{n \to +\infty} \mathrm{Re}(\gamma^n) \\ \beta &= \lim_{n \to +\infty} \mathrm{Im}(\gamma^n) \end{aligned}$$

A simple calculation gives us $\alpha = e^{\lambda - \frac{1}{\tau}} \cos(\lambda)$ and $\beta = e^{\lambda - \frac{1}{\tau}} \sin(\lambda)$.
We conclude from this that the equivalence of the two bifurcation equations corresponds to $\lambda\tau = 1$ and $\alpha^2 + \beta^2 = 1$.
Finally, we can investigate the period of the oscillations. It should be remembered that the period of the continuous-time oscillator equals $\frac{2\pi}{\lambda}$ whereas it is $\frac{2\pi}{\theta}$ for the discrete one ($\theta = arg(\alpha + i\beta)$). Since $\theta = \lambda$, it can be observed that the periods are identical at the bifurcation point.

0.8 Coupling two oscillators

In this section, we will first discuss the problem of an eventual phase lag for oscillators linked by one connection, the dynamic of the oscillators being either continuous or discrete in time. Then, we will discuss the case of two oscillators linked by more than one connection arriving at the conclusion that the phase lag is difficult to assess with a discrete-time evolution whereas, it can be derived from a continuous one.

0.8.1 One connection

Discrete model

We are going to discuss one excitatory connection between the two excitatory neurons, as shown in the following diagram:

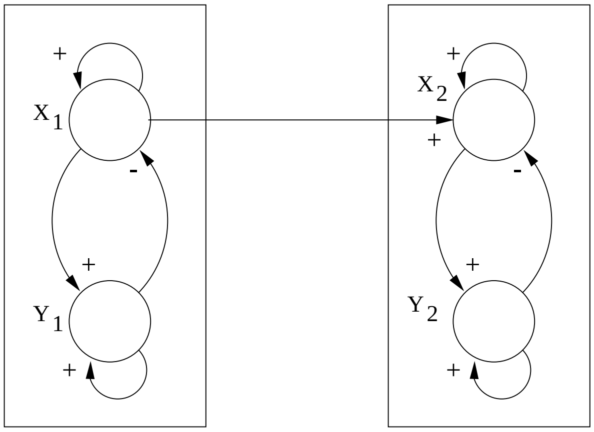

Figure 7: Representation of two modules with a connection between their excitatory neurons. The signs indicate whether the action in the direction of the arrow is inhibitory or excitatory.

Thus, the evolution of (x_1, y_1) is unchanged, whereas that of (x_2, y_2) is given by:

$$\begin{cases} x_2(t+1) &= \tanh(\alpha x_2(t) - \beta y_2(t) + \mathrm{c}\ x_1(t)) \\ y_2(t+1) &= \tanh(\beta x_2(t) + \alpha y_2(t)) \end{cases}$$

where $c > 0$.
At this stage, we have only carried out numerical simulations in which we have chosen $\beta > \alpha$, which means that the effect of one neuron on the other is greater than that exerted by the latter neuron on itself. Furthermore, when such a choice is made, the dynamic of an isolated oscillator presents no fixed points (see section 3.1). We expect that the dynamic described by (x_2, y_2) behaves similarly when c is small (this can be checked numerically). Let us denote $p(t)$ the phase lag between the oscillators at time t. We investigate the evolution of $(p(t), p(t+1))$ versus the synaptic weight, c. For a wide range of values of c, the phase lag between each oscillator is quasi-periodic (it might be periodic for special values of c); a limit cycle appears. If we proceed further, and study the evolution of this phase lag as a function of connection c, it appears that the stronger the connection, the bigger the cycle. This fact may be explained by a change in the frequency of one of the two connected oscillators: this change is all the greater since c is large. Figure 8 displays phase diagrams $(p(t), p(t+1))$ for several values of c (it is important to note that we are careful to maintain a pacemaker activity for both oscillators, whatever the value of parameter c).

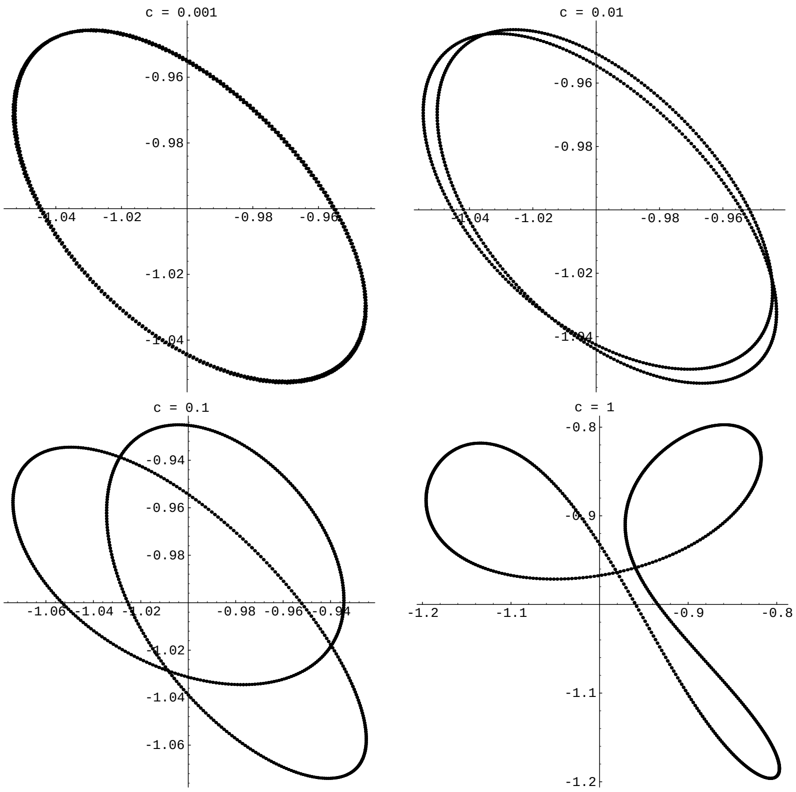

Figure 8: Figure (A) shows the phase diagram $(p(t), p(t+1))$ for a connection strength equal to 0.001 , $\alpha = 0.6$ *and* $\beta = 1$. *Figure (B) is similar to figure (A) for a connection strength equal to* 0.01, $\alpha = 0.6$ *and* $\beta = 1$. *In figure (C), the connection strength is equal to* 0.1, β *with* α *remaining unchanged. In figure (D), the connection strength is equal to* 1, β *and* α *remaining unchanged.*

Throughout the simulations, it appears that such a behavior may easily be reproduced for any other kind of connection.

Continuous-time version

Here, as with the discrete-time system, the excitatory neurons are coupled by a uni-directional excitatory connection. The evolution of the first oscillator remains unchanged, and that of the second gives:

$$\begin{cases} \frac{dU_{x_2}(t)}{dt} &= -\frac{U_{x_2}(t)}{\tau} + \tanh(\lambda U_{x_2}(t)) - \tanh(\lambda U_{y_2}(t)) + \tanh(cU_{x_1}(t)) \\ \frac{dU_{y_2}(t)}{dt} &= -\frac{U_{y_2}(t)}{\tau} + \tanh(\lambda U_{y_2}(t)) + \tanh(\lambda U_{x_2}(t)) \end{cases} \tag{3}$$

where $c > 0$.
Simulations show that the phase lag between the oscillators converges to a constant, d. Furthermore, d depends neither on the initial conditions of the system, nor on λ and τ. The coupling strength, c, only affects the time needed to establish the asymptotic phase lag. The value of d approaches $T/8$ if the system parameters approach the bifurcation point $\lambda\tau = 1$, where T is the period of the system. The analysis of system (3) makes it possible to determine analytically the effect of any possible connection between two units of two different oscillators on their phase lag. Details are given in the following property:

Property 6 *If we assume that the phase lag induced by the first oscillator on the second converges to a constant value, d, the phase lag induced by any other type of connection equals one of the values,* $\pm d + k\frac{T}{4}$ *where* $k \in \{0,1,2,3\}$*, provided that the strength of this connection is low.*
Proof: Let us first consider an oscillator (U_x, U_y), evolving as follows

$$\begin{cases} \frac{dU_x(t)}{dt} &= -\frac{U_x(t)}{\tau} + \tanh(\lambda U_x(t)) - \tanh(\lambda U_y(t)) \\ \frac{dU_y(t)}{dt} &= -\frac{U_y(t)}{\tau} + \tanh(\lambda U_y(t)) + \tanh(\lambda U_x(t)) \end{cases}$$

The invariance shown when the vector field is rotated through $\frac{\pi}{2}$ gives:

$$\begin{aligned} &1) U_x(t) = U_y(t + \tfrac{T}{4}) \\ &2) - U_x(t) = U_x(t + \tfrac{T}{2}) \\ &3) - U_y(t) = U_y(t + \tfrac{T}{2}) \end{aligned}$$

If we consider that the excitatory connection, c, in (3), is small, those relationships are still valid for the excited oscillator. It is, therefore, possible to imply the following:

- If a connection is replaced by an identical connection, but in the opposite direction, the phase difference is replaced by its opposite value, i.e., $d* = -d \bmod T$
- if a connection changes its sign, i.e., an excitatory connection becomes inhibitory or vice versa, the phase difference is increased by $\frac{T}{2}$, i.e., $d* = d + \frac{T}{2} \bmod T$
- If a connection originated from the excitatory neuron is replaced by an identical connection originating from the inhibitory neuron, the phase lag is increased by $\frac{T}{4}$, i.e., $d* = d + \frac{T}{4} \bmod T$

– If a connection targeting the excitatory neuron is replaced by an identical connection targeting the inhibitory neuron, the phase difference is decreased by $\frac{T}{4}$, i.e., $d* = d - \frac{T}{4}$ mod T

The phase difference induced by any connection is obtained by applying one or more of the above rules.■

0.8.2 Several connections

Discrete model

The connections between the oscillators are as follows:

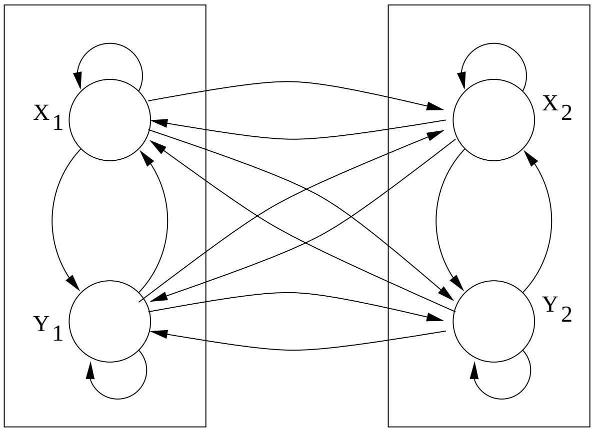

Figure 9: Representation of two completely connectd modules; an arrow represents a connection between two neurons. The action of one neuron to another is not specified.

Simulations are made for the connection strength evolving in $[c_{min}, c_{max}]$. When c_{min} equals c_{max}, the oscillators tend to synchronize. In contrast, if c_{min} is different from c_{max}, the phase lag between oscillators, $p(t)$, is such that map $p(t+1)$ versus $p(t)$ is a cycle. Numerically, we may observe that the bigger the difference between c_{max} and c_{min}, the bigger this cycle. The dynamic properties of the system are similar to these obtained with one connection; $(p(t), p(t+1))$ tends towards the cycle exponentially faster when the connection strength increases.

Continuous model

The evolution system has the following form:

$$\left\{ \begin{array}{lcl} \frac{dU_{x_i}(t)}{dt} & = & -\frac{U_{x_i}(t)}{\tau} + \tanh(\lambda U_{x_i}(t)) - \tanh(\lambda U_{x_i}(t)) + \sum_{i \neq j} \tanh(c_{x_i x_j} U_{x_j}(t) + c_{x_i y_j} U_{y_j}(t)) \\ \frac{dU_{y_i}(t)}{dt} & = & -\frac{U_{y_i}(t)}{\tau} + \tanh(\lambda U_{y_i}(t)) + \tanh(\lambda U_{x_i}(t)) + \sum_{i \neq j} \tanh(c_{y_i x_j} U_{x_j}(t) + c_{y_i y_j} U_{y_j}(t)) \end{array} \right.$$

where i and j are equal to 1 or 2. If the coupling between two oscillators comprises several connections, simulations show that the phase lag is still asymptotically stable. If the oscillators are near the bifurcation point, the connections can be classified into four classes, according to the phase lag they induce. Connections inducing opposite phase lags ($T/8$ is opposite to $5T/8$, and $3T/8$ is opposite to $7T/8$) anneal each other and the total effect of these connections is negligible. The weighted mean value of the remaining connections equals the phase lag induced by the set of connections. In particular, if all connections anneal each other, the oscillators act almost independently, and the phase lag depends on the initial conditions of the system. Hoppensteadt and Izhikevich (1997) have also studied this kind of problem using their "canonical" model, which has given similar results. They have found that with the appropriate connections, the phase lag can be monitored, and, in addition, they have shown that some connection diagrams enable the oscillators to behave independently.

0.9 Network of connected oscillators

We have just shown that the phase lag between two oscillators may be monitored using a continuous-time evolution, but no such conclusion is valid for the discrete-time model, even close to the bifurcation point. Nevertheless, as the number of oscillators increases, the phase lag between oscillators can no longer be assessed using continuous-time evolution. For this reason, synchronization is achieved by means of external stimuli. Using discrete-time evolution systems, we have investigated the desynchronization aspect of the labelling hypothesis. Such a study could also have been carried out with the continuous-time version.

0.9.1 Dynamic of the network

We now consider a network comprising N oscillators, whose dynamic gives:

$$\left\{\begin{array}{lcl} x_i(t+1) & = & a_i \tanh\{\alpha_i x_i(t) - \beta_i y_i(t) + \sum\limits_{j=1, j\neq i}^{N} (c_{x_i x_j} x_j(t) + c_{x_i y_j} y_j(t))\} \\ \\ y_i(t+1) & = & a_i \tanh\{\beta_i x_i(t) + \alpha_i y_i(t) + \sum\limits_{j=1, j\neq i}^{N} (c_{y_i x_j} x_j(t) + c_{y_i y_j} y_j(t))\} \end{array}\right. \tag{4}$$

where $i \in \{1..N\}$ and (x_i, y_i) describes the electrical response of the excitatory and inhibitory neurons for the i^{th} oscillator. The synaptic weights follow Dale's principle, i.e., c_{yx} and c_{xx} are both positive, and, c_{xy} and c_{yy} are both negative. Moreover, a simple change in variable makes it possible to consider a_i as being equal to 1. At a given time, t, we apply an input characterized by (r, θ), to each oscillator, giving the following dynamic:

$$\left\{\begin{array}{lcl} x_i(t+1) & = & \tanh\{\alpha_i x_i(t) - \beta_i y_i(t) + \sum\limits_{j=1, j\neq i}^{N} (c_{x_i x_j} x_j(t) + c_{x_i y_j} y_j(t))\} + r\cos(\theta) \\ y_i(t+1) & = & \tanh\{\beta_i x_i(t) + \alpha_i y_i(t) + \sum\limits_{j=1, j\neq i}^{N} (c_{y_i x_j} x_j(t) + c_{y_i y_j} y_j(t))\} + r\sin(\theta) \end{array}\right.$$

the next stages are calculated as follows:

$$\left\{\begin{array}{lcl} x_i(t+1) & = & \tanh\{\alpha_i x_i(t) - \beta_i y_i(t) + \sum\limits_{j=1, j\neq i}^{N} (c_{x_i x_j} x_j(t) + c_{x_i y_j} y_j(t))\} \\ y_i(t+1) & = & \tanh\{\beta_i x_i(t) + \alpha_i y_i(t) + \sum\limits_{j=1, j\neq i}^{N} (c_{y_i x_j} x_j(t) + c_{y_i y_j} y_j(t))\} \end{array}\right.$$

The aim of such an input is to synchronize the network, in order to study its resulting behavior, especially its desynchronization. Furthermore, we require two other major properties. Firstly, in the absence of stimulation, the activity of the network must remain low, which means that the network is naturally well desynchronized. Secondly, the network must have at least one pacemaker oscillator, which means that the vector (0,...,0) is not an attractor for the network characterized by $(x_1, y_1, \ldots, x_N, y_N)$. We will see that when the above conditions are fulfilled, the resulting behavior is that of a desynchronized network comprising pacemaker oscillators, regardless of the values of r and θ.

0.9.2 Different kinds of architectures

Many articles discuss neural networks with simple architectures such as the "ring network" (Pasemann, 1995) or the chain of oscillators (Wang, 1996), giving rise to simple mathematical studies. In contrast, we have chosen to impose the weakest possible conditions on connections. This approach does not allow us to carry out exact calculations and we must therefore often make mathematically relevant approximations.

Definitions

For the purposes of our study, we introduce two other entities in order to characterize the state of the network. We refer to the following entities as the mean activity of the excitatory and the mean activity of the inhibitory neurons, respectively:

$$\mathbf{E}(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t)$$

$$\mathbf{I}(t) = \frac{1}{N} \sum_{i=1}^{N} y_i(t)$$

A good test for network desynchronization is $\lim_{t \to \infty} (\mathbf{E}(t), \mathbf{I}(t)) = (0, 0)$.

Fully connected neural network

Using a fully connected network, we have investigated the conditions needed to fulfill the requirements laid out in section 6.1. For that purpose, we may make the following assumptions (H):

1. $c_{xx} = c_{yx} = c_e$ and $c_{yy} = c_{xy} = -c_i$ where c_e and c_i are both positive.
2. all oscillators are identical.
3. the network remains near the secondary Hopf bifurcation.

Assuming that (H) holds true, the requirement of a well desynchronized network implies the following:

Property 7 *Let $f_e = c_e(N-1)$ and $f_i = c_i(N-1)$ be the excitatory and the inhibitory strengths of the network, respectively.*

Using $\Delta(q) = (f_e - f_i)^2 - 4\beta(\beta + f_e + f_i)$, we need:

$$\begin{cases} \text{if } \Delta(q) > 0 & : \quad \max(|\alpha + \frac{f_e - f_i}{2} + \frac{1}{2}\sqrt{\Delta(q)}|, |\alpha + \frac{f_e - f_i}{2} - \frac{1}{2}\sqrt{\Delta(q)}|) > 1 \\ \text{if } \Delta(q) < 0 & : \quad \alpha^2 + \beta^2 + f_e(\alpha + \beta) + f_i(\beta - \alpha) < 1 \end{cases}$$

The proof of this property is given in Appendix A.
The requirement for the existence of at least one pacemaker oscillator reads as follows:

Property 8 *If the network satisfies property 7, using $\Delta(p) = 4\beta(c_i + c_e - \beta) + (c_e - c_i)^2$, we need:*

$$\begin{cases} \text{if } \Delta(p) > 0 & : \quad \max(|\alpha - \frac{c_e - c_i}{2} + \frac{1}{2}\sqrt{\Delta(p)}|, |\alpha - \frac{c_e - c_i}{2} - \frac{1}{2}\sqrt{\Delta(p)}|) > 1 \\ \text{if } \Delta(p) < 0 & : \quad \alpha^2 + \beta^2 - \beta(c_e + c_i) + \alpha(c_i - c_e) > 1 \end{cases}$$

The proof of this property is given in Appendix B.
Interpretation of these properties
First, we may conclude from the demonstration of property 7 that when (H) holds true, the network is equivalent to the following interaction diagram:

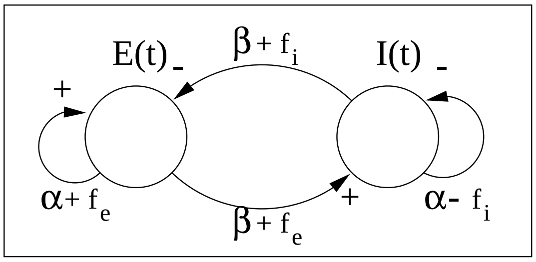

Figure 10 : Figure shows the equivalence between a microscopic description of a network of N fully connected modules and a macroscopic description with a single module comprising one excitatory neuron $E(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t)$ and one inhibitory neuron $I(t) = \frac{1}{N}\sum_{i=1}^{N} y_i(t)$, where $x_i(t)$ and $y_i(t)$ are the respective activities of the excitatory neuron and of the inhibitory neuron of the ith module. In the microscopic description, the intramodule connections are equal to β and α, the connection between two neurons of different modules is c_e when the connection is excitatory and c_i when the connection is inhibitory. In the macroscopic description, $f_i = c_i(N-1)$ and $f_e = c_e(N-1)$. The signs indicate whether the connections in the direction of the arrows are excitatory or inhibitory.

If we wish to take a plausible approach, we have $\alpha - f_i \leq 0$, leading to a negative autocatalytic interaction for $\mathbf{I(t)}$. Furthermore, if we assume $f_e \approx f_i$, property 7 implies that the condition to be fulfilled is:

$$\alpha^2 + \beta^2 + 2\beta f_e < 1$$

The size of the network is thus of major importance: drastic conditions must be imposed on the connections in order to obtain a naturally well desynchronized network. In addition, the criteria put forward in property 7 may be used to monitor the desynchronization speed. For instance, if $4\beta(\beta+f_e+f_i) > (f_e-f_i)^2$, the desynchronization is monitored by $\alpha^2+\beta^2+f_e(\alpha+\beta)+f_i(\alpha-\beta) < 1$. The following graph is an illustration of this:

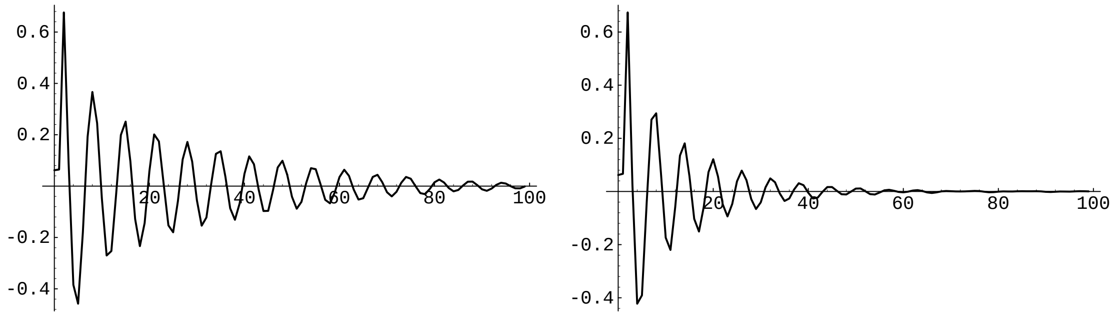

Figure 11 : Figure (A) represents E(t) $= \frac{1}{N}\sum_{i=1}^{N} x_i(t)$ *with one hundred modules ,* $c_e = 0.002$, $c_i = 0.084$, $\alpha = 0.9$ *and* $\beta = 0.5$. *Figure (B) represents of E(t)* $= \frac{1}{N}\sum_{i=1}^{N} x_i(t)$ *with one hundred modules ,* $c_e = 0.002$, $c_i = 0.01$, $\alpha = 0.9$ *and* $\beta = 0.5$.

If we now look back to property 8, we may see that that when one oscillator is a pacemaker, the others also have to be pacemakers. Numerically, this is still valid, even far from the secondary Hopf bifurcation. In addition, the necessary criterion to obtain at least one pacemaker neuron is independent of the size of the network.
Properties 8 and 9 are difficult to fulfill simultaneously. For instance, when $c_e \approx c_i$, the network must verify:

$$\begin{cases} \alpha^2+\beta^2-2c_e\beta & > & 1 \\ \alpha^2+\beta^2+2\beta f_e & < & 1 \end{cases}$$

These inequations obviously cannot be satisfied simultaneously. In order to enable the network to verify properties 8 and 9, we must reduce the number of connections.

Partially connected neural network

Here, each oscillator, i, has been connected only to its neighbourhood V(i), comprising its eight nearest neighbours. We should first highlight the changes that are induced by these new connections, with respect to desynchronization. After this, we will investigate the possible changes that could be generated by a different discrete-time dynamic. With the connections diagram described above, and with the same kind of dynamic as that described in section 6.1, property 7 is changed into:

Property 9 *Assume that the hypothesis (H) of section 6.2.2 holds true. The criterion stated in property 7 is still valid, for large networks, if we replace f_e by $f_{ev} = 8c_e$ and f_i by $f_{iv} = 8c_i$.*

Proof: Because (H) holds true, we are near the secondary Hopf bifurcation. The evolution of $(\mathbf{E}(t), \mathbf{I}(t))$ is governed by:

$$\mathbf{E}(t+1) = \alpha\mathbf{E}(t) - \beta\mathbf{I}(t) + \frac{1}{N} + \frac{c_e}{N}\underbrace{\sum_{i=1}^{N}\sum_{j\in V(i)} x_j(t)}_{S} - \frac{c_i}{N}\sum_{i=1}^{N}\sum_{j\in V(i)} y_j(t)$$

$$\mathbf{I}(t+1) = \beta\mathbf{E}(t) + \alpha\mathbf{I}(t) + \frac{1}{N} + \frac{c_e}{N}\sum_{i=1}^{N}\sum_{j\in V(i)} x_j(t) - \frac{c_i}{N}\sum_{i=1}^{N}\sum_{j\in V(i)} y_j(t)$$

A calculation distinguishing the different kinds of oscillators (oscillators of the center of the network, or those of the edge) yields:

$$\frac{1}{N}S = 8\mathbf{E}(t) + O(\frac{1}{\sqrt{N}})$$

We can now deduce that the system satysfied by $(\mathbf{E}(t), \mathbf{I}(t))$ is as follows:

$$\left\{\begin{array}{lcl} \mathbf{E}(t+1) & = & (\alpha + 8c_e)\mathbf{E}(t) - (\beta + 8c_i)\mathbf{I}(t) + O(\frac{1}{\sqrt{N}}) \\ \mathbf{I}(t+1) & = & (\beta + 8c_e)\mathbf{E}(t) + (\alpha - 8c_i)\mathbf{I}(t) + O(\frac{1}{\sqrt{N}}) \end{array}\right.$$

For sufficiently large values of N, the system above is similar to that we obtain in the proof of property 7 with f_e replaced by $f_{ev} = 8c_e$ and f_i by $f_{iv} = 8c_i$. The conclusions are similar to those for property 7.■

What is interesting in this property is that the desynchronization depends only on the neighbourhood, V(i), but not on the size, N, of the network, contrary to what is found with fully connected networks. Though it is easier to desynchronize a partially connected network than a fully connected one, maintaining a pacemaker activity while desynchronizing remains difficult. We will now make use of the observation made in section 3.1 in order to derive a new kind of evolution for network units. Indeed, oscillator evolution after synchronization gives:

$$\left\{\begin{array}{lcl} x_i(t+1) & = & \tanh_\epsilon(\alpha x_i(t) - \beta y_i(t) + \sum_{j\in V(i)} (c_e x_i(t) - c_i y_j(t))) \\ y_i(t+1) & = & \tanh_\epsilon(\beta x_i(t) + \alpha y_i(t) + \sum_{j\in V(i)} (c_e x_i(t) - c_i y_j(t))) \end{array}\right.$$

where $\tanh_\epsilon(u)$ is defined by: $\exists\eta/\tanh_\epsilon(u) = \frac{1-e^{2(1+\eta)u}}{1+e^{2(1+\eta)u}}$.

We may assume that hypothesis (H) of section 6.2.2 still holds true. With this new kind of evolution, we are only able to produce simulations. A first observation is that it is really easy to obtain a pacemaker oscillator by using $\alpha^2 + \beta^2 > 1$. After synchronization, desynchronization may be obtained for a range of parameters greater than that of property 9. Figure 12 shows a configuration in which the network does not desynchronize, using the evolution model described in section 6.1, but does with the above one.

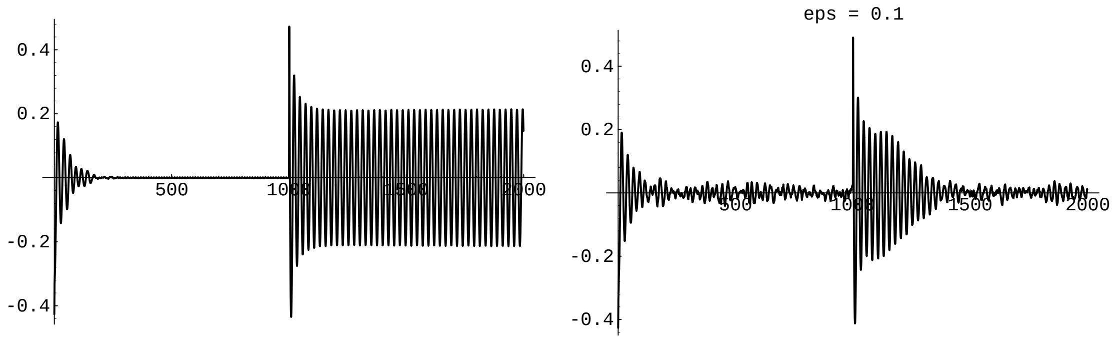

Figure 12: In figure (A) we consider the representation of E(t) $= \frac{1}{N}\sum_{i=1}^{N} x_i(t)$ *for values of paramaters which lead to good desynchronization of the system. We apply an input at* $t = 1000$ *to each neuron of every module, the input being identical for each neuron. The modules remain synchronized even after some time. On the other hand, in figure(B) we consider a compact set valued iteration defined by a random value* η *in* $[-\epsilon, \epsilon]$, $\epsilon = 0.1$, *and :*

$$\begin{cases} x(t+1) & = & \tanh((1+\eta)(\alpha x(t) - \beta y(t))) \\ y(t+1) & = & \tanh((1+\eta)(\beta x(t) + \alpha x(t))) \end{cases}$$

Other parameters are identical to those of figure (A). The input is of the same kind as in figure (A). We observe a desynchronization.

0.10 Conclusion

We have presented both a discrete-time version and a continuous-time version of the same neural oscillator. Throughout our study, we have attempted to show that the continuous version undergoes a bifurcation for some values of the parameters, while, the discrete-time model has two bifurcation points (it is well known that a discrete-time dynamic is more complex than its continuous counterpart). When parameters are chosen appropriately, both models exhibit oscillatory behaviors. With such values of the parameters, we have put forward a relationship between the models at the birth of oscillations. We have then investigated monitoring of the phase lag between oscillators, through mutual connections. For the continuous time model, we obtained some analytical results, which are in accordance with those obtained by Hoppensteadt

and Izhikevitch (1997), when the coupling consists of one connection. These results are extended to several connections by the use of numerical simulations.
In contrast, the discrete-time model does not allow such an analysis. In any case, it was observed that it becomes difficult to assess the phase lag for a network of more than two oscillators, with respect to the continuous model. For this reason, we have decided to achieve synchronization in larger networks by means of inputs.
Then, we have investigated the desynchronization of synchronized networks, but only for a discrete-time dynamic. This has seldom been done before. We have therefore carried out a study near the secondary Hopf bifurcation point for the discrete-time model. We have managed to obtain analytical results which emphasize that fully connected networks are not relevant to any modelling of desynchronization, since they impose overly drastic conditions on the connections. Conversely, a sparser connected network allows us to maintain an oscillatory activity after stimulation, as well as a weak amplitude for the mean activity. Secondly, we also notice that macroscopic description is possible for such a network. This may be an advantage from a biological point of view, in that recordings of electrical activity cannot be achieved for a single neuron. Only the activity of a network is accessible.
>From this point of view it may be seen that this model, which allows us to switch from a microscopic to a macroscopic description, may prove to be of value. Finally, we made an attempt, with a new kind of evolution model, which has given promising simulation results, with respect to desynchronization, but so far lacks any analytical results. Further investigations should be carried out in this direction, in order to work towards biological plausibility.

APPENDIX A

Property 7 Let us call $f_e = c_e(N-1)$and $f_i = c_i(N-1)$ these are the excitatory and the inhibitory strengths of the network, respectively.
Using $\Delta(q) = (f_e - f_i)^2 - 4\beta(\beta + f_e + f_i)$, we need :

$$\begin{cases} \text{if } \Delta(q) > 0 & : \max(|\alpha + \frac{f_e - f_i}{2} + \frac{1}{2}\sqrt{\Delta(q)}|, |\alpha + \frac{f_e - f_i}{2} - \frac{1}{2}\sqrt{\Delta(q)}|) > 1 \\ \text{if } \Delta(q) < 0 & : \alpha^2 + \beta^2 + f_e(\alpha+\beta) + f_i(\beta - \alpha) < 1 \end{cases}$$

Proof : Near the secondary Hopf bifurcation, the dynamic is governed by :

$$\begin{cases} x_i(t+1) &= \alpha_i x_i(t) - \beta_i y_i(t) + \sum_{j=1, j\neq i}^{N} (c_e x_j(t) - c_i y_j(t)) \\ y_i(t+1) &= \beta_i x_i(t) + \alpha_i y_i(t) + \sum_{j=1, j\neq i}^{N} (c_e x_j(t) - c_i y_j(t)) \end{cases}$$

Since the oscillators are assumed to be identical, we write $\alpha_i = \alpha$ and $\beta_i = \beta$ which implies that the activities of the excitatory and of the inhibitory neurons evolve as follows :

$$\begin{cases} \mathbf{E}(t+1) &= (\alpha + c_e(N-1))\mathbf{E}(t) - (\beta + c_i(N-1))\mathbf{I}(t) \\ \mathbf{I}(t+1) &= (\beta + c_e(N-1))\mathbf{E}(t) + (\alpha - c_i(N-1))\mathbf{I}(t) \end{cases}$$

Using $f_e = c_e(N-1)$ and $f_i = c_i(N-1)$ we rewrite the system in the form :

$$\begin{cases} \mathbf{E}(t+1) &= (\alpha + f_e)\mathbf{E}(t) - (\beta + f_i)\mathbf{I}(t) \\ \mathbf{I}(t+1) &= (\beta + f_e)\mathbf{E}(t) + (\alpha - f_i)\mathbf{I}(t) \end{cases}$$

The characteristic polynom of the evolution matrix is :

$$q(x) = x^2 - (2\alpha - f_i + f_e)x + \alpha^2 + \beta^2 + \alpha(f_e - f_i) + \beta(f_i + f_e)$$

Using its discriminant :

$$\Delta(q) = (f_e - f_i)^2 - 4\beta(\beta + f_e + f_i)$$

we are led to the following conclusions :

1. if $\Delta(q) < 0$
   We have two conjuguate eigenvalues, $vp1$ and $vp2$. The condition required for (0,0) to be an attractor is thus :

   $$|vp1| = |vp2| = \alpha^2 + \beta^2 + f_e(\alpha + \beta) + f_i(\beta - \alpha) < 1$$

2. if $\Delta(q) > 0$
   the condition required for (0,0) to be an attractor is

   $$max(|vp_1|, |vp_2|) < 1$$

   where :

   $$vp_1 = \alpha + \frac{f_e - f_i}{2} + \sqrt{(\frac{f_e - f_i}{2})^2 - \beta(\beta + f_e + f_i)}$$

   $$vp_2 = \alpha + \frac{f_e - f_i}{2} - \sqrt{(\frac{f_e - f_i}{2})^2 - \beta(\beta + f_e + f_i)} \quad \blacksquare$$

APPENDIX B

Property 8 If the network satisfies property 7, using $\Delta(p) = 4\beta(c_i + c_e - \beta) + (c_e - c_i)^2$ we need :

$$\begin{cases} \text{if } \Delta(p) > 0 & : \max(|\alpha - \frac{c_e - c_i}{2} + \frac{1}{2}\sqrt{\Delta(p)}|, |\alpha - \frac{c_e - c_i}{2} - \frac{1}{2}\sqrt{\Delta(p)}|) > 1 \\ \text{if } \Delta(p) < 0 & : \alpha^2 + \beta^2 - \beta(c_e + c_i) + \alpha(c_i - c_e) > 1 \end{cases}$$

Proof : For the sake of simplicity, we denote $(x_1(t), y_1(t), \ldots, x_N(t), y_N(t))$ by $z(t)$ where N is the number of oscillator. Since (H) holds true, the dynamic of $z(t)$ is governed by :

$$z(t+1) = DF(0)z(t)$$

where :

$$DF(0) = \begin{pmatrix} \alpha & -\beta & c_e & -c_i & c_e & -c_i \\ \beta & \alpha & c_e & \vdots & \vdots & \vdots \\ c_e & -c_i & \ddots & -c_i & \vdots & \vdots \\ \vdots & \vdots & \ldots & \ddots & c_e & -c_i \\ \vdots & \vdots & \ldots & \ldots & \alpha & -\beta \\ c_e & -c_i & \ldots & \ldots & \beta & \alpha \end{pmatrix}$$

The condition we are looking for is $\rho(DF(0)) > 1$, where $\rho(M) = \max_{\alpha \in S_p(M)} |\alpha|$. If we call N the number of oscillators, let P_N be the following polynom :

$$P_N(x) = det(DF(0) - xI_{2N})$$

Our aim is to determine the roots of this polynom which are the eigenvalues of $DF(0)$.
Simple calculations lead to the recurrence formula :
if $N \geqslant 2$:

$$P_N(x) = p(x)[P_{N-1}(x) + p^{N-2}(x)((c_i - c_e)x + (c_e - c_i)\alpha + (c_e + c_i)\beta)]$$

where

$$p(x) = x^2 + (c_e - c_i - 2\alpha)x + \alpha^2 + \beta^2 - \alpha(c_e - c_i) + \beta(-c_i - c_e)$$

and if $N = 1$:

$$P_1(x) = x^2 - 2\alpha x + \alpha^2 + \beta^2$$

This finally implies that :

$$P_N(x) = p^{N-1}(x)q(x)$$

where $q(x)$ is the polynom introduced in the demonstration of property 7.

$$\begin{aligned} \Delta(p) &= 4\beta(c_i + c_e - \beta) + (c_e - c_i)^2 \\ \Delta(q) &= -4\beta(\beta + (N-1)(c_e + c_i)) + (N-1)^2(c_e - c_i)^2 \end{aligned}$$

A) conditions for p
The following study leads to conditions that are independent of the size of the system

1. $4\beta(c_i + c_e - \beta) + (c_e - c_i)^2 < 0$
   In this case, we have two conjugate eigenvalues of order N-1. To get at least one eigenvalue with its module greater than one we need to have :
   $$\alpha^2 + \beta^2 - \beta(c_e + c_i) + \alpha(c_i - c_e) > 1$$
   When $c_i = c_e$ we must have $\alpha^2 + \beta^2 - 2c_e\beta > 1$
2. $4\beta(c_i + c_e - \beta) + (c_e - c_i)^2 > 0$
   We have two real eigenvalues, v_1, v_2 which are :
   $$\alpha - \frac{c_e - c_i}{2} \pm \sqrt{\beta(c_i + c_e - \beta) + (\frac{c_e - c_i}{2})^2}$$
   The condition we were looking for is as follows :
   $$max(|v_1|, |v_2|) > 1$$
   If $c_i = c_e$ the eigenvalues are equal to $\alpha \pm \sqrt{\beta(2c_e - \beta)}$
   and the condition becomes :
   $$\alpha > 1 - \sqrt{(\beta(2c_e - \beta))}$$

B) conditions for q The same study can be carried changing c_e into $f_e = c_e(N-1)$ and c_i into $f_i = c_i(N-1)$. The calculation leads to the conditions required for the activity of the excitatory and the inhibitory neurons not to tending towards (0,0). Therefore, the roots of q cannot be bigger than 1, since we have assumed that the network verifies property 7.■

A

Première annexe

A.1 Démonstration du théorème 3 chapitre 3

Nous développons ici une démonstration disponible dans [85], sous une autre forme. On considère les espaces $V_1 = \left\{h[n-2l]\right\}_{l\in\mathbb{Z}}$ et $W_1 = \left\{g[n-2l]\right\}_{l\in\mathbb{Z}}$ qui vérifient :

$$V_0 = V_1 \oplus^{\perp} W_1.$$

Soit les familles :

$$\begin{aligned} V_j &= \left\{h_j[n-2^j l]\right\}_{l\in\mathbb{Z}} \\ V_{j+1} &= \left\{h_{j+1}[n-2^{j+1} l]\right\}_{l\in\mathbb{Z}} \\ W_{j+1} &= \left\{g_{j+1}[n-2^{j+1} l]\right\}_{l\in\mathbb{Z}} \end{aligned}$$

On va montrer que les familles génératrices de ces espaces sont orthogonales et que :

$$V_j = V_{j+1} \oplus^{\perp} W_{j+1}$$

Montrons $V_{j+1} \subset V_j$ et $W_{j+1} \subset V_j$:

$$\begin{aligned} h_{j+1}[l-2^{j+1}n] &= \textstyle\sum_k h[k]h_j[l-2^{j+1}n-2^j k] \\ &= \textstyle\sum_k h[k]h_j[l-2^j(2n+k)] \\ &= \textstyle\sum_k h_j[l-2^j k]h[k-2n] \end{aligned}$$

De la même façon, on a :

$$g_{j+1}[l-2^{j+1}n] = \sum_k h_j[l-2^j k]g[k-2n]$$

Montrons que $\left\{h_{j+1}[l-2^{j+1}n], g_{j+1}[l-2^{j+1}n]\right\}_{n\in\mathbb{Z}}$ forme une base orthonormée de V_j.
On procède par récurrence :
supposons que, jusqu'au rang j, l'on ait:

$$V_{j-1} = V_j \oplus^{\perp} W_j$$

et que $\left\{h_j[n-2^j l]\right\}_{l\in\mathbb{Z}}$ et $\left\{g_j[n-2^j l]\right\}_{l\in\mathbb{Z}}$ soient respectivement des bases orthonormées alors :

$$< h_{j+1}[l], h_{j+1}[l-2^{j+1}n] >=< \sum_k h[k]h_j[l-2^j k], \sum_k h[k-2n]h_j[l-2^j k] >$$

D'après l'orthonormalité de $\left\{h_j[l-2^j k]\right\}_{l\in\mathbb{Z}}$, il vient :

$$< h_{j+1}[l], h_{j+1}[l-2^{j+1}n] >= \sum_k h[k]h[k-2n] = \delta[n]$$

De la même façon, en utilisant $\sum_k g[k]g[k-2n] = \delta[n]$:

$$< g_{j+1}[l], g_{j+1}[l-2^{j+1}n] >= \delta[n]$$

Enfin :

$$< g_{j+1}[l], h_{j+1}[l-2^{j+1}n] >= \sum_k g[k]h[k-2n] = 0$$

V_{j+1} et W_{j+1} sont donc supplémentaires orthogonaux et les familles qui les génèrent sont orthonormées. Pour montrer que :

$$V_j = V_{j+1} \oplus^{\perp} W_{j+1},$$

il suffit alors de montrer que la norme l_2 des coefficients est conservée par passage de l'espace V_j aux espaces V_{j+1} et W_{j+1}.
Pour tout élément f de V_j, on peut écrire :

$$\begin{array}{ccc} s_{j+1}[n] & = & < f[l], h_{j+1}[l-2^{j+1}n] > \\ = \sum_k h[k-2n] < f[l], h_j[l-2^j k] > & = & \sum_k h[k-2n]s_j[k] \end{array}$$

et

$$d_{j+1}[n] = \sum_k g[k-2n]s_j[k]$$

Du fait que les filtres sont parfaits et que $\tilde{h} = h$ et $\tilde{g} = g$, on obtient :

$$s_j(n) = \sum_l s_{j+1}(l)h[n-2l] + \sum_l d_{j+1}[l]g[n-2l]$$

De l'orthogonalité de la famille $\left\{h[n-2l], g[n-2l]\right\}_{l\in\mathbb{Z}}$, on déduit:

$$\sum_n s_j[n]^2 = \sum_n s_{j+1}[n]^2 + \sum_n d_{j+1}[n]^2,$$

puis que :

$$\sum_{j=1}^{J}\sum_n d_j[n]^2 + \sum_n s_J[n]^2 = \sum_n s_0[n]^2 = \sum_n x[n]^2,$$

car $s_0 = x$. Donc, on a bien la résultat souhaité : la famille $\left\{h_J[n-2^J l], g_j[n-2^j l]_{1\leq j\leq J}\right\}_{l\in\mathbb{Z}}$ est orthonormée.

A.2 Démonstration du théorème 4 chapitre 3

La démonstration proposée ici est originale. Pour plus de commodité, on notera

$$\Phi_J[k] = (\tfrac{1}{a})^{\frac{J}{2}} h_J[k]$$
$$\Psi_j[k] = (\tfrac{1}{a})^{\frac{j}{2}} g_j[k]$$

Supposons $|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = a$. Il faut donc montrer que

$$\sum_{l\in\mathbb{Z}} < x[k], \Phi_J[k-l] >^2 + \sum_{j=1}^{J} \sum_{l\in Z} < x[k], \Psi_j[k-l] >^2 = < x[k], x[k] >$$

Montrons tout d'abord que les fonctions Φ_J et Ψ_j vérifient :

$$|\hat{\Phi}_J(\omega)|^2 + \sum_{j=1}^{J} |\hat{\Psi}_j(\omega)|^2 = 1$$

Par récurrence, nous avons :
pour $J = 1$

$$|\hat{\Phi}(\omega)|^2 + |\hat{\Psi}(\omega)|^2 = \frac{1}{a}(|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2) = 1$$

Supposons que, jusqu'au rang $J-1$, on ait :

$$|\hat{\Phi}_{J-1}(\omega)|^2 + \sum_{j=1}^{J-1} |\hat{\Psi}_j(\omega)|^2 = 1$$

Alors au rang J, on obtient:

$$|\hat{\Phi}_J(\omega)|^2 + \sum_{j=1}^{J} |\hat{\Psi}_j(\omega)|^2$$

$$= |\hat{\Phi}_J(\omega)|^2 + |\hat{\Psi}_J(\omega)|^2 + \sum_{j=1}^{J-1} |\hat{\Psi}_j(\omega)|^2$$

D'après les relations définissant h_i et g_i, il vient :

$$\hat{h}_{i+1}(\omega) = \hat{h}(2^i\omega)\hat{h}_i(\omega)$$
$$\hat{g}_{i+1}(\omega) = \hat{g}(2^i\omega)\hat{h}_i(\omega)$$

De ceci, on tire :

$$|\hat{\Phi}_J(\omega)|^2 + |\hat{\Psi}_J(\omega)|^2 = |\hat{\Phi}_{J-1}(\omega)|^2$$

d'où l'égalité cherchée.
On réécrit alors les coefficients de la somme :

$$< x[k], \Phi_J[k-l] >= x * \bar{\Phi}_J[l]$$

et

$$< x[k], \Psi_j[k-l] >= x * \bar{\Psi}_j[l]$$

Considérons alors les transformées de Fourier:

$$\widehat{x * \bar{\Phi}_J}(\omega) = \hat{x}(\omega)\hat{\Phi}_J^*(\omega)$$

De cela, on déduit :

$$\frac{1}{2\pi}\int_0^{2\pi} |\widehat{x * \bar{\Phi}_J}(\omega)|^2 d\omega = \sum_l < x[k], \Phi_J[k-l] >^2$$

et, par ailleurs :

$$\frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 |\hat{\Phi}_J^*(\omega)|^2 d\omega = \frac{1}{2\pi}\int_0^{2\pi} |\widehat{x * \bar{\Phi}_J}(\omega)|^2 d\omega = \sum_l < x[k], \Phi_J[k-l] >^2$$

De la même manière, pour le filtre Ψ_j, on a :

$$\frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 |\hat{\Psi}_j^*(\omega)|^2 d\omega = \frac{1}{2\pi}\int_0^{2\pi} |\widehat{x * \bar{\Psi}_j}(\omega)|^2 d\omega = \sum_l < x[k], \Psi_j[k-l] >^2$$

D'où, l'on tire :

$$\frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 (|\hat{\Phi}_J^*(\omega)|^2 + \sum_{j=1}^{j=J} |\hat{\Psi}_j^*(\omega)|^2) d\omega = \frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 d\omega =< x[k], x[k] >$$

On a donc un frame orthogonal.
Réciproquement, supposons le frame orthogonal, on a alors :

$$\sum_{l\in\mathbb{Z}} < x[k], \Phi_J[k-l] >^2 + \sum_{j=1}^{J}\sum_{l\in Z} < x[k], \Psi_j[k-l] >^2 =< x[k], x[k] > \quad \forall x \in l_2$$

Cela se réécrit sous la forme :

$$\frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 (|\hat{\Phi}_J^*(\omega)|^2 + \sum_{j=1}^{j=J} |\hat{\Psi}_j^*(\omega)|^2) d\omega = \frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 d\omega =< x[k], x[k] > \quad \forall x \in l_2$$

D'après la conservation du produit scalaire par transformation de Fourier et, du fait que la transformée de Fourier est une isométrie de l_2 dans $L^2([0, 2\pi])$, il vient :

$$\frac{1}{2\pi}\int_0^{2\pi} |\hat{x}(\omega)|^2 (|\hat{\Phi}_J^*(\omega)|^2 + \sum_{j=1}^{j=J} |\hat{\Psi}_j^*(\omega)|^2 - 1) d\omega = 0 \quad \forall \hat{x} \in L^2([0, 2\pi])$$

Il s'agit alors de montrer que :

$$\forall f \in L^2([0, 2\pi]) \int_0^{2\pi} f^2(t) g(t) = 0 \rightarrow g = 0$$

Supposons g continue, soit I_+ (respectivement I_-) les intervalles sur lesquels g est positif (respectivement négatif). En prenant comme fonction f la fonction constante égale à 1 sur $[0, 2\pi]$, on obtient :

$$\int_{I_+} g(t) = -\int_{I_-} g(t)$$

En prenant ensuite f valant 1 sur I_+ et $\sqrt{2}$ sur I_-, il vient :

$$\int_{I_+} g(t) = -2 \int_{I_-} g(t)$$

Donc, on en déduit que g est nulle sur I_- et par conséquent sur I_+. Revenons alors à notre problème, pour conclure que :

$$|\hat{\Phi}^*_J(\omega)|^2 + \sum_{j=1}^{j=J} |\hat{\Psi}^*_j(\omega)|^2 = 1$$

D'où l'on tire que :

$$|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = a$$

A.3 Démonstration du théorème 6 chapitre 3

La démonstration proposée ici est originale. Utilisons les mêmes notations que dans la section précédente et supposons que $|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = a$.
Montrons que :

$$|\hat{\Phi}_J(\omega_1)\hat{\Phi}_J(\omega_2)|^2 + \sum_{j=1}^{J} |\hat{\Phi}_j(\omega_1)\hat{\Psi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Phi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Psi}_j\omega_2)|^2 = 1$$

Procédons par récurrence :
Pour $J = 1$, on a :

$$|\hat{\Phi}(\omega_1)\hat{\Phi}(\omega_2)|^2 + |\hat{\Phi}(\omega_1)\hat{\Psi}(\omega_2)|^2 + |\hat{\Psi}(\omega_1)\hat{\Phi}(\omega_2)|^2 + |\hat{\Psi}(\omega_1)\hat{\Psi}(\omega_2)|^2$$

$$= |\hat{\Phi}(\omega_1)|^2(|\hat{\Phi}(\omega_2)|^2 + |\hat{\Psi}(\omega_2)|^2) + |\hat{\Psi}(\omega_1)|^2(|\hat{\Phi}(\omega_2)|^2 + |\hat{\Psi}(\omega_2)|^2) = 1$$

Supposons que, jusqu'au rang $J-1$, on ait :

$$|\hat{\Phi}_{J-1}(\omega_1)\hat{\Phi}_{J-1}(\omega_2)|^2 + \sum_{j=1}^{J-1} |\hat{\Phi}_j(\omega_1)\hat{\Psi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Phi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Psi}_j(\omega_2)|^2 = 1$$

Alors, au rang J :

$$|\hat{\Phi}_J(\omega_1)\hat{\Phi}_J(\omega_2)|^2 + \sum_{j=1}^{J} |\hat{\Phi}_j(\omega_1)\hat{\Psi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Phi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Psi}_j(\omega_2)|^2$$

$$= |\hat{\Phi}_J(\omega_1)\hat{\Phi}_J(\omega_2)|^2 + |\hat{\Phi}_J(\omega_1)\hat{\Psi}_J(\omega_2)|^2 + |\hat{\Psi}_J(\omega_1)\hat{\Phi}_J(\omega_2)|^2 + |\hat{\Psi}_J(\omega_1)\hat{\Psi}_J(\omega_2)|^2 +$$
$$\sum_{j=1}^{J-1} |\hat{\Phi}_j(\omega_1)\hat{\Psi}_j(\omega_2)|^2 + |\hat{\Psi}_j(\omega_1)\hat{\Phi}_j(\omega_2)|^2 + |\hat{\Psi}_J(\omega_1)\hat{\Psi}_J(\omega_2)|^2$$

Or :

$$|\hat{\Phi}_J(\omega_1)\hat{\Phi}_J(\omega_2)|^2 + |\hat{\Phi}_J(\omega_1)\hat{\Psi}_J(\omega_2)|^2 + |\hat{\Psi}_J(\omega_1)\hat{\Phi}_J(\omega_2)|^2 + |\hat{\Psi}_J(\omega_1)\hat{\Psi}_J(\omega_2)|^2 =$$
$$|\hat{\Phi}_J(\omega_1)|^2(|\hat{\Phi}_J(\omega_2)|^2 + \hat{\Psi}_J(\omega_2)|^2) + |\hat{\Psi}_J(\omega_1)|^2(|\hat{\Phi}_J(\omega_2)|^2 + \hat{\Psi}_J(\omega_2)|^2)$$

$$= |\hat{\Phi}_{J-1}(\omega_1)|^2 |\hat{\Phi}_{J-1}(\omega_2)|^2,$$

d'où le résultat. Par ailleurs, on a :

$$\sum_l \sum_m \sum_{k,p} x[k,p] \Phi_J[k-l] \Phi_J[p-m] e^{-i(l\omega_1 + m\omega 2)}$$

$$= \hat{x}(\omega_1, \omega_2) \hat{\Phi}_J^*(\omega_1) \hat{\Phi}_J^*(\omega_2)$$

Puis, en intégrant sur $[0, 2\pi] \times [0, 2\pi]$:

$$\frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} |\hat{x}(\omega_1, \omega_2)|^2 |\hat{\Phi}_J^*(\omega_1)|^2 |\hat{\Phi}_J^*(\omega_2)|^2 = \sum_l \sum_m < x[k,p], \Phi_J[k-l] \Phi_J[p-m] >^2$$

On peut faire le même type de calcul avec les autres éléments de la famille, pour obtenir :

$$\sum_k \sum_p x[k,p]^2 = \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} |\hat{x}(\omega_1, \omega_2)|^2 d\omega_1 d\omega_2$$

$$= \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} |\hat{x}(\omega_1, \omega_2)|^2 (|\hat{\Phi}_J^*(\omega_1)|^2 |\hat{\Phi}_J^*(\omega_2)|^2 +$$

$$\sum_{j=1}^{J} (|\hat{\Phi}_j^*(\omega_1) \hat{\Psi}_j^*(\omega_2)|^2 + |\hat{\Psi}_j^*(\omega_1) \hat{\Phi}_j^*(\omega_2)|^2 + |\hat{\Psi}_j^*(\omega_1) \hat{\Psi}_j^*(\omega_2)|^2) d\omega_1 d\omega_2$$

$$= \sum_l \sum_m < x[k,p], \Phi_J[k-l] \Phi_J[p-m] >^2 +$$

$$\sum_{j=1}^{J} \sum_l \sum_m < x[k,p], \Phi_j[k-l] \Psi_j[p-m] >^2 + < x[k,p], \Psi_j[k-l] \Phi_j[p-m] >^2 +$$

$$< x[k,p], \Psi_j[k-l] \Psi_j[p-m] >^2$$

Donc, la famille est bien un frame orthogonal. La réciproque est identique à celle faite en une dimension.

A.4 Démonstration du théorème 9 chapitre 3

La démonstration proposée ici est originale. Supposons que les filtres h et g vérifient :

$$|\hat{h}(\omega_1) \hat{h}(\omega_2)|^2 + |\hat{h}(\omega_1) \hat{g}(\omega_2)|^2 + |\hat{h}(\omega_2) \hat{g}(\omega_1)|^2 = 1$$

On démontre alors facilement que :

$$\forall J \in N, \quad |\hat{h}_J(\omega_1) \hat{h}_J(\omega_2)|^2 + \sum_{j=0}^{J} |\hat{h}_j(\omega_1) \hat{g}_j(\omega_2)|^2 + |\hat{g}_j(\omega_1) \hat{h}_j(\omega_2)|^2 = 1$$

De ceci, on déduit, en utilisant le même type d'argument que dans A.3, que :

$$\sum_k \sum_p x[k,p]^2 = \sum_k \sum_l (< x[k,p], h_J[k-l] h_J[p-m] >^2 +$$

$$\sum_{j=1}^{J} < x[k,p], h_j[k-l]g_j[p-m] >^2 + < x[k,p], g_j[k-l]h_j[p-m] >^2)$$

Donc la famille est bien un frame orthogonal.
Un raisonnement identique à celui effectué dans A.2 permet de montrer la réciproque.

B

Deuxième annexe

B.1 Démonstration de la convergence des algorithmes utilisant des centres mobiles

Cette démostration est reprise de [88]. Considérons un ensemble fini n de points, répartis en k classes . On considère initialement k points $c_1, \cdots, c_k$ et les classes associées E_{c_i}. Alors, nous avons les inégalités suivantes :

$$\frac{1}{n}\sum_{i=1}^{k}(\sum_{j\in E_{c_i}} d(j,c_i)^2) \geq \frac{1}{n}\sum_{i=1}^{k}(\sum_{j\in E_{c_i}} d(j,g_i)^2) \geq \frac{1}{n}\sum_{i=1}^{k}(\sum_{j\in E_{g_i}} d(j,g_i)^2) \geq \frac{1}{n}\sum_{i=1}^{k}(\sum_{j\in E_{g_i}} d(j,p_i)^2)$$

où g_i est barycentre de E_{c_i} et p_i est le barycentre de E_{g_i}. D'après le théorème de König-Huygens, le centre de gravité g d'un ensemble E vérifie :

$$g = argmin(\sum_{j\in E} d(j,v)^2)$$

Donc, la première et la dernière inégalité sont prouvées. La seconde inégalité se prouve en remarquant qu'un point quelconque j est lié au point g_i dans le membre de gauche de la seconde inégalité qui est barycentre de E_{c_i}, mais ce n'est pas le point g_i le plus proche de j, alors que, par construction, tel est le cas dans le membre de droite de la seconde inégalité. La variance inter-classe définie par :

$$\sum_{j\in E_{c_i}} d(j,g_i)^2$$

avec g_i barycentre de E_{c_i}, diminue au cours des itérations, donc l'algorithme converge.

Bibliographie

[1] P. Abry and P. Flandrin : *On the initialization of the discrete wavelet transform algorithm.* IEEE signal processing letters, vol. 1, no. 2, 32-34, 1994.

[2] A. Atiya and P. Baldi : *Oscillations and synchronizations in neural networks : an exploration of the labelling hypothesis.* International Journal of Neural Systems, vol. 1, 103-124, 1989.

[3] R. Azencott, J.P. Wang and L. Younes : *Texture classification using windowed Fourier filters.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 2, 148-153, 1997.

[4] K. Berkner and R.O. Wells : *A new hierarchical scheme for approximating the continuous wavelet transform with applications to edge detection.* IEEE Signal Processing Letters, vol. 6, no. 8, 193-195, 1999.

[5] K. Berkner and R.O. Wells : *A fast approximation to the continuous wavelet transform with applications.* Proceeding of the Asilomar conference, Pacific Grove, CA, 1997.

[6] R.M Borisyuk R.M. and A.B. Kirilov : *Bifurcation analysis of a neural network model.* Biological Cybernetics, vol. 66, 319-325, 1992.

[7] C. Bouman and B. Liu : *Multiple resolution segmentation of textured images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, 99-113,1991.

[8] A.C. Bovik, M. Clark and W.S. Geisler : *Multichannel texture analysis using localized spatial filters.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no.1, 55-73, 1990.

[9] P. Brodatz : *Textures : A photographic album for artists and designers.* Dover, New York, 1966.

[10] J. Canny: *A computational approach to edge detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.8, 679-698, 1986.

[11] P. Carre, H. Leman, C. Fernandez and C. Marque : *Denoising of the uterine EHG by undecimated wavelet transform.* IEEE Transactions on Biomedical Engineering, vol. 45, no. 9, 1104-1113, 1998.

[12] F. Catté, J-M. Morel, P-L. Lions and T. Coll : *Image selective smoothing and edge detection by non linear diffusion.* SIAM J. Numer. Anal., vol. 29, 182-193, 1992.

[13] E. Cesmeli and D.L. Wang : *Texture segmentation using Gaussian-Markov random fields and oscillator networks.* IEEE Transactions on Neural Networks, vol. 12, no. 2, 394-404, 2001.

[14] F. Chapeau-Blondeau and G. Chauvet : *Stable, oscillatory and chaotic regimes in the dynamics of small neural oscillators with delay.* Neural Network, vol. 5, 735-743, 1992.

[15] R. Chellappa and S. Chatterjee : *Classification of textures using Gaussian-Markov random fields.* IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 33, 959-963, 1985.

[16] T. Chang and C.C.J Kuo : *Texture analysis and classification with tree-structured wavelet transform.* IEEE Transactions on Image Processing, vol. 2, no. 4, 429-441, 1993.

[17] Y.T. Chan and K.C. Ho : *Multiresolution analysis, its link to the discrete parameter wavelet transform and its initialization.* IEEE transactions on signal processing, vol. 44, no. 4, 1001-1006, 1996.

[18] B.B. Chaudhuri and N. Sarkar : *Texture segmentation using fractal dimension.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 1, 72-77, 1995.

[19] M. Clerc : *Analyse par ondelettes de processus localement dilatés, et application au gradient de texture.* Ph.D Thesis, Ecole Polytechnique, 1999.

[20] J.M. Coggins and A.K. Jain : *A Spatial filtering approach to texture analysis.* Pattern Recognition Letters, vol.3, no.3, 195-203, 1985.

[21] G.H. Cottet and M. El Ayyadi : *A Volterra type model for image processing.*, IEEE Transactions on Image Processing, vol. 7, no. 3, 292-303, 1998.

[22] G.B Coleman and H.C Andrews : *Image segmentation by clustering.* Proceedings IEEE, vol.67, no. 5, 773-785, 1979.

[23] C. Coroyer and P. Duvaut : *Apports des outils fondés sur les corrélations d'ordre* 3 *à l'analyse de textures.* Traitement du Signal, vol. 15, no. 2, 129-138, 1998.

[24] I. Daubechies: *Ten lectures on wavelets.* SIAM,1992.

[25] J.G Daugman : *Two-dimensional spectral analysis of cortical receptive profile.* Vision Research, vol. 20, 847-857, 1980.

[26] J. Demongeot, P. Kulesa and J.D. Murray : *Compact set valued flows II : applications in biological modelling.* C. R. Acad. Sci., vol.324, series II b, 107-115, 1997.

[27] J. Demongeot, J. Aracena,S. Ben Lamine, S. Meignen, A. Tonnelier : *Dynamical systems and biological regulations.* E.Goles and S.Martinez (eds.),Complex Systems, Kluwer Amsterdam, 105-149, 2001.

[28] D. Dunn and W. Higgins : *Optimal Gabor filters for texture segmentation.* IEEE Transactions on Image Processing, vol. 4,no .7, 947-964, 1995.

[29] D. Dunn, W.E. Higgins and J. Wakeley : *Texture segmentation using 2-D Gabor elementary functions.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 2, 130-149, 1994.

[30] J.S. Duncan and N. Ayache : *Medical image analysis : progress over two decades and the challenges ahead.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, 85-105.

[31] B. Ermentrout and N. Kopell : *Learning of phase lages in coupled oscillators.* Neural Computation, vol. 6, 225-241, 1994.

[32] N. Fatemi-Ghomi : *Performance measures for wavelet-based segmentation Algorithms.* Ph.D Thesis, Surrey University, 1997.

[33] J.M. Francos, A.Z. Meiri and B. Porat : *A unified texture model based on 2D wold-like decomposition.* IEEE Transactions on Signal Processing, vol. 41, no. 8, 2665-2677, 1993.

[34] C. Gasquet and P.Witomski : *Analyse de Fourier et applications.* Masson,1995.

[35] S. Geman and D. Geman : *Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.6, 721-741, 1984.

[36] D. Geman, S. Geman, C. Graffigne and P. Dong : *Boundary detection by constrained optimization* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, 609-628, 1990.

[37] M.A. Gluck : *Computational models of hippocampal function in memory.* Hippocampus, vol. 6. New-York : Wiley-Liss.

[38] C.M. Gray and W. Singer : *Stimulus-specific neural oscillations of cat visual cortex.* Proceedings of the national Academy of Sciences, USA, vol. 86, 1698-1702, 1989.

[39] A. Goltsev and D.C. Wunsch : *Inhibitory connections in the assembly neural network for texture segmentation* Neural Network, vol. 11, 951-962, 1998.

[40] H. Greenspan, R. Goodman, R. Chellapa and C.H. Anderson : *Learning discrimination rules in a multiresolution system.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 9, 894-901, 1994.

[41] J. Guckenheimer : *Isochrons and phaseless sets.* Journal of Mathematical Biology, vol. 2, 259-273, 1975.

[42] R.M. Haralick : *Statistical and structural approaches to texture.* Proceedings IEEE, vol.67, 786-804, 1979.

[43] T. Hofmann,J. Puzicha and J.M. Buhmann : *Unsupervised texture segmentation in a deterministic annealing framework.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, 803-818, 1998.

[44] F.C. Hoppensteadt and E.M. Izhikevich : *Weakly connected neural networks.* In J.E. Marsden , L. Sirovitch , F. John, Synaptic organization of the brain, 353-379. New-York : Springer-Verlag.

[45] R.L. Horner, J.A Innes, H.B Holden and A. Guz : *Afferent pathway for pharyngeal dilator reflex to negative airway pressure in man : a study using upper airway anesthesia.* Journal of physiology (London), vol. 436, 31-34, 1991.

[46] Hsi-Chin Hsin : *Texture segmentation using modulated wavelet transform* IEEE Transactions on Image Processing, vol. 9, no. 7, 1299-1302, 2000.

[47] S. Jaffard, Y. Meyer and S. D. Ryan : *Wavelets : Tools for science and technology.* SIAM, 2001.

[48] A. K. Jain and F. Farrokhnia : *Unsupervised texture segmentation using Gabor filters.* Pattern Recognition, vol.24, no.12, 1167-1190, 1991.

[49] A. K. Jain : *Algorithm for clustering data.* Prentice Hall, 1988.

[50] A. K Jain and D.Zongker: *Feature selection : evaluation, application, and small sample performance.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19, no.2, 153-158, 1997.

[51] A. K. Jain : *A sinusoidal family of unitary transform.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 4, 356-365, october 1979.

[52] A. K. Jain and K. Karu : *Learning texture discrimination masks.* IEEE Transactions on Pattern Analysis and Machine Intelligence,vol. 18, 195-205, 1996.

[53] B. Jawerth and W. Sweldens : *An overview of wavelet based multiresolution analyses.* SIAM Review, vol.36, no. 3, 377-412, 1994.

[54] B. Julesz : *Textons, the elements of texture perception and their interactions.* Nature, vol. 290, March 1981.

[55] L. M. Kaplan : *Extended fractal analysis for texture classification and segmentation.* IEEE Transactions on Image Processing, vol. 8, no. 11, 1572-1585, 1999.

[56] M. Khalil and J. Duchêne : *Uterine EMG analysis : a dynamic approach for change detection and classification.* IEEE Transactions on Biomedical Engineering, vol. 47, no. 6, 748-756, 2000.

[57] S. Krishnamachari and R. Chellappa : *Multiresolution Gauss-Markov random field models for texture segmentation.* IEEE Transactions on Image Processing, vol. 6, no. 2, 251-267, 1997.

[58] A. Laine and J. Fan : *Texture classification by wavelet packet signatures.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, 1186-1191, 1993.

[59] A. Laine and J. Fan : *An adaptive approach for texture segmentation by multi-channel wavelet frames.* Center for Computer Vision and Visualization, TR-93-025, 1993.

[60] A. Laine and J. Fan : *Frame representation for texture segmentation.* IEEE Transactions on Image Processing, vol. 5, 771-780, 1996.

[61] A.D. Lanterman, U. Grenander and M.I. Miller : *Bayesian segmentation via asymptotic partition functions.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 4, 337-347, 2000.

[62] Z. Li and J.J. Hopfield : *Modelling of olfactory bulb and its neural ring network.* Biological Cybernetics, vol. 61, 379-392, 1989.

[63] S. Liapis, N. Alvertos, and G. Tziritas : *Maximum likelihood classification and bayesian texture segmentation using discrete wavelet frames.* International Conference on Digital Signal Processing, Santorini, 1997.

[64] S. Mallat : *A theory for multiresolution signal decomposition : the wavelet representation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, 675-693, 1989.

[65] S. Mallat and S. Zhong : *Characterization of signals from multiscale edges.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 7, 710-732, 1992.

[66] S. Mallat : *A wavelet tour of signal processing.* Academic Press,1998.

[67] S. Mallat and W. L. Hwang : *Singularity detection and processing with wavelets.* IEEE Transactions on Information Theory, vol. 38, no. 2, 617-643, 1992.

[68] S.Mallat : *Wavelets for a vision.* Proceedings of the IEEE, vol. 84, no. 4, 604-614, 1996.

[69] J. Mao and A. K. Jain : *Texture classification and segmentation using multiresolution simultaneous autoregressive model.* Pattern Recognition, vol. 25, no.2, 173-188, 1992.

[70] Y. Meyer : *Les ondelettes : algorithmes et applications.* Paris, 1992.

[71] Y. Meyer : *Wavelets and Operators.* Advanced mathematics. Cambridge University Press, 1992.

[72] J-M. Morel and S. Solimini : *Variationnal method in image segmentation.* Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser, 1995.

[73] A. Papoulis : *Probability random variables and stochastic processes.* New York : McGraw-Hill, 1965.

[74] F. Pasemann : *Characterization of periodic attractors in neural ring network.* Neural Network, vol. 8, 421-429, 1995.

[75] T. Pham Dinh and J. Demongeot : *Simulation of biological oscillator : the respiratory system.* Journal of Theorical Biology, vol. 103, 113-132.

[76] S. Pittner and S. Kamarthi : *Feature extraction from wavelet coefficient for pattern recognition tasks.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vo. 21, no. 1, 83-88, 1999.

[77] L. Pietronero and R. Kupers : *Fractals in Physics.* L. Pietronero and E.Tosatti,eds, North-Holland, Amsterdam, 1986.

[78] R. Porter and N. Canagarajah : *A robust automatic clustering scheme for image segmentation using wavelets.* IEEE Transactions on Image Processing, vol. 5, no. 4, 662-665, 1996.

[79] M. Porat and Y. Zeevi : *Localized texture processing in vision : analysis and synthesis in Gaborian space.* IEEE Transactions on Biomedical Engineering, vol. 36, 115-129, 1989.

[80] P. Pudil, J. Novovicova and J. Kittler : *Floating search methods in feature selection.* Pattern Recognition Letters, vol. 15, 1119-1125, 1994.

[81] T. Ojala and Matti Pietikäinen : *Unsupervised texture segmentation using feature distrubtions.* Pattern Recognition, vol. 32, 477-486, 1999.

[82] T. Randen and J. Husoy : *Filtering for texture classification : a comparative study.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, 291-310, 1999.

[83] T. Randen and J. H. Husoy : *Texture segmentation using filters with optimized energy separation.* IEEE Transactions on Image Processing, vol. 8,no. 4, april 1999.

[84] T. R. Reed and J.M. Hans du Buf : *A review of recent texture segmentation and feature extraction techniques.* CVGIP : Image Understanding, vol. 57, 359-372, 1993.

[85] O. Rioul : *A discrete-time multiresolution theory*, IEEE Transactions on Signal Processing, vol. 41, no. 8, 2591-2606, 1993.

[86] A. Ron and Z.Shen : *Construction of compactly Supported Affine Frames in* $L^2(\mathbb{R}^d)$ Springer-Verlag Electronic Production, 1997.

[87] A. Rosenfeld and E. Troy : *Visual texture analysis*, Conference Record for Symposium on Feature Extraction and Selection in Pattern Recognition, Argonne, IL , IEEE publication , 115-124, 1970.

[88] G.Saporta : *Probalités, analyse de données et statistiques*, 1990, Editions Technip.

[89] J.Strand and T.Taxt : *Local frequency features for texture classification.* Pattern Recognition, vol. 27, no. 10, 1397-1406,

[90] W. Sun and X. Zhou : *Sampling theorem for wavelet subspaces : error estimate and irregular sampling.* IEEE transactions on signal processing, vol. 48,no. 1, 223-226, 2000.

[91] W. Sweldens : *The lifting scheme : a construction of the second generation wavelets.* SIAM J. of Math. Analysis, vol. 29, 511-546, 1997.

[92] A. Teuner, O. Pichler and J. Hosticka : *Unsupervised textures segmentation of images using tuned matched gabor filters.* IEEE Transactions on Image Processing, vol. 4, no. 6, 863-870, 1995.

[93] R. Thomas and R. D'Ari : *Biological Feedback.* Boston : CRC Press, 1990.

[94] A. Tonnelier, S. Meignen, H. Bosch and J. Demongeot : *Synchronization and desynchronization of neural oscillators.* Neural Networks, vol. 12, 1213-1228, 1999.

[95] M.Unser : *Texture classification and segmentation using wavelets frames.* IEEE Transactions on Image Processing, vol. 4, no. 11, 1549-1560, 1995.

[96] M. Unser and A. Aldroubi : *A review of wavelets in bomedical applications.* Proceedings of the IEEE, vol. 84,no. 4, 626-638, 1996.

[97] M. Unser and M. Eden : *Multiresolution feature extraction and selection for texture feature segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, 717-727, 1989.

[98] G. Van de Wouver : *Wavelets for multiscale texture analysis.* Ph.D thesis, Anvers University, 1998.

[99] G. Van de Wouver, P. Scheunders and D. Van Dyck : *Statistical texture characterization from discrete wavelet representation.* IEEE Transactions on Image Processing, vol. 8, no. 4, 592-598, 1999.

[100] M. Vetterli and C. Herley : *Wavelets and filter banks : theory and design.* IEEE Transactions on signal processing, vol. 40, no. 9, 2207-2232, 1992.

[101] VisTex. Color image database. http://www-white.media.mit.edu/vismod/imagery/VisionTexture, MIT Media Lab, 1995.

[102] C. Von der Malsburg : *Models of Neural Networks II.* In E. Domany, J.L. van Hemmen and K. Schulten, chapter 2. Berlin : Springer-Verlag, 1981.

[103] D.L. Wang : *Synchronous oscillations based on lateral connections.* In J.Sirosh, R. Miikkulainen and Y. Choe (Eds.), Lateral connections in the cortex : structure and function, 1996.

[104] A.T. Winfree : *Patterns of phase compromise in biological cycles.* Journal of Mathematical Biology,vol. 1, 73-95, 1974.

[105] H.R. Wilson and J.D. Cowan : *Excitatory and inhibitory interactions in localized populations of model neurons.* Biophysical Journal, 1-24, 1972.

[106] X-G. Xia, C.C. Jay Kuo, Z. Zhang : *Wavelet coefficient computation with optimal prefiltering.* IEEE transactions on signal processing, vol. 42, no. 8, 2191-2197, 1994.

[107] S. R. Yhann and T. Y. Young : *Boundary localization in texture segmentation.* IEEE Transactions on Image Processing, vol. 4, no. 6, 849-856, 1996.

[108] C. Zarowski : *An approach to initializing the wavelet packet transform.* IEEE signal processing letters,vol. 4, no. 5, 132-135, 1997.

[109] X-P. Zhang, L-S. Tian and Y-N Peng : *From the wavelet series to the discrete wavelet transform-the initialization.* IEEE transactions on signal processing, vol. 44, 129-133, 1996.

Résumé

Nous nous intéressons dans cette thèse à la segmentation d'images texturées à l'aide de frame d'ondelettes. Après avoir étudié les frames, en particulier les relations qui lient l'approche continue à l'approche discrète, nous nous focalisons particulièrement sur l'étude des problèmes d'échelle liés à la segmentation des macro-textures. Nous proposons des algorithmes particulièrement adaptés au traitement spécifique de ce type de textures, qui utilisent de nouvelles caractéristiques : les facteurs principaux d'une analyse en composantes principales. Des liens sont établis avec des caractéristiques plus communément utilisées. Nous montrons ensuite que la régularité Lipschitzienne des textures est un paramètre moins fiable que les facteurs principaux, aux petites échelles. Enfin, nous nous intéressons à la définition des frontières des textures, et construisons un algorithme permettant leur reconstruction partielle. La seconde partie est consacrée à l'étude d'un algorithme de décomposition multi-échelles permettant le suivi d'extrema le long des échelles. Nous regardons les applications possibles de cet algorithme à la détection des bords d'images bruitées ; puis, ces techniques de suivi d'extrema sont appliquées à la détermination du temps de réaction du muscle génioglosse, sur des enregistrements EMG.

Mots-clés: segmentation , textures, ondelettes, analyse en composantes principales, détection d'extrema par ondelettes

Abstract

We study textured images segmentation using wavelet frames. After a careful study of wavelet frames, we investigate the problem of macro-textures image segmentation using such frames. We propose a new kind of algorithm to treat this issue, based on Karhunen-Loève approximations of the textures and the direction of the corresponding eigenvectors. Links are established with commonly used features for textures characterization. We then estimate the Lipschitz regularity of some textures, to show its unstability with the size of the neighbourhood used for computation. Last, we address the problem of the definition of the texture boundaries and we give an insight of what makes its particularity. In a second part, we focus on an algorithm that enables to follow wavelet extrema through scale, known as Berkner transform. We modify it a little and show reconstruction formula. Applications to edges detection in noisy images is then discussed. We end by developping an algorithm to assess the reaction time of a muscle to a stimulus using Berkner decomposition.

Keywords: segmentation, textures, wavelets, Karhunen-Loève approximations, wavelet maxima detection.