Chapitre 6

Analyse de textures géométriques

Les images naturelles sont composées d'un ensemble de parties relativement homogènes que l'on appelle textures. Cette définition est vague et demande à être confrontée à l'expérience, par exemple la prise de photographies comme dans l'album de textures de Brodatz [22].

L'importance des textures pour la perception a été initialement remarquée par Gibson [87] mais le travail de Julesz [103] a été l'un des premiers à essayer de donner un sens à la notion de texture. En graphisme 3D, les textures sont des éléments essentiels pour enrichir une image de synthèse, voir l'article de synthèse de Heckbert [91].

Dans cette partie, nous exposons de façon informelle un modèle pour les textures géométriques ayant un comportement turbulent. Une transformée en bandelettes utilisant un champ d'association multi-échelles est utilisé afin d'exploiter ce type de régularité. La géométrie peut ainsi être représentée à l'aide d'un champ que l'on peut modéliser de façon statistique. Une application à la synthèse de textures turbulentes permet de valider le modèle et la transformée.

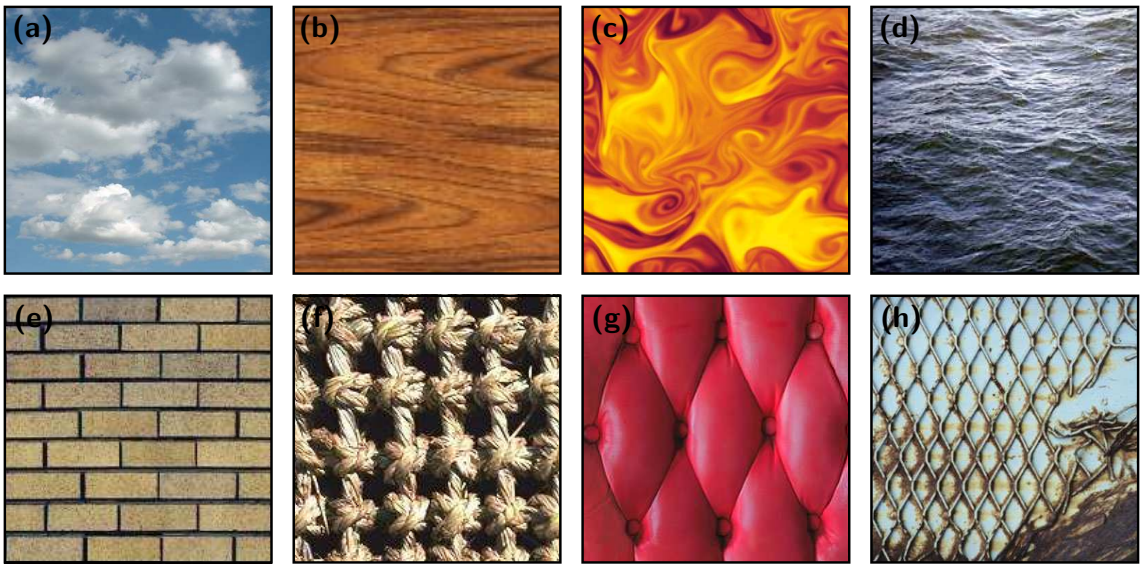

Fig. 6.1 *Exemples de structures géométriques naturelles et synthétiques.*

6.1 INTRODUCTION

6.1.1 TEXTURES NATURELLES ET SYNTHÉTIQUES

Bien que certaines textures aient une nature fractale peu régulière (comme par exemple un nuage, image 6.1 (a)), de nombreux phénomènes naturels créent des textures avec des structures géométriques (comme par exemple les rainures du bois, figure 6.1 (b)). La géométrie de ces textures naturelles est souvent turbulente et elles sont même souvent issues de la physique des fluides (figure 6.1 (c-d)).

Les textures synthétiques, le plus souvent crées par l'homme, sont encore plus régulières et géométriques (figure 6.1 (e-h)). Possédant souvent de nombreuses symétries et périodicités, cette classe de textures est difficile à appréhender à l'aide d'outils purement statistiques. Nous allons donc essentiellement nous concentrer sur les textures possédant une géométrie turbulente.

6.1.2 MODÉLISATION DES TEXTURES GÉOMÉTRIQUES

Le modèle mathématique introduit à la section 1.1.1 pour l'analyse d'images géométriques ne convient pas à l'analyse des textures telles que celles représentées à la figure 6.1 (b-d). Ces textures ne sont pas régulières par morceaux, puisqu'elles sont composées d'un grand nombre de stries turbulentes. Ces stries sont relativement parallèles mais elles sont amenées à se rapprocher voire à se rencontrer, ce qui crée des chocs et des points singuliers.

La transformée en bandelettes orthogonales présentée aux chapitres précédents n'est pas adaptée à l'analyse de ce type de régularité. La subdivision du domaine suivant un quadtree est trop rigide et ne se prête pas à une analyse statistique et la subdivision fixe utilisée pour la compression d'image (voir section 1.5.2) réduit beaucoup trop la longueur des courbes que l'on peut analyser. Tout en suivant le principe de base des bandelettes orthogonales d'appariement de points sur un domaine de coefficients d'ondelettes, on va utiliser à la section 6.3 une nouvelle transformée adaptée. Cette transformée a été introduite par Stéphane Mallat et fait l'objet d'un brevet [130]. Elle utilise un champ d'association multi-échelles qui paramétrise une transformée adaptée.

6.2 SYNTHÈSE DE TEXTURES

Le problème de la synthèse de textures consiste à générer une image texturée se conformant à un modèle, à des mesures, ou bien à un exemple. Juger de la qualité d'une texture synthétisée est essentiellement un problème subjectif qui touche à la perception humaine. Ce problème complexe est cependant simple à mettre en pratique. Dans le cas où l'on dispose d'un modèle, on peut par exemple superposer un échantillon de la texture synthétisée sur la texture d'origine. La synthèse est un succès si l'œil humain n'arrive pas à retrouver la trace de la nouvelle texture.

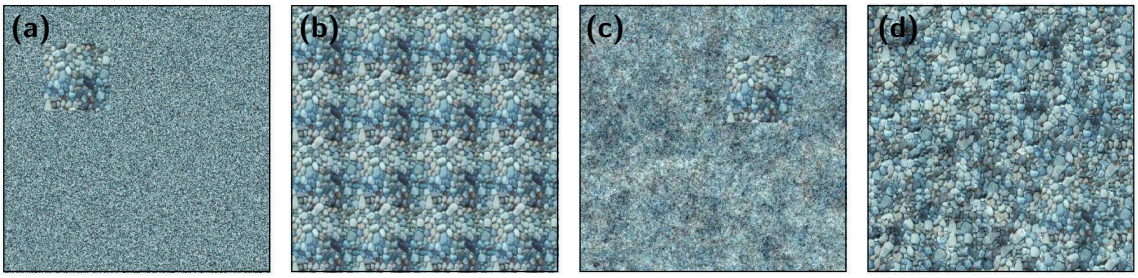

Fig. 6.2 *Comparaison de différentes méthodes de synthèse de textures.*

La figure 6.2 montre ce procédé de validation sur quatre méthodes de synthèse de plus en plus complexes (et satisfaisantes).

- En (a), on génère une image de bruit possédant les mêmes couleurs.
- En (b), on recopie côte-à-côte l'échantillon d'origine.
- En (c), on génère une image de bruit possédant les mêmes histogrammes de coefficients d'ondelettes (voir section 6.2.2)
- En (d), on génère une nouvelle image en recopiant de façon intelligente des parties de la texture d'origine (algorithme de Efros et Leung [78]).

6.2.1 État de l'art

Comme expliqué par Haralick [90], les méthodes de synthèse de textures peuvent être classifiées comme structurelles ou statistiques. Les méthodes statistiques sont généralement de bas niveau et permettent une modélisation globale de la classe de textures à synthétiser. Les méthodes structurelles essaient d'assembler les objets constitutifs de la texture.

Textures naturelles et méthodes procédurales. L'idée originale de Pearson [148] est de modéliser des textures naturelles (taches, zébrures, etc.) à partir d'équations aux dérivées partielles non-linéaires. L'équation de réaction-diffusion a été utilisée en graphisme par Witkin et Kaas [203] puis par Turk [187].

Les méthodes par agrégation de cellules de Fleischer et al. [83] et par subdivision de Worley [207] sont des variations autour des méthodes de réaction-diffusion. Dans toutes ces méthodes, la géométrie se crée progressivement à partir d'interactions locales.

Une autre classe de méthodes, ne nécessitant pas de résolution d'équations différentielles, utilise la génération de coefficients dans un domaine transformé. L'idée originale de Perlin [150, 151] consiste à sommer des bruits blancs filtrés à différentes échelles. Ce type de méthodes paramétriques (aussi appelées procédurales) est étudié plus en détails à la section 6.2.2, comme point de départ de méthodes non paramétriques utilisant des mesures issues d'une texture modèle.

De nombreuses autres méthodes en graphisme essaient de modéliser directement des textures naturelles, comme par exemple en simulant l'évolution de l'apparence des surfaces comme le font Dorsey et al. [74] ainsi que Chen et al. [41].

Modélisation statistique. Les approches statistiques dédiées à l'analyse et à la synthèse de textures, bien que n'étant pas les plus efficaces pour des applications graphiques, proposent une vraie modélisation de la classe de textures. Ainsi les méthodes par calculs des histogrammes de coefficients en ondelettes de Heeger et Bergen [92] sont adaptées aux textures peu régulières voire fractales.

Les méthodes utilisant des champs de Markov ont été proposées initialement par Cross et Jain [51] Chellappa et Kashyap [40] ainsi que Kashyap et Lapsa [105] pour modéliser et synthétiser les textures. Popat [158] a repris ces travaux pour obtenir le premier algorithme efficace de synthèse de textures. L'équipe de Mumford [140, 211] a systématisé et théorisé cette approche pour analyser et synthétiser les dépendances présentes dans les images naturelles.

Les modélisations les plus efficaces des coefficients d'ondelettes utilisent la structure arborescente de la décomposition. Les premières méthodes proposées utilisent une estimation non-paramétrique des histogrammes conditionnels à travers les échelles et ont été appliquées avec succès par De Bonnet [19] puis Paget et Longstaff [147] à la synthèse de textures.

Des méthodes plus directes comme celle de Portilla et Simoncelli [159] utilisent une modélisation paramétrique des histogrammes conditionnels et donnent de bons résultats.

Méthodes par recopie. Les méthodes les plus efficaces de synthèse de textures procèdent par recopie des pixels de l'image d'origine, introduites par Efros et Leung [78] et popularisées par Wei et Levoy [200]. Ces algorithmes gênèrent ainsi les pixels un à un, en cherchant dans l'image d'origine un petit voisinage ressemblant au voisinage que l'on est en train de synthétiser. Les algorithmes plus récents procèdent par recopie de parties entières de textures de façon à conserver les structures d'origine. On peut citer celui d'Ashikhmin [11], d'Efros et Freeman [79], de Kwatra et al. [112] et de Cohen et al. [46].

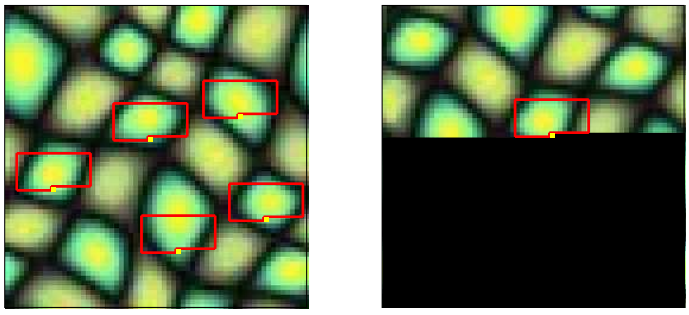

Fig. 6.3 *Construction d'une texture par recherche de meilleurs appariements.*

Des procédures d'optimisation plus globales comme celles de Demanet et al. [61] (pour l'inpainting) et de Kwatra et al. [111] permettent d'avoir une synthèse indépendante de l'ordonnancement des points ainsi que d'incorporer des contraintes additionnelles.

Ce type de méthodes a été aussi appliquées à la désocclusion (ou inpainting) par Crimisi et al. [50] Sun et al. [181] et Demanet et al. [61]. Le problème difficile de la désocclusion est la synthèse de parties texturées tout en respectant les structures géométriques qui touchent le bord de la région à remplir.

Importance de la géométrie. La prise en compte de la géométrie via la détection des contours (voir section 1.3.3) a été utilisée pour la synthèse de textures. Certaines méthodes comme celle de Wu et Yu [208] tentent d'améliorer le respect des bords lors de la recopie. Les algorithmes de Liu et al. [125] et de Matusik et al. [134] exploitent des informations géométriques lors de l'interpolation de textures.

Des méthodes de synthèse de texture comme celle de Neyret [142] utilisent un champ de vecteurs pour synthétiser des textures turbulentes. Elles s'inspirent des algorithmes de visualisation de champs de vecteurs, dont les plus connus sont le *Spot Noise* de Wijk [190] ainsi que LIC de Cabral et [142][27]. Ces algorithmes calculent des convolutions curvilignes le long du flot pour générer une texture turbulente.

La géométrie d'une texture peut aussi être analysée à l'aide des symétries et périodicités de l'image d'origine, comme expliqué dans les travaux de Liu et son équipe [123, 124].

6.2.2 Génération de texture et représentation creuse

Synthèse et régularité. Une méthode simple pour générer des fonctions avec une régularité globale proche de C^α consiste à utiliser une réalisation d'un processus stationnaire X dont la puissance spectrale $P_X(\omega)$ satisfait

$$P_X(\omega) = |\omega|^{-2\alpha}. \tag{6.1}$$

La figure 6.4 montre des exemples de telles fonctions 2D pour différentes valeurs de α.

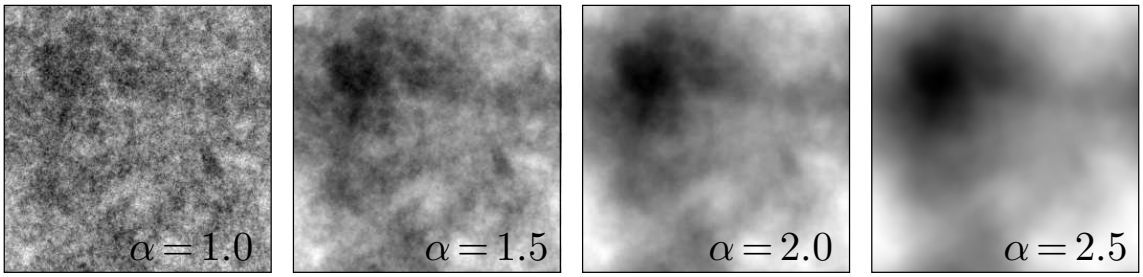

Fig. 6.4 *Textures stochastiques par filtrage de la transformée de Fourier.*

Cependant, comme le montre la figure 6.4, la puissance spectrale n'est pas caractéristique des images naturelles (voir aussi la section 1.3.1 pour une discussion sur ce sujet).

La transformée de Fourier est capable de mesurer la régularité globale d'une fonction. Pour analyser la régularité locale, qui peut éventuellement varier d'un point à l'autre, il faut utiliser une transformée en ondelettes. La décroissance des coefficients en ondelettes est en effet quasiment caractéristique de la régularité hölderienne, puisqu'au voisinage d'un point x où une fonction bidimensionnelle f est C^α, on a

$$|\langle f, \psi_{jn}\rangle| \leqslant C\, 2^{j(\alpha+1)} \qquad \text{pour} \qquad 2^j n \text{ voisin de } x.$$

Il est donc naturel d'utiliser la transformée en ondelettes pour analyser des signaux avec un comportement fractal [10].

Pour la synthèse de textures, les premières méthodes ont été proposées par Perlin [150, 151] pour créer des images fractales (montagnes, nuages, etc). Elles consistent à sommer des

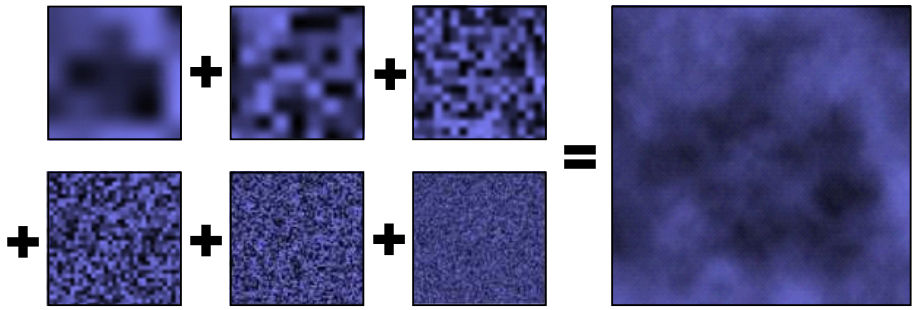

Fig. 6.5 *Exemple de construction d'un Perlin Noise.*

bruits blancs filtrés à plusieurs résolutions. La figure 6.5 montre une image de nuages ainsi obtenue. Cette méthode donne des résultats assez proches des bruits obtenus par filtrage de fourier (image 6.4) mais avec une souplesse bien plus grande offerte par la manipulation des différentes échelles. Cette méthode a été améliorée par Cook et DeRose [49] en utilisant des processus gaussiens dont les coefficients en ondelettes suivent une loi gaussienne de variance dépendant de l'échelle (voir figure 6.6).

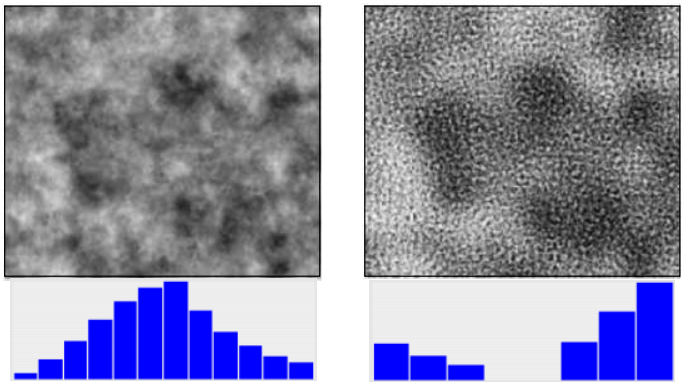

Fig. 6.6 *Exemples de bruits en ondelettes.*

Synthèse par égalisation d'histogrammes. Pour la synthèse non paramétrique, Heeger et Bergen [92] utilisent l'égalisation des histogrammes des coefficients d'ondelettes. Cette méthode donne de bons résultats pour les textures homogènes et fractales. L'idée est d'utiliser une réalisation d'un processus aléatoire dont les coefficients multi-échelles sont indépendants et suivent une loi fixée à chaque échelle, qui est estimée à partir de la texture d'origine.

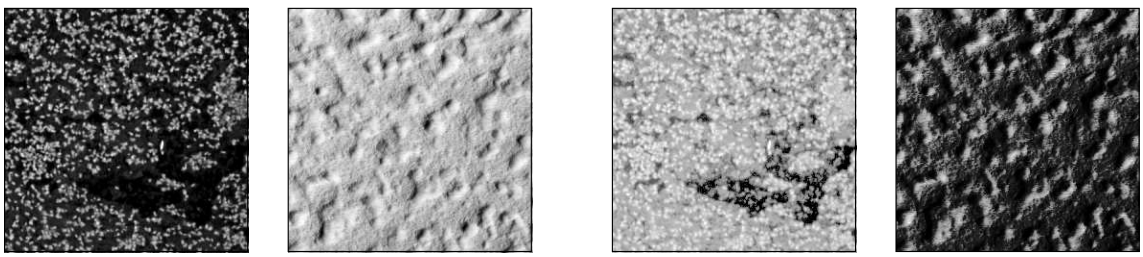

Fig. 6.7 *Exemple d'égalisation d'histogrammes.*

L'égalisation d'histogrammes consiste à composer une fonction g par une fonction croissante

φ de telle manière que l'histogramme des valeurs de $\varphi \circ g$ soit proche de celui d'une fonction f donnée à l'avance. Pour des images f et g composées de pixels prenant des valeurs dans $\{0, \ldots, 255\}$, l'algorithme d'égalisation est simple. Il consiste à calculer les histogrammes p et les distributions cumulées c

$$\forall\, y \in \{0, \ldots, 255\}, \quad p_f[y] \stackrel{\text{def.}}{=} \operatorname{Card}\{x \setminus f[x] = y\} \qquad \text{et} \qquad c_f[y] \stackrel{\text{def.}}{=} \sum_{t=0}^{y} p_f[t].$$

En supposant pour simplifier que c_g est croissante, on peut calculer $\varphi \stackrel{\text{def.}}{=} (c_g)^{-1} \circ c_f$. La figure 6.7 montre un exemple d'égalisation d'histogrammes.

Le pseudo-code 2 explique le principe de l'algorithme égalisation des histogrammes de coefficients d'ondelettes. La figure 6.8 montre des exemples de succès de cette méthode mais aussi les problèmes qui ont lieu lorsque l'on s'intéresse à des images avec des structures géométriques.

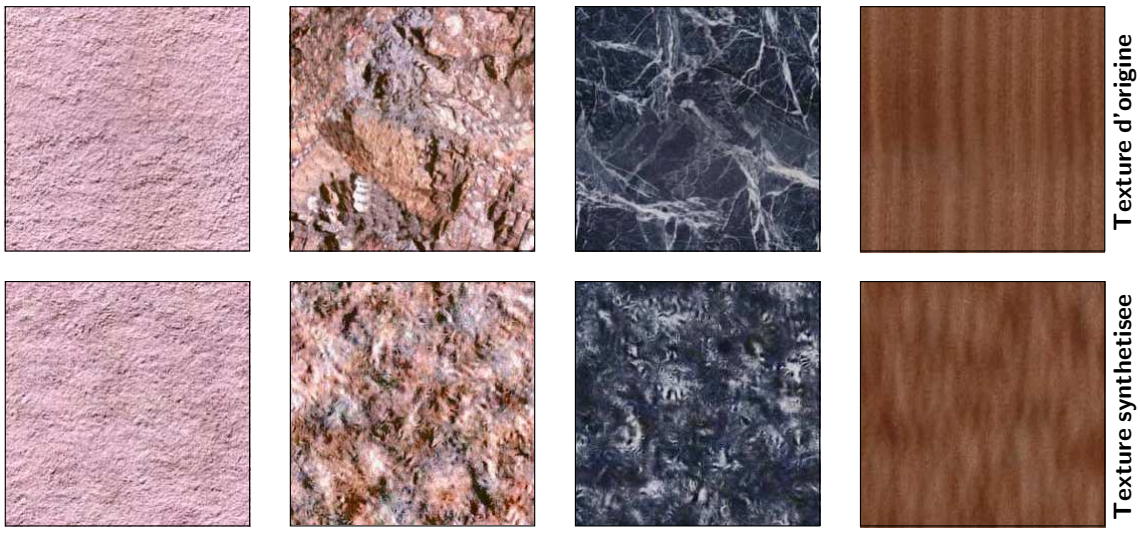

Fig. 6.8 *Exemples de synthèse de texture par égalisation d'histogrammes. Textures fractales (gauches) et textures avec des structures géométriques (droite).*

Indépendance et représentation creuse. Toutes les méthodes de génération de coefficients aléatoires dans un domaine transformé consistent à imposer une certaine représentation creuse de la texture. Ainsi la synthèse sera un succès si la classe de textures considérée se représente de façon creuse à l'aide de la transformation employée. Pour éviter d'avoir à modéliser les corrélations qui subsistent dans une transformée en ondelettes, une alternative est donc d'utiliser une transformée qui représente de façon plus creuse les textures qui nous intéressent.

Nous souhaitons utiliser une approche d'égalisation d'histogrammes semblable à celle de Heeger et Bergen [92] mais en étendant la méthode à des textures ayant des structures géométriques. Ceci aura pour avantage d'éviter les méthodes complexes d'échantillonnages de distributions de probabilité (échantillonneur de Gibbs [140], projections alternées [159], etc), tout en ayant une réelle intuition sur les propriétés des textures simulées, via une construction add-hoc de la transformée.

Il faut donc garantir que les histogrammes de coefficients transformés sont représentatifs de la classe de textures que l'on souhaite synthétiser.

Fonction B = `synthese_egalisation`(A, B).

Entrée : texture A.
Sortie : texture synthétisée B.

Initialisation : B = `bruit_blanc`().

Calcul de la transformée de l'image d'origine :
$[A_0, A_1, \ldots, A_m]$ = `transforme`(A).
Répéter jusqu'à convergence :
Égalisation spatiale : B = `egalisation`(A, B).
Passage en domaine transformé : $[B_0, B_1, \ldots, B_m]$ = `transforme`(M).
Pour chaque $j = 0, \ldots, m$: B_j = `egalisation`(A_j, B_j).
Passage en domaine spacial : B = `transforme_inverse`($B_0, \ldots, B_m$).

N.B. : Les fonctions `transforme` et `transforme_inverse` sont inverses l'une de l'autre et calculent une représentation multirésolution de l'image. La fonction `egalisation` réalise l'égalisation d'histogrammes.

Table 2: Pseudo-code pour la synthèse de texture par égalisation d'histogrammes.

6.3 TRANSFORMÉE EN BANDELETTES PAR GROUPEMENT

On souhaite réaliser une transformation de l'image qui donne une représentation creuse d'une texture géométrique. Le but est ainsi d'avoir une représentation dont les coefficients sont au maximum décorrélés les uns des autres pour avoir une modélisation simple à l'aide de statistiques du premier ordre (c'est-à-dire à l'aide des histogrammes des coefficients).

On s'autorise beaucoup de souplesse dans la construction de la représentation, en particulier, on renonce à l'orthogonalité et on va même construire une représentation redondante. Ainsi on pourra retrouver par exemple une certaine invariance par translation qui fait défaut à la représentation en ondelettes et en bandelettes orthogonales.

L'invariance par translation est une propriété naturelle lorsque l'on ne s'intéresse pas à la compression d'un signal. Donoho et Johnstone [70] ont utilisé la transformée en ondelettes redondante pour le débruitage et Li et al. [120] utilisent le même type de transformations pour la modification de contraste.

6.3.1 FLOT GÉOMÉTRIQUE ET FLOT OPTIQUE

La géométrie d'une texture peut être représentée à l'aide d'un flot le long duquel s'alignent les motifs de la texture. La figure 6.9 montre un possible flot de texture. Des algorithmes de calculs ont été proposés par Ben-Shahar et Zucker [17] ansi que par l'équipe de Morel [64]. Ces méthodes reposent sur des hypothèses proches de la théorie de la Gestalt comme par exemple la bonne continuation, voir par exemple le livre fondateur de Wertheimer [202].

Dans la suite de cette section, nous allons décrire une transformée géométrique qui évite la

construction d'un tel flot. A la place nous allons utiliser le concept de champ d'association. Cette notion est fréquemment employée pour décrire les interactions parmi les neurones de V1 dans le cortex visuel, voir la sous-section 1.3.2 pour plus de détails.

Dans cette sous-section, nous donnons quelques pistes pour comprendre pourquoi la notion de flot de texture (ou en vidéo de flot optique) est mal adaptée au problème de la représentation de la géométrie.

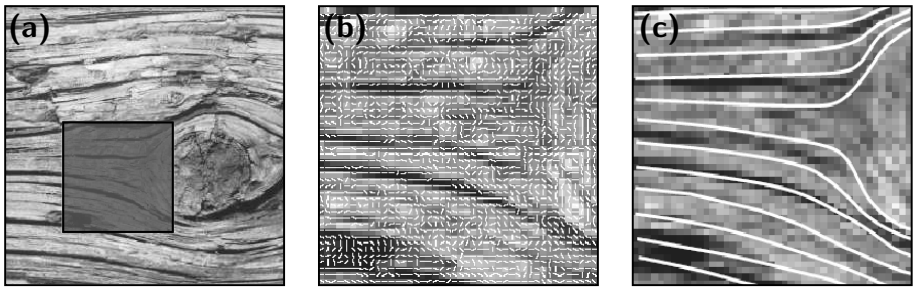

Fig. 6.9 *Géométrie d'une texture de bois.*

Tout comme la recherche des contours d'une image (voir section 1.3.3), le problème de l'extraction d'un tel flot est mal posé, car la texture est souvent buitée et plus ou moins floue. Pour obtenir un flot régulier, les méthodes classiques basées sur des quantités différentielles doivent être régularisées comme le font Xu et Prince [210] ou bien chaînées globalement par exemple avec la méthode de *tensor voting* introduite par Medioni et son équipe [135].

Cependant, pour l'utilisation d'un flot géométrique à des fins de compression ou de modélisation statistique, il faut un flot capable de suivre précisément et sur des distances longues les motifs de la texture. La figure 6.10 montre une approche d'extraction de la géométrie sous forme de flot à chaque échelle d'une transformée en ondelettes invariante par translation. On voit que cette approche n'arrive pas à relier des coefficients en ondelettes semblables mais très distants, car il est difficile, avec des mesures locales, d'obtenir des lignes intégrales qui ne divergent pas.

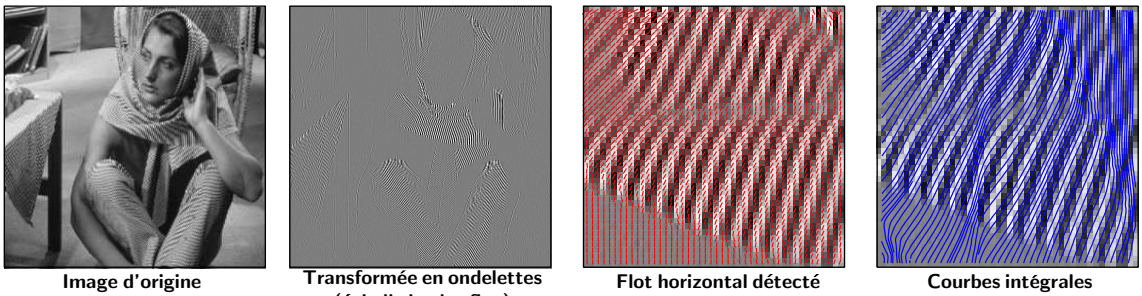

Fig. 6.10 *Exemple de flot et de lignes intégrales calculées sur une transformée en ondelettes l'algorithme de Horn et Schunk* [98] sur des tranches 1D le long de la direction verticale.

Ce problème est ainsi voisin du calcul de flot optique en analyse et compression video (bien que notre problème soit 2D alors que la video contient une dimension temporelle supplémentaire). La détection de flot optique est régularisée soit en calculant des correspondances entre images sur des voisinages assez larges comme dans l'algorithme de Lucas et Kanade [126], soit par des méthodes variationnelles globales en espace comme l'algorithme de Horn et Schunk [98] voir en temps et en espace comme expliqué par Weickert et Schnörr [201].

6.3.2 CHAMP D'ASSOCIATION MULTI-ÉCHELLES

La transformée en bandelettes par groupement est un algorithme qui a été développé par Stéphane Mallat au sein de la société Let It Wave et qui est déposée dans le brevet [130]. Ce travail utilise un cas particulier de cet algorithme qui est adapté a notre problème d'analyse de texture.

L'idée principale est d'exploiter la redondance géométrique qui s'exerce sur de longues distances en définissant un champ d'association multi-échelles. Ce champ est utilisé pour réaliser groupements entre des couples de points de plus en plus distants dans l'image. La figure 6.11 montre une portion d'un tel champ le long d'une singularité géométrique turbulente. Les vecteurs représentent quelques exemples de couples de points que l'on apparie le long de la géométrie. Ce champ se construit petit à petit depuis les courtes distances (échelle fine) vers les distances plus longues (échelles grossières).

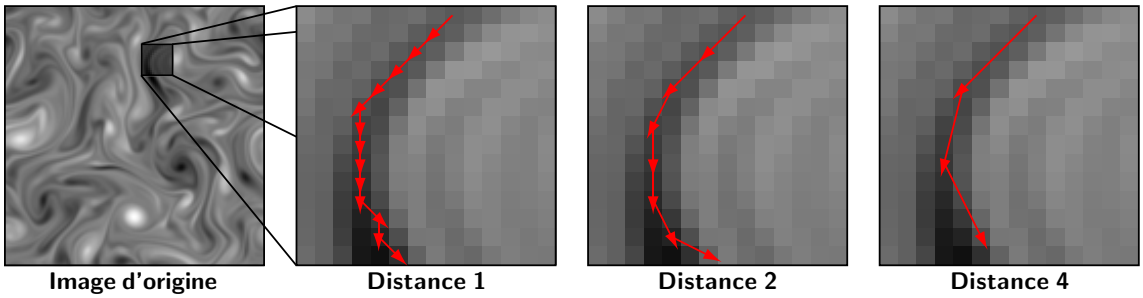

Fig. 6.11 *Illustration du principe d'un champ d'association multi-échelles.*

On peut rapprocher cette construction d'un procédé de quantification de la géométrie. Ainsi plus la géométrie doit relier des points éloignés, plus elle doit être quantifiée de façon précise. Cette méthode permet d'assurer un suivi de la singularité sur une longue distance, ce qui est impossible avec un algorithme calculant un flot à la résolution la plus fine (voir figure 6.10). En effet, même si le champ courte distance est quelque peu erroné (dû à la faible précision), le champ à l'échelle suivante peut corriger les erreurs en introduisant une petite modification. Ce procédé est assez semblable au système dynamique contrôlant la répartition de l'erreur de quantification d'un codeur Σ/Δ, étudié en détails par Daubechies et al. [57].

Dans les sous-sections suivantes, on va expliquer comment on peut calculer ce champ grâce à des appariements de voisinages. Tout comme le font les méthodes de calcul de flot optique décrites à la section précédente, l'utilisation d'un petit voisinage autour de chaque point permet de régulariser le champ ainsi détecté pour avoir un problème de représentation bien posé.

Pour le débruitage d'images, des méthodes par appariement de voisinages ont été récemment proposées par Buades et al. [24], qui sont inspirées des méthodes de synthèse de textures par recopie telles celles de Efros et Leung [78]. Cependant, le problème du débruitage est plus simple que celui de la compression ou de l'analyse de textures. En effet, il n'est pas nécessaire de calculer explicitement un flot optique, comme l'explique Buades et al. [23] et encore moins d'imposer une régularité de ce flot (ce qui est nécessaire pour éviter les ambiguïtés locales causées par la présence de contours).

Pour obtenir une représentation creuse d'une image, au contraire du débruitage, il ne suffit pas d'avoir des petits coefficients transformés, il faut aussi avoir un champ d'association

géométrique régulier. En effet, ce champ, qui paramétrise la transformée, fait partie intégrante de la représentation de l'image et doit donc aussi être codé à l'aide de peu de coefficients (pour avoir une compression efficace ou bien une modélisation statistique qui capture les caractéristiques de l'image).

6.3.3 Transformée en bandelettes par groupements

La transformée de Haar est la plus simple des transformées en ondelettes. Une étape de la transformée consiste à remplacer deux coefficients voisins (a, b) par leur moyenne et leur différence

$$(a, b) \qquad \longrightarrow \qquad (m, d) \stackrel{\text{def.}}{=} \left(\frac{a+b}{\sqrt{2}}, \frac{a-b}{\sqrt{2}} \right). \tag{6.2}$$

Cette transformée a l'avantage d'avoir une interprétation intuitive sous forme de moyennage et d'être très facilement inversible. L'idée est de partir de cette étape élémentaire et de l'appliquer de façon successive sur des couples de deux pixels.

Une étape de l'algorithme sur toute l'image. On suppose donc que l'on dispose d'une liste de couples de pixels (x_k, y_k) avec x_k et $y_k \in \{0, \ldots, n-1\}^2$, où n est la taille de l'image M que l'on souhaite transformer. On impose que les points x_k soient tous différents et dans la pratique, on choisit tous les points de l'image M. On définit une matrice de poids W initialisée à $\mathrm{W}[x] = 1$ pour tous les points x, ainsi qu'une matrice de détails D qui sera remplie séquentiellement.

À chaque étape k de la transformée, on effectue la mise à jour

$$\left\{ \begin{array}{l} \mathrm{M}[y_k] \leftarrow (\mathrm{W}[x_k]\,\mathrm{M}[x_k] + \mathrm{W}[y_k]\,\mathrm{M}[y_k])/(\mathrm{W}[x_k] + \mathrm{W}[y_k]), \\ \mathrm{D}[x_k] \leftarrow (\mathrm{M}[x_k] - \mathrm{M}[y_k])\sqrt{\mathrm{W}[x_k]\mathrm{W}[y_k]}/\sqrt{\mathrm{W}[x_k] + \mathrm{W}[y_k]}, \\ \mathrm{D}[y_k] \leftarrow \mathrm{W}[x_k] + \mathrm{W}[y_k]. \end{array} \right.$$

L'algorithme va progressivement remplir la matrice D de détails, en mettant en même temps à jour les matrices de poids et de moyenne M (qui remplace l'image de départ).

Le point important est de choisir de façon intelligente ces couples de pixels. Tout d'abord, on considère les pixels x_k selon un ordre prédéfini (par exemple ligne après ligne). On va parcourir plusieurs fois l'ensemble des pixels (chaque passage sur toute l'image correspondra à une échelle de la transformée de Haar). Dans la suite on note $\ell \in \{0, 1, \ldots, \mathrm{L}-1\}$ le numéro de l'échelle considérée. À chaque pixel x_k on associe un pixel $y_k = x_k + \mathrm{F}_\ell[x_k]$, où F_ℓ est le champ d'association géométrique à l'échelle ℓ.

Choix des couples de points. Pour chaque pixel x_k, on calcule le champ d'association F_ℓ selon un critère géométrique. Si on note

$$x_k + \mathrm{V} = \{x_k + t \ \backslash \ t \in \mathrm{V}\} \qquad \text{avec} \qquad \mathrm{V} = \{-d/2, \ldots, d/2\}^2$$

le voisinage de $d \times d$ pixels autour de x_k (où d est une constante), on cherche le pixel y_k ayant le voisinage le plus proche. Plus précisément, on choisit

$$y_k = \operatorname*{argmin}_{y \in \Delta_\ell(x_k)} \sum_{t \in \mathrm{V}} |\mathrm{M}[x_k + t] - \mathrm{M}[y + t]|^2 .$$

La région $\Delta_\ell(x_k)$ est un sous-ensemble de pixels, qui force les vecteurs du champ d'association F_ℓ à avoir une taille d'environ 2^ℓ. Elle est définie à l'aide du champ précédent $\mathrm{F}_{\ell-1}$ de la façon suivante :

$$\Delta_\ell(x_k) = ((u_k + \mathrm{F}_{\ell-1}[u_k]) + \mathrm{V}) \cap (\mathrm{F}_{\ell-1}[x_k])^+ \qquad \text{avec} \qquad u_k \stackrel{\text{def.}}{=} x_k + \mathrm{F}_{\ell-1}[x_k]$$

et où l'on a noté $(\mathrm{F}_{\ell-1}[x_k])^+$ le demi plan défini par

$$(\mathrm{F}_{\ell-1}[x_k])^+ \stackrel{\text{def.}}{=} \{x \neq x_k \ \setminus \langle x - (u_k + \mathrm{F}_{\ell-1}[u_k]),\, \mathrm{F}_{\ell-1}[x_k] + \mathrm{F}_{\ell-1}[u_k] \rangle > 0 \}$$

On doit restreindre la recherche à une zone de plus en plus éloignée du point x_k pour obliger la transformée à aller moyenner des valeurs correspondant à des coefficients de plus en plus distants. Ceci est nécessaire pour pouvoir associer la valeur de ℓ à une échelle de bandelettes.

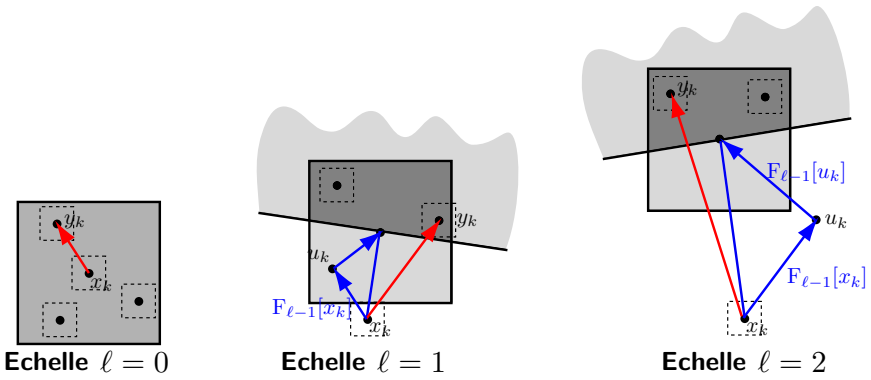

Fig. 6.12 *Recherche du champ d'association par meilleur appariement. À chaque échelle, la région $\Delta_\ell(x_k)$ est la zone la plus foncée.*

La figure 6.12 montre trois exemples de recherche de champs d'association à des échelles ℓ différentes. On voit que le carré de recherche $((u_k + \mathrm{F}_{\ell-1}[u_k]) + \mathrm{V})$ est à chaque fois plus éloigné du point x_k. La condition $y_k \in \Delta_\ell(x_k)$ empêche le champ d'association de revenir en arrière.

Algorithme de calcul en plusieurs étapes. Pour obtenir l'algorithme complet, on effectue la transformation détaillée précédemment de façon répétée pour plusieurs échelles $\ell = 0, \ldots, \mathrm{L}-1$. Chaque étape, pour une échelle ℓ fixée, stocke les matrices de détails D dans une variable D_ℓ ainsi que le champ d'association F_ℓ.

Le pseudo-code 3 détaille les différentes étapes de l'algorithme qui est un cas particulier du brevet [130]. Cet algorithme renvoie des coefficients de détails $\{\mathrm{D}_\ell\}_{\ell=0}^{\mathrm{L}-1}$ ainsi qu'une image basse fréquence D_L. L'algorithme peut être employé dans une version simplifiée où il ne calcule par les champs F_ℓ, ceux-ci étant supposés donnés par l'utilisateur par l'intermédiaire des variables $\widetilde{\mathrm{F}}_\ell$.

La figure 6.13 montre les coefficients de détails, ainsi que le champ d'association (affiché sur l'image de moyenne temporaire A à l'étape ℓ). On voit que le champ est de plus en plus long et qu'il y a très peu de coefficients de détails non nuls.

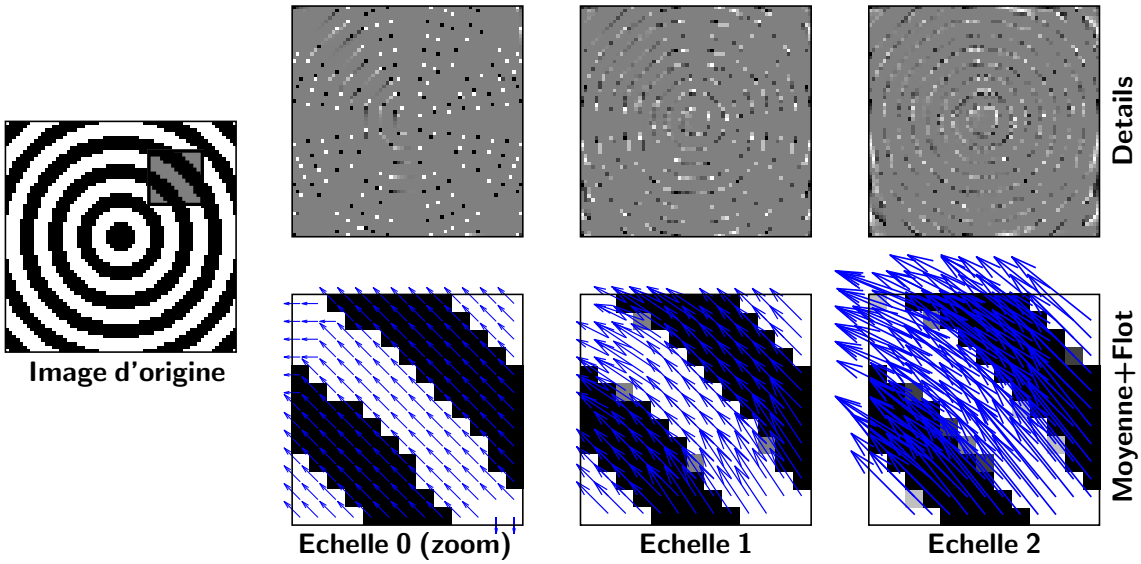

Fig. 6.13 *Champ d'association géométrique multi-échelles détecté par l'algorithme, coefficients de moyenne et de détail.*

Fonction $[\mathrm{D}_0, \ldots, \mathrm{D}_\mathrm{L}, \mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}, \mathrm{W}] =$ `transfo_bandelettes`(M).
ou Fonction $[\mathrm{D}_0, \ldots, \mathrm{D}_\mathrm{L}, \mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}, \mathrm{W}] =$ `transfo_bandelettes`$(\mathrm{M}, \widetilde{\mathrm{F}}_0, \ldots, \widetilde{\mathrm{F}}_{\mathrm{L}-1})$.

Entrée : image M.
Sortie : coefficients de détails $\mathrm{D}_0, \ldots, \mathrm{D}_{\mathrm{L}-1}$,
moyenne basse fréquence D_L, poids W,
champ d'association multi-échelles $\mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}$, avec $\mathrm{F}_\ell[x] \in \mathbb{R}^2$.

Initialisation : $\mathrm{A} = \mathrm{M}$.

Répéter pour chaque échelle $\ell = 0, \ldots, \mathrm{L}-1$:
  Répéter pour chaque pixel x_k :
    Si $\ell > 0$, calcul du centre de la zone de recherche
      $z_k = x_k + \mathrm{F}_{\ell-1}[x_k] + \mathrm{F}_{\ell-1}[x_k + \mathrm{F}_{\ell-1}[x_k]]$,
      (sinon on pose $z_k = x_k$).
    Si le champ $\widetilde{\mathrm{F}}_\ell$ n'est pas fourni, recherche du meilleur appariement :
      pour chaque pixel $y \in z_k + \mathrm{V}$, calculer
        $\mathrm{E}(y) = \sum_{s \in \mathrm{V}} |\mathrm{A}[y+s] - \mathrm{A}[x_k+s]|^2$,
      retenir $y_k = \mathrm{argmin}(\mathrm{E}(y))$ et poser $\mathrm{F}_\ell[x_k] = y_k - x_k$.
    (sinon on pose $\mathrm{F}_\ell[x_k] = \widetilde{\mathrm{F}}_\ell[x_k]$ et $y_k = x_k + \mathrm{F}_\ell[x_k]$)
    Transformée de Haar : poser
      $\mathrm{D}_\ell[x_k] = (\mathrm{A}[x_k] - \mathrm{A}[y_k])\sqrt{\mathrm{W}[x_k]\mathrm{W}[y_k]}/\sqrt{\mathrm{W}[x_k] + \mathrm{W}[y_k]}$,
      $\mathrm{A}[y_k] = (\mathrm{W}[x_k]\mathrm{A}[x_k] + \mathrm{W}[y_k]\mathrm{A}[y_k])/(\mathrm{W}[x_k] + \mathrm{W}[y_k])$,
      $\mathrm{W}[y_k] = \mathrm{W}[x_k] + \mathrm{W}[y_k]$.
Enregistrer les basses fréquences $\mathrm{D}_\mathrm{L} = \mathrm{A}$.

N.B. : La fonction a deux prototypes suivant que l'utilisateur fournit un champ précalculé $\widetilde{\mathrm{F}}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}$, ou bien que l'algorithme doit calculer lui-même le champ $\mathrm{F}_0, \ldots, \mathrm{F}_\ell$ par recherche de meilleurs appariements.

Table 3: Pseudo-code pour la transformée en bandelettes par groupements.

6.3.4 TRANSFORMÉE EN BANDELETTES SUR UN DOMAINE MULTIRÉSOLUTION

Tout comme pour la transformée en bandelettes orthogonales, la transformée complète en bandelettes par groupements s'obtient par la succession d'une transformée multirésolution, puis d'une transformée de Haar le long d'un champ d'association à chaque échelle de la transformation, comme expliqué dans le brevet [130].

La première raison en faveur de cette succession de deux transformées est théorique. Les théorèmes d'approximation en bandelettes orthogonales ne sont valides que si l'on enchaîne une transformée en ondelettes puis une transformée anisotrope adaptée. À l'échelle d'un pixel, les bandelettes (orthogonales tout comme par groupements) ne sont pas capables d'absorber les parties régulières et les transitions lissées. La deuxième raison est pratique puisque les fonctions de base bandelettes ne sont pas régulières si on les considère dans le domaine discret et qu'au contraire les fonctions de base obtenues avec la succession des deux transformées sont régulières (voir section 2.2.3).

Interprétation comme une décomposition. L'algorithme de transformée en bandelettes par groupements présenté à la sous-section 6.3.3 correspond à la décomposition d'une image discrétisée sur une frame $\{g^{\omega}_{\ell m}\}_{\ell m}$ qui dépend d'un champ d'association multi-échelles ω. Le paramètre ω représente ainsi la géométrie paramétrisant la transformation en bandelettes.

Le fait que la famille $\{g^{\omega}_{\ell m}\}_{\ell m}$ forme une frame signifie que la transformée en bandelettes conserve relativement bien l'énergie, c'est à dire qu'il existe deux constantes a et b telles que

$$\forall\, \mathrm{A} \in \Theta, \qquad a\,\|\mathrm{A}\|^2_{\ell^2} \leqslant \sum_{\ell,m} \langle \mathrm{A},\, g^{\omega}_{\ell m}\rangle^2 \leqslant b\,\|\mathrm{A}\|^2_{\ell^2}.$$

Cette propriété n'est pas triviale et nécessite que la géométrie de la texture soit suffisamment régulière. Ceci suppose des conditions techniques sur l'ensemble Θ des textures considérées qui ne sont pas développées dans cette thèse.

On peut appliquer cette transformée par groupements à chaque échelle d'une transformée multirésolution. Dans la suite, on note $\{\mathrm{A}_j\}_j$ les coefficients de la décomposition d'une image A sur une frame multirésolution $\{\psi_{jn}\}_{j,n}$ (par exemple une base d'ondelettes ou une frame laplacienne). Formellement, on a

$$\mathrm{A}_j[n] \stackrel{\text{def.}}{=} \langle \mathrm{A},\, \psi_{jn}\rangle,$$

où j est un paramètre d'échelle et n un paramètre spacial.

Cette transformée par groupements sur chaque ensemble de coefficients A_j permet de calculer

$$\langle \mathrm{A}_j,\, g^{\omega}_{\ell m}\rangle \stackrel{\text{def.}}{=} \langle \mathrm{A},\, g^{\omega}_{j\ell m}\rangle \qquad \text{où} \qquad g^{\omega}_{j\ell m} \stackrel{\text{def.}}{=} \sum_n g^{\omega}_{\ell m}[n]\,\psi_{jn}.$$

On obtient ainsi une frame $\{g^{\omega}_{j\ell m}\}_{j,\ell,m}$ qui dépend de trois paramètres
- l'échelle $0 \leqslant j < \mathrm{J}$ de la transformée multirésolution,
- l'échelle $0 \leqslant \ell < \mathrm{L}$ de la transformée par groupements,
- l'index spatial m de la transformée par groupements.

On remarque que dans la suite, on utilise la même géométrie ω pour toutes les échelles j, bien que ceci ne soit pas a priori requis par l'algorithme. Ceci permet d'avoir une synthèse de textures de meilleure qualité, car il est difficile de prendre en compte une éventuelle modification de la géométrie à travers les échelles.

On note $\{\widetilde{g}^{\omega}_{j\ell m}\}_{j,\ell,m}$ une famille biorthogonale de cette frame, que l'on utilise pour la reconstruction à l'aide de

$$\mathrm{A} = \sum_{j,\ell,m} \langle \mathrm{A},\, g^{\omega}_{j\ell m} \rangle\, \widetilde{g}^{\omega}_{j\ell m}.$$

Cette formule de reconstruction se calcule numériquement à l'aide d'un algorithme de transformée inverse qui fait l'objet du code 4.

Fonction $\mathrm{M} = \texttt{transfo\_bandelettes\_inverse}(\mathrm{D}_0, \ldots, \mathrm{D}_\mathrm{L}, \mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}, \mathrm{W})$.

Entrée : coefficients de bandelettes $\mathrm{D}_0, \ldots, \mathrm{D}_\mathrm{L}$, matrice de poids W
champ d'association multi-échelles $\mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}$, avec $\mathrm{F}_\ell[x] \in \mathbb{R}^2$.
Sortie : image M.

Initialisation : $\mathrm{M} = \mathrm{D}_\mathrm{L}$ (basse fréquences).

Répéter pour chaque échelle $\ell = \mathrm{L}-1, \mathrm{L}-2, \ldots, 1$:
  Répéter pour chaque pixel x_k :
    Pixel couplé $y_k = x_k + \mathrm{F}_\ell[x_k]$, transformée de Haar inverse :
      $\mathrm{M}[y_k] = \mathrm{M}[y_k] - \mathrm{D}[x_k]\sqrt{\mathrm{W}[y_k] - \mathrm{W}[x_k]}/\sqrt{\mathrm{W}[y_k]\mathrm{W}[x_k]}$.

Table 4: Pseudo-code pour la transformée en bandelettes par groupements inverse.

6.4 Synthèse géométrique avec des bandelettes

Dans le but de synthétiser des textures géométriques, nous allons utiliser la transformée en bandelettes par groupements. Nous envisageons deux scénarios pour cette synthèse, suivant que l'on dispose ou non d'une géométrie a priori.

6.4.1 Algorithme générique

L'algorithme générique de synthèse de textures à l'aide de la transformée en bandelettes utilise l'égalisation d'histogrammes des coefficients en bandelettes par groupements calculés sur une représentation multirésolution $\mathrm{A}_0, \ldots, \mathrm{A}_{\mathrm{J}-1}$ de la texture d'origine A.

Dans la suite, on suppose que l'on veut synthétiser une texture discrétisée B de $n \times n$ pixels à partir d'une texture originale A de même taille.

Fonction B = `synthese_bandelettes`(A).

Entrée : texture originale A de taille $n \times n$.
Sortie : texture synthétisée B.

Initialisation : B = `bruit_blanc`(n).

Calcul du champ d'association à synthétiser
$[\widetilde{\mathrm{F}}_1, \ldots, \widetilde{\mathrm{F}}_{\mathrm{L}-1}]$ = `calcul_champ`(A).
Égalisation des histogrammes en espace
A = `egalisation`(A, B).
Calcul de la transformée multirésolution de la texture d'origine
$[\mathrm{A}_0, \ldots, \mathrm{A}_m]$ = `transforme`(A).
Répéter jusqu'à convergence :
  Calcul de la transformée multirésolution de la texture synthétisée
  $[\mathrm{B}_0, \ldots, \mathrm{B}_m]$ = `transforme`(B).
  Répéter pour chaque échelle $j = 0, \ldots, m$:
    Égalisation des histogrammes multirésolution
    A_j = `egalisation`($\mathrm{A}_j, \mathrm{B}_j$).
    Calcul du champ d'association et des coefficients de bandelettes à l'échelle j
    $[\mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}, \mathrm{D}_0, \ldots, \mathrm{D}_{\mathrm{L}}]$ = `transfo_bandelettes`(A_j).
    Calcul des coefficients de bandelettes de l'image à synthétiser à l'échelle j
    $[\widetilde{\mathrm{D}}_0, \ldots, \widetilde{\mathrm{D}}_{\mathrm{L}}]$ = `transfo_bandelettes`($\mathrm{B}_j, \widetilde{\mathrm{F}}_0, \ldots, \widetilde{\mathrm{F}}_{\mathrm{L}-1}$).
    Répéter pour chaque échelle $\ell = 0, \ldots, \mathrm{L}-1$:
      Égalisation des histogrammes en bandelettes
      $\widetilde{\mathrm{D}}_\ell$ = `egalisation`($\widetilde{\mathrm{D}}_\ell, \mathrm{D}_\ell$).
    Calcul de la transformée en bandelettes inverse de la texture synthétisée
    B_j = `transfo_bandelettes_inverse`($\widetilde{\mathrm{D}}_0, \ldots, \widetilde{\mathrm{D}}_{\mathrm{L}}, \widetilde{\mathrm{F}}_0, \ldots, \widetilde{\mathrm{F}}_{\mathrm{L}-1}$).
  Calcul de la transformée en multirésolution inverse de la texture synthétisée
  B = `transforme_inverse`($\mathrm{B}_0, \ldots, \mathrm{B}_m$).

N.B. : La fonction `egalisation` fait l'objet du pseudo-code 2 et la fonction `transfo_bandelettes` est décrite dans le pseudo-code 3. La fonction `transforme` réalise une transformée multirésolution (transformée en ondelettes ou transformée laplacienne par exemple). La fonction `calcul_champ` détermine un champ d'association à partir de la texture d'origine ou à partir de données fournies par l'utilisateur (voir les sous-sections suivantes).

Table 5: Pseudo-code pour l'algorithme générique de synthèse de texture.

Modélisation probabiliste. On a vu au paragraphe 6.3.4 que l'algorithme de transformée par groupements, lorsqu'il est appliqué sur une représentation multirésolution $\{\mathrm{A}_j\}_{j=0}^{\mathrm{J}-1}$ d'une texture A, correspond à la décomposition de A dans une frame $\{g^{\omega}_{j\ell m}\}_{j,\ell,m}$.

Le paramètre ω représente la géométrie, qui peut être représentée sous la forme d'un champ d'association, ou, comme on le verra à la sous-section 6.4.3, sous une forme plus compacte.

Dans cette section, on suppose que l'on dispose du paramètre géométrique $\widetilde{\omega}$ de la texture B à synthétiser. Ce paramètre peut être fourni par l'utilisateur (comme c'est le cas à la section 6.4.2) ou bien déterminé par une réalisation d'un processus aléatoire (comme c'est le cas à la section 6.4.3).

La texture synthétisée B s'obtient comme réalisation d'un vecteur aléatoire X tel que

$$\mathrm{X} = \sum_{j,\ell,m} \mathrm{X}_{j\ell m}\, \widetilde{g}^{\widetilde{\omega}}_{m}$$

où $\mathrm{X}_{j\ell m}$ sont des variables aléatoires identiquement distribuées pour chaque couple (j,ℓ) d'échelles. La loi de ces variables est donc supposée indépendante de la position $m \in \mathrm{I}_m$

$$\forall\, m \in \mathrm{I}_m, \quad p_{j\ell}(y) \stackrel{\text{def.}}{=} \mathrm{Prob}(\mathrm{X}_{j\ell m} = y),$$

ce qui correspond à une hypothèse de stationarité. On suppose de plus que les textures vérifient une hypothèse d'ergodicité, ce qui signifie que l'on peut estimer $p_{j\ell}$ à l'aide des histogrammes

$$\widetilde{p}_{j\ell}(y) \stackrel{\text{def.}}{=} \frac{1}{\mathrm{Card}(\mathrm{I}_m)} \mathrm{Card}\left\{ m \in \mathrm{I}_m \;\middle\backslash\; \left| \langle \mathrm{B},\, g^{\widetilde{\omega}}_{j\ell m} \rangle - y \right| \leqslant \varepsilon \right\},$$

où les y parcourent un ensemble fini de valeurs, qui sont les centres des boîtes de taille ε de l'histogrammes approché.

Algorithme de calcul. Pour synthétiser une réalisation B de ce processus, on peut utiliser la procédure d'égalisation d'histogrammes présentée à la section 6.2.2 et qui fait l'objet du pseudo-code 2.

Le pseudo-code 5 détaille les étapes de l'algorithme obtenu, qui sont semblables à celles de l'algorithme 2. La seule différence est la nécessité de calculer un champ d'association $\mathrm{F}_0, \ldots, \mathrm{F}_{\mathrm{L}-1}$ pour la texture d'origine A et un autre champ $\widetilde{\mathrm{F}}_0, \ldots, \widetilde{\mathrm{F}}_{\mathrm{L}-1}$ pour la texture B en cours de synthèse. Le champ d'origine est calculé à l'aide de l'algorithme `transfo_bandelettes` qui a été décrit à la section précédente et fait l'objet du code 3. Le champ de la texture à synthétiser est calculé une fois pour toutes par la fonction `calcul_champ`. Les sections qui suivent décrivent deux façons différentes de calculer ce champ.

6.4.2 Synthèse avec un flot géométrique fixé

Dans un premier temps, nous supposons que l'utilisateur fournit un champ de vecteurs très précis et nous déduisons un champ d'association multi-échelles $\widetilde{\omega}$. Ceci nous permet de synthétiser une texture qui « suit » ce flot de vecteurs et présente des caractéristiques proches d'une texture d'origine. Bien sûr, si la texture donnée en exemple possède une géométrie totalement différente du champ utilisé, cette méthode ne synthétisera pas une texture conforme à l'original.

Cette méthode fournit un algorithme pour générer une texture utile pour visualiser un champ de vecteurs. De nombreux algorithmes permettent une visualisation de champs de vecteurs, ils ont été présentés à la section 6.2.1. Bien que la comparaison avec notre algorithme sorte du cadre de cette thèse, on peut remarquer que notre méthode possède l'avantage d'utiliser une texture turbulente arbitraire comme modèle.

Calcul du champ d'association. Nous décrivons maintenant une implémentation possible de la fonction `calcul_champ` nécessaire au pseudo-code 5.

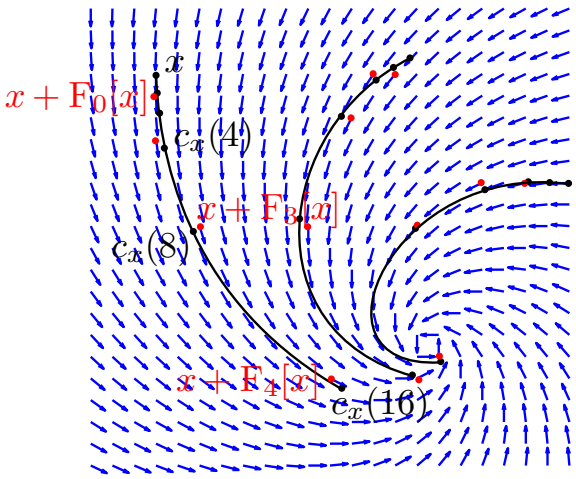

Fig. 6.14 *Calcul d'un champ d'association en suivant les lignes intégrales d'un flot.*

On suppose que l'utilisateur fournit un flot mono-résolution V de norme unitaire

$$\forall\, x \in [0,n]^2, \qquad \mathrm{V}(x) \in \mathbb{R}^2 \qquad \text{et} \qquad \|\mathrm{V}(x)\| = 1.$$

La plupart des données représentées sous forme de champs de vecteurs proviennent de simulation de mécanique des fluides. Dans la suite de cette sous-section, nous utilisons un modèle analytique simple de champ construit par superposition de champs potentiels centrés autour d'un point singulier, comme l'explique De Wijk [190]. Chaque champ élémentaire est décrit par son centre p_i, son terme source s_i et sa vorticité r_i. Le champ V est défini par

$$\mathrm{V}(x) = \sum_i \begin{pmatrix} s_i & -r_i \\ r_i & s_i \end{pmatrix} \frac{x - p_i}{\|x - p_i\|^2}.$$

La figure 6.15 montre deux champs de vecteurs

Ce flot mono-résolution permet de définir un champ d'appariement en suivant les lignes intégrales du flot

$$\forall\, x \in [0,n]^2, \quad \begin{cases} \dfrac{\partial c_x}{\partial t}(t) = \mathrm{V}(c_x(t)), \\ c_x(0) = x. \end{cases}$$

La courbe c_x est la ligne de flot issue de x, dont elle s'éloigne avec une vitesse unitaire constante. Ceci nous amène à définir, pour chaque échelle $\ell \geqslant 0$, le champ d'association

$$\forall\, x \in \{0,\ldots,n-1\}^2, \qquad \mathrm{F}_\ell[x] = x - [c_x(2^\ell)],$$

où $[a]$ représente le point à coordonnées entières le plus proche de a. La figure 6.14 montre un exemple de calcul de ce champ d'association.

Une fois que l'on dispose de ce champ d'appariement, on peut effectuer une synthèse de textures en utilisant l'algorithme déjà présenté au pseudo-code 5.

Résultats. La figure 6.16 montre les résultats de synthèses obtenus avec différents types de transformées multirésolution, qui implémentent la fonction `transforme` du pseudo-code 5. Nous avons testé :

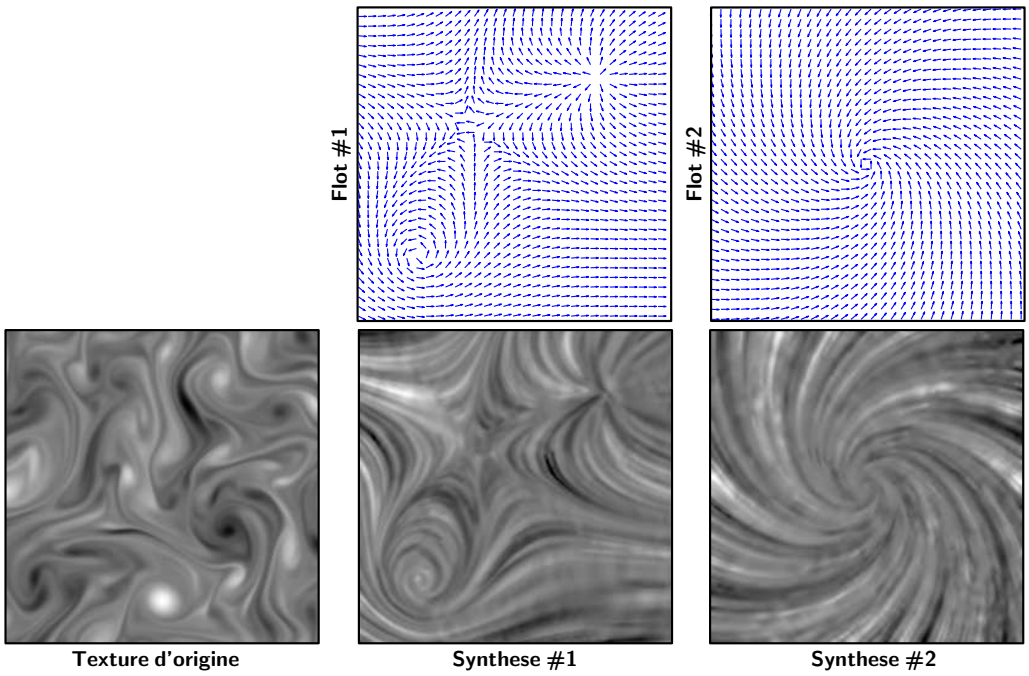

Fig. 6.15 *Exemples de champs de vecteurs synthétiques (champ en haut et lignes intégrales en bas) avec les synthèses de textures correspondantes.*

- en (b) aucune transformée n'est utilisée et on a $A_0 = A$, $m = 0$. Le résultat de synthèse n'est pas satisfaisant et montre bien qu'il est nécessaire de réaliser la transformée en bandelettes par groupements sur un domaine multirésolution. En effet, on retrouve bien les stries noires et blanches de la texture d'origine, mais la largeur de ces stries est beaucoup trop faible. Ceci est dû au fait que les bandelettes par groupements ne sont pas capables de capturer la régularité qui existe dans la direction orthogonale à la géométrie.
- en (c) une transformée en ondelettes invariante par translation est utilisée. Le résultat de synthèse n'est pas satisfaisant dans les zones où la géométrie est horizontale. Ceci est dû au fait que la texture originale ne contient pas de structure horizontale. Ce cas est certes extrême mais, généralement, le découplage des orientations empêche une bonne continuation des motifs passant de l'horizontale à la verticale.
- en (d) une pyramide laplacienne comme décrite par Simoncelli et al. [176] est utilisée. La figure 6.17 montre un exemple de décomposition à l'aide de cette pyramide laplacienne. Le résultat de synthèse est meilleur, parce que cette transformée est pratiquement invariante par translation et ne privilégie aucune orientation.

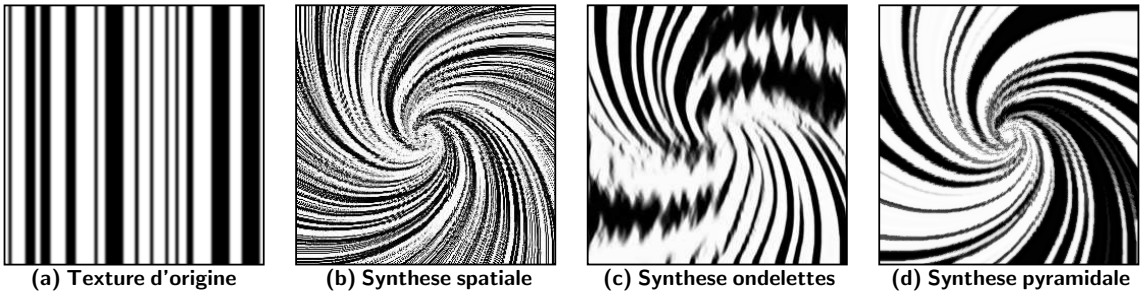

Fig. 6.16 *Comparaison des différentes transformées sur lesquelles on calcule le champ d'association.*

Fig. 6.17 *Transformée Laplacienne pyramidale.*

La figure 6.18 montre d'autres résultats de synthèse. On voit que pour la texture (b) la transformée en bandelettes a été capable de capturer une régularité longue distance, mais avec cependant quelques interruptions. Dans les deux cas, l'utilisation d'une transformée sur un domaine pyramidal multirésolution permet de capturer la largeur variable des stries, tout en autorisant un passage ininterrompu de l'horizontale à la verticale. La texture (c) présente une géométrie plus complexe à cause de l'enchevêtrement permanent des filaments verticaux et horizontaux. La synthèse en bandelettes n'est pas capable de représenter ce phénomène (car la texture synthétisée doit se conformer strictement au flot fourni par l'utilisateur). Cependant, la faible régularité de la texture d'origine se traduit par des interruptions fréquentes des structures géométriques synthétisées, ce qui donne un aspect assez proche de la texture d'origine.

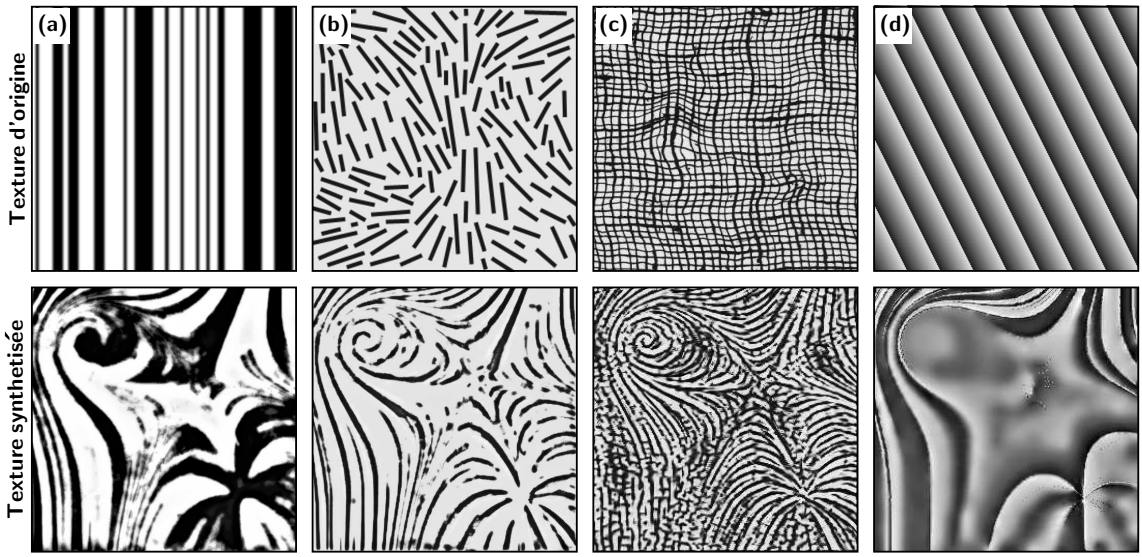

Fig. 6.18 *D'autres exemples de synthèses de textures avec flot géométrique fixé.*

L'algorithme de synthèse 5 est itératif et il est nécessaire d'effectuer une dizaine d'itérations pour obtenir une synthèse de qualité. La figure 6.19 montre l'évolution de la synthèse. Bien que l'on ne se préoccupe pas des questions de convergence de l'algorithme, on peut noter que le nombre d'itérations nécessaires est bien plus grand que pour l'algorithme simple de synthèse non-géométrique 2. Ceci est dû à la complexité des interactions entre les structures géométriques, qui peuvent s'amplifier ou bien s'anihiler au cours des itérations.

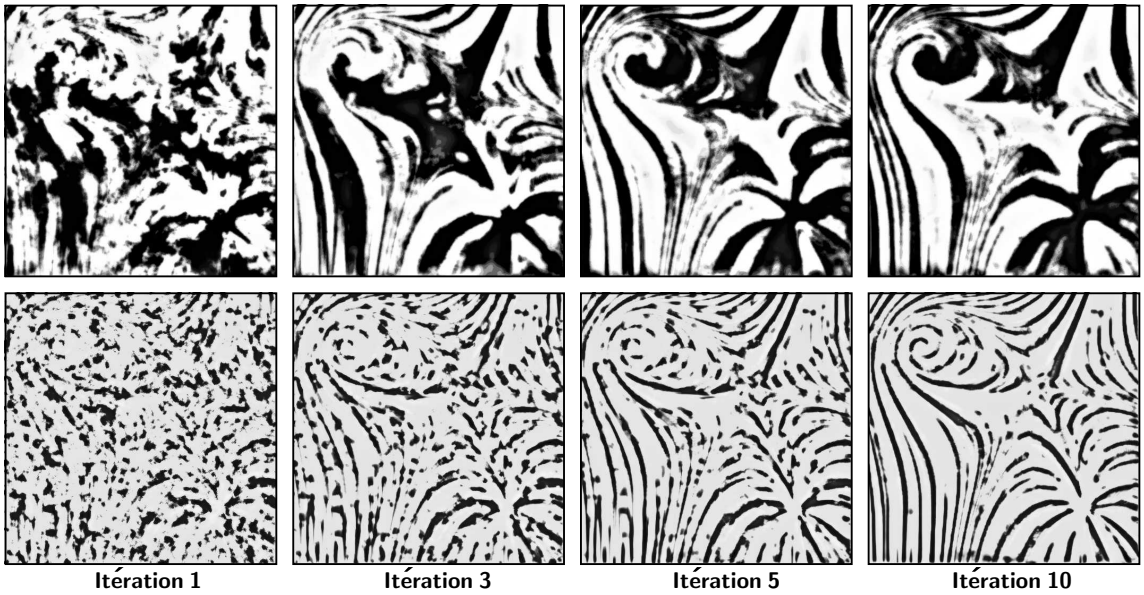

Fig. 6.19 *Progression de la synthèse de textures.*

6.4.3 SYNTHÈSE DE LA GÉOMÉTRIE

Nous souhaitons maintenant réellement synthétiser la géométrie $\widetilde{\omega}$ d'une nouvelle texture B. On utilise les champs d'association $\{\mathrm{F}_\ell\}_\ell$ de la texture d'origine A et construire des nouveaux champs d'association $\{\widetilde{\mathrm{F}}_\ell\}_\ell$ possédant des caractéristiques semblables. Ce nouveau champ sera calculé à l'aide de la réalisation d'un processus aléatoire dont la loi de probabilité est estimée à partir du champ $\{\mathrm{F}_\ell\}_\ell$.

Ceci fournira une nouvelle implémentation de la fonction `calcul_champ`(A) nécessaire au pseudo-code 5 qui permet de calculer $\widetilde{\omega}$. Cette nouvelle implémentation prend vraiment en compte le contenu de la texture A et n'utilise pas de flot mono-résolution fourni par l'utilisateur.

Représentation angulaire du champ d'association. Pour synthétiser les champs $\widetilde{\mathrm{F}}_\ell$, nous allons une fois de plus utiliser une procédure d'égalisation d'histogrammes. Comme on l'a vu à la section 6.2.2, il faut disposer de quantités fournissant une représentation creuse des champs d'association. Pour cela, nous commençons par effectuer un changement de variables permettant de ne manipuler que des orientations

$$\theta_\ell \stackrel{\text{def.}}{=} \operatorname{atan}((\mathrm{F}_\ell)_1/(\mathrm{F}_\ell)_2) \qquad \text{où} \qquad \mathrm{F}_\ell[x] = ((\mathrm{F}_\ell)_1[x], (\mathrm{F}_\ell)_2[x]).$$

Il est important de noter qu'il s'agit réellement d'une variable d'orientation, car le sens de la géométrie importe peu pour la synthèse. Ainsi les angles θ_ℓ peuvent être considérés modulo π. Ceci présente l'avantage d'enlever les chocs existants lorsqu'un champ d'association change brusquement d'orientation.

La figure 6.20 (ligne du haut) montre les champs angulaires $\cos(2\theta_\ell)$ pour une texture turbulente. On s'aperçoit que ces champs se ressemblent beaucoup et que, en un certain sens les angles se raffinent à travers les échelles. Il est donc naturel de ne pas synthétiser chaque θ_ℓ indépendamment, mais plutôt de synthétiser le différentiel existant entre θ_ℓ et $\theta_{\ell+1}$.

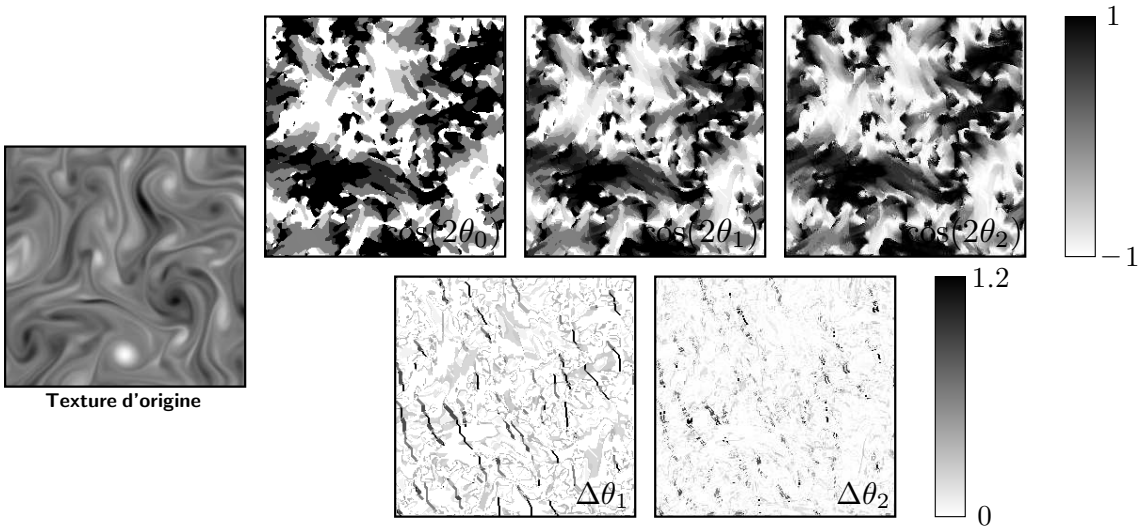

Fig. 6.20 *Cartes d'orientations θ_ℓ et d'orientations relatives $\Delta\theta_\ell$.*

Ainsi, on utilise le fait qu'il est possible, à partir d'un champ F_ℓ, d'extrapoler un champ $\mathrm{F}_\ell^\uparrow$ en composant deux fois le flot à l'échelle ℓ

$$\mathrm{F}_\ell^\uparrow[x] \stackrel{\text{def.}}{=} \mathrm{F}_\ell[x] + \varepsilon \mathrm{F}_\ell[x + \mathrm{F}_\ell[x]]$$

La quantité $\varepsilon \in \{+1, -1\}$ est calculée de façon à ce que $\mathrm{F}_\ell[x + \mathrm{F}_\ell[x]]$ et $\mathrm{F}_\ell[x]$ pointent dans le même sens. Ceci nous permet alors de définir un différentiel angulaire entre deux échelles

$$\theta_\ell^\uparrow \stackrel{\text{def.}}{=} \operatorname{atan}((\mathrm{F}_\ell^\uparrow)_1/(\mathrm{F}_\ell^\uparrow)_2) \qquad \text{et} \qquad \Delta_{\ell+1} = \theta_{\ell+1} - \theta_\ell^\uparrow.$$

La figure 6.21 illustre le principe du calcul de ces différentiels angulaires.

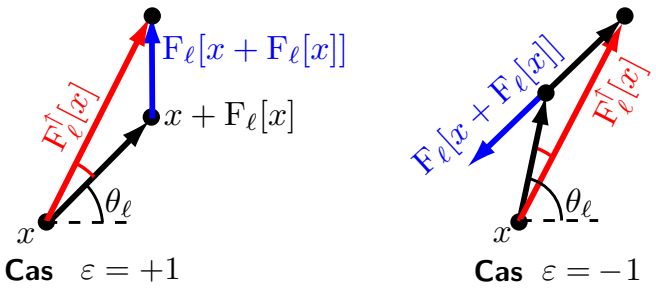

Fig. 6.21 *Calcul du champ extrapolé $\mathrm{F}_\ell^\uparrow$ et du différentiel angulaire $\Delta\theta_\ell$.*

Fonctions à valeurs circulaires et synthèse de directions. Il faut garder à l'esprit que θ_0 n'est pas une fonction à valeurs dans $[0, \pi]$, mais réellement une fonction à valeurs dans le cercle $\mathbb{R}/(\pi\mathbb{R})$. Traiter θ_0 comme une fonction à valeurs scalaires imposerait l'existence de singularités artificielles lorsque $\theta_0(x)$ passe de 0 à π. Ces singularités seraient difficiles à synthétiser, car elles sont difficiles à représenter en ondelettes.

La prise en compte de données à valeurs dans S^1, comme le fait par exemple Perona pour la diffusion [152], est l'un des nouveaux champs d'activité en vision par ordinateur. Chan et Chen expliquent ainsi l'importance des « images non plates » [37], qui sont à valeurs

dans un groupe de Lie (c'est à dire une surface possédant une structure algébrique). La régularisation des données d'IRM est un problème d'actualité qui met en jeu des images à valeurs tensorielles et nécessite des algorithmes adaptés comme celui de Chefd'hotel et al. [39].

Le traitement d'image n'a en revanche développé que peu d'algorithmes pour analyser ce type de signaux. Récemment, Ur-Uaman et al. [189] ont expliqué le formalisme de la transformée en ondelettes pour les fonctions à valeurs dans une variété.

Pour analyser F_0 et synthétiser $\widetilde{\mathrm{F}}_0$, nous avons utilisé une variante très simple de cette transformation, qui consiste à employer une transformée de Haar modifiée. On remplace les moyennes et les différences scalaires par des moyennes et des différences obtenues sur le cercle modulo π. Si l'on utilise une représentation des directions à l'aide de réels entre 0 et π, ceci consiste à remplacer l'étape élémentaire de Haar décrite à l'équation (6.2) par

$$(a,b) \rightarrow (m,d) \stackrel{\text{def.}}{=} \begin{cases} \frac{1}{2}(a+b, a-b) & \text{si} \quad |x-y| \leqslant \pi/2, \\ \frac{1}{2}(a+b+\pi, a-b+\pi)\,\%\pi & \text{si } |x-y| > \pi/2 \quad \text{et} \quad a \leqslant b. \\ \frac{1}{2}(a+b+\pi, a-b-\pi)\,\%\pi & \text{si } |x-y| > \pi/2 \quad \text{et} \quad a > b. \end{cases} \tag{6.3}$$

où $x\%\pi$ renvoie le représentant de x modulo π appartenant à $[0, \pi[$.

Paramétrisation de la géométrie. Contrairement à ce que l'on a fait à la section 6.4.2, on va maintenant paramétrer la géométrie $\widetilde{\omega}$ non plus directement par les champs $\{\widetilde{\mathrm{F}}_\ell\}_\ell$, mais par les variables angulaires

$$\widetilde{\omega} \stackrel{\text{def.}}{=} (\widetilde{\theta}_0, \Delta\widetilde{\theta}_1, \ldots \Delta\widetilde{\theta}_{\mathrm{L}-1}).$$

On peut en effet passer simplement de la représentation par variables angulaires à la représentation par champs d'association.

Il suffit d'appliquer un algorithme itératif, qui intialise $\widetilde{\mathrm{F}}_0$ à l'aide de $\widetilde{\theta}_0$ puis, pour chaque $\ell = 1, \ldots, \mathrm{L}-1$

- calcule le champ extrapolé $\widetilde{\mathrm{F}}^{\uparrow}_{\ell-1}$.
- cherche le point $y = x + \widetilde{\mathrm{F}}_\ell[x]$ au voisinage de $y_0 = x + \widetilde{\mathrm{F}}^{\uparrow}_{\ell-1}[x]$ tel que

$$\operatorname{atan}((y_1 - x_1)/(y_2 - x_2)) \quad \text{soit proche de} \quad \widetilde{\theta}_{\ell-1}[x] + \Delta\widetilde{\theta}_{\ell-1}[x].$$

- enregistre la valeur $\widetilde{\mathrm{F}}_\ell[x] = y - x$.

Modélisation probabiliste. Pour simplifier les notations, on définit

$$\widetilde{\omega} = (\widetilde{\omega}_0, \ldots, \widetilde{\omega}_{\mathrm{L}-1}) \stackrel{\text{def.}}{=} (\widetilde{\theta}_0, \Delta\widetilde{\theta}_1, \ldots \Delta\widetilde{\theta}_{\mathrm{L}-1})$$

les différentes composantes de la géométrie à synthétiser. On rappelle que $\widetilde{\omega}_0$ est une image à valeurs dans le cercle S^1 et que pour $\ell > 0$, $\widetilde{\omega}_\ell$ est à valeurs dans $\mathbb{R}$.

On calcule ainsi le paramètre $\widetilde{\omega}$ comme la réalisation d'un processus aléatoire

$$\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_{\mathrm{L}-1}),$$

dont la distribution de probabilité est estimée à l'aide de la géométrie de la texture d'origine.

Le paramètre $\widetilde{\omega}_\ell$ est donc une réalisation d'un vecteur aléatoire Ω_ℓ tel que

$$\Omega_\ell = \Phi_\ell^{-1}(\{\Omega_{\ell in}\}_{\ell,i,n})$$

où

$$\Phi_\ell : \left\{ \begin{array}{ll} & f \in \mathbb{R}^{\mathrm{N}} \\ \text{ou} & f \in (\mathrm{S}_1)^{\mathrm{N}} \end{array} \right\} \mapsto \{f_{in}\}_{i,n} \tag{6.4}$$

est une transformée en ondelettes à valeurs dans $\mathbb{R}$ ou dans S^1 (on a noté i l'indice d'échelle et n l'indice spatial). Pour $\ell = 0$, on utilise en effet une transformée en ondelettes de Haar circulaire définie à l'aide des étapes élémentaires 6.3. Pour $\ell > 1$, on utilise une transformée en ondelettes 2D classique.

Les coefficients en ondelettes $\{\Omega_{\ell in}\}_{\ell in}$ du processus Ω_ℓ sont alors supposés être des variables aléatoires identiquement distribuées pour chaque couple (ℓ, i) d'échelles. Comme on l'a déjà fait à la section 6.4.1, on peut estimer, pour chaque (ℓ, i) les densités de probabilité des coefficients $\{\Omega_{\ell in}\}_n$ à l'aide des histogrammes empiriques des coefficients $\Phi(\omega_\ell)$, où ω_ℓ est la géométrie de la texture A d'origine.

Algorithme de calcul. On peut donc générer des nouveaux paramètres angulaires $\widetilde{\omega}_\ell$ en utilisant une procédure d'égalisation d'histogrammes similaire à celle décrite par le pseudo-code 2.

Une fois ces paramètres synthétisés, on peut générer un champ d'association $\{\widetilde{\mathrm{F}}_\ell\}_\ell$. Ceci constitue une nouvelle implémentation de la fonction `calcul_champ`(A) du code générique 5.

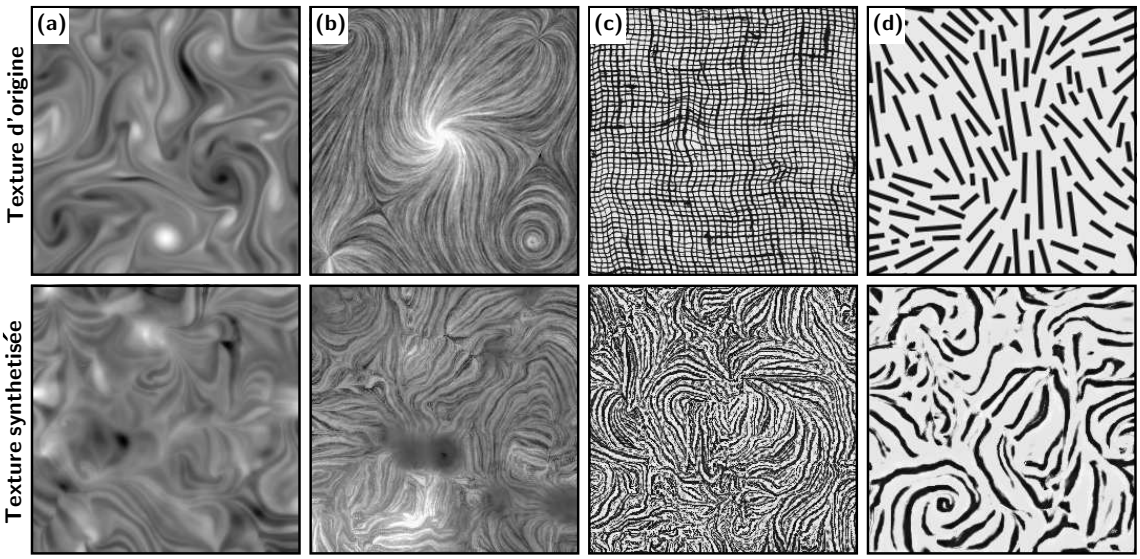

Fig. 6.22 *Synthèse de texture avec synthèse du champ d'association et des coefficients de bandelettes.*

Résultats. La figure 6.22 montre des résultats de synthèse de textures turbulentes. On observe que les textures synthétisées possèdent des structures géométriques relativement longues.

En (a) et (b), les structures géométriques sont un peu plus courtes que celles d'origine et la géométrie est globalement moins régulière. Ceci provient de la modélisation des champs

d'association par des coefficients d'ondelettes de différences angulaires qui n'est pas assez efficace.

En (c) la géométrie est totalement différente de la géométrie initiale (composée d'un enchevêtrement de courbes exclusivement horizontales et verticales). Ceci est dû au fait que notre transformée n'est pas capable de prendre en compte la juxtaposition de deux orientations (la géométrie doit être localement parallèle).

En (d), on observe une géométrie courbe (ou rectiligne sur des courtes distances), différente de la géométrie initiale. Les champs d'association sont en effet synthétisés indépendamment des coefficients de bandelettes. Ceci est problématique pour les textures possédant des zones sans géométrie.