HAL
archives-ouvertes.fr

Analyse / synthèse de champs de tenseurs de structure : application à la synthèse d'images et de volumes texturés

Adib Akl

To cite this version:

Adib Akl. Analyse / synthèse de champs de tenseurs de structure : application à la synthèse d'images et de volumes texturés. Synthèse d'image et réalité virtuelle [cs.GR]. Université de Bordeaux, 2016. Français. <NNT : 2016BORD0009>. <tel-01298560>

HAL Id: tel-01298560

https://tel.archives-ouvertes.fr/tel-01298560

Submitted on 6 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR

SPÉCIALITÉ : AUTOMATIQUE, PRODUCTIQUE, SIGNAL ET IMAGE, INGÉNIERIE COGNITIQUE

Par Adib AKL

ANALYSE/SYNTHÈSE DE CHAMPS DE TENSEURS DE STRUCTURE

Application à la synthèse d'images et de volumes texturés

Sous la direction de : M. Christian GERMAIN (Professeur des Universités)
co-directeur : M. Jean-Pierre DA COSTA (Maître de Conférences)
co-encadrant : M. Marc DONIAS (Maître de Conférences)
co-encadrant : M. Charles YAACOUB (Professeur Associé)

Soutenue le 11 Février 2016

M. Philippe BOLON	Professeur, Polytech Annecy-Chambéry	Président
Mme. Christine GUILLEMOT	Directrice de Recherche, INRIA	Rapporteur
Mme. Agnès DESOLNEUX	Directrice de Recherche, CNRS	Rapporteur
M. Bruno GALERNE	Maître de Conférences, Univ. Paris Descartes	Examinateur
M. Christian Germain	Professeur, Univ. de Bordeaux	Membre du Jury
M. Jean-Pierre DA COSTA	Maître de Conférences, Univ. de Bordeaux	Membre du Jury
M. Marc Donias	Maître de Conférences, Bordeaux INP	Membre du Jury
M. Charles YAACOUB	Professeur Associé, Univ. Saint-Esprit de Kaslik	Membre du Jury

ANALYSE/SYNTHÈSE DE CHAMPS DE TENSEURS DE STRUCTURE

Application à la synthèse d'images et de volumes texturés

Cette thèse s'inscrit dans le contexte de la synthèse d'images texturées. Dans l'objectif d'assurer une reproduction fidèle des motifs et des variations d'orientations d'une texture initiale, un algorithme de synthèse de texture à deux étapes « structure/texture » est proposé. Il s'agit, dans une première étape, de réaliser la synthèse d'une couche de structure caractérisant la géométrie de l'exemplaire et représentée par un champ de tenseurs de structure et, dans une deuxième étape, d'utiliser le champ de structure résultant pour contraindre la synthèse d'une couche de texture portant des variations plus locales. Une réduction du temps d'exécution est ensuite développée, fondée notamment sur l'utilisation de pyramides Gaussiennes et la parallélisation des calculs mis en œuvre.

Afin de démontrer la capacité de l'algorithme proposé à reproduire fidèlement l'aspect visuel des images texturées considérées, la méthode est testée sur une variété d'échantillons de texture et évaluée objectivement à l'aide de statistiques du 1er et du 2nd ordre du champ d'intensité et d'orientation. Les résultats obtenus sont de qualité supérieure ou équivalente à ceux obtenus par des algorithmes de la littérature. Un atout majeur de l'approche proposée est son aptitude à synthétiser des textures avec succès dans de nombreuses situations où les algorithmes existants ne parviennent pas à reproduire les motifs à grande échelle.

L'approche de synthèse structure/texture proposée est étendue à la synthèse de texture couleur. La synthèse de texture 3D est ensuite abordée et, finalement, une extension à la synthèse de texture de forme spécifiée par une texture imposée est mise en œuvre, montrant la capacité de l'approche à générer des textures de formes arbitraires en préservant les caractéristiques de la texture initiale.

Mots clés :

Synthèse de texture, tenseur de structure, texture couleur, texture volumique, champ de tenseurs contrainte, synthèse multi-échelle.

ANALYSIS/SYNTHESIS OF STRUCTURE TENSOR FIELDS

Application to the synthesis of textured images and volumes

This work is a part of the texture synthesis context. Aiming to ensure a faithful reproduction of the patterns and variations of orientations of the input texture, a two-stage structure/texture synthesis algorithm is proposed. It consists of synthesizing the structure layer showing the geometry of the exemplar and represented by the structure tensor field in the first stage, and using the resulting tensor field to constrain the synthesis of the texture layer holding more local variations, in the second stage. An acceleration method based on the use of Gaussian pyramids and parallel computing is then developed.

In order to demonstrate the ability of the proposed algorithm to faithfully reproduce the visual aspect of the considered textures, the method is tested on various texture samples and evaluated objectively using statistics of 1st and 2nd order of the intensity and orientation field. The obtained results are of better or equivalent quality than those obtained using the algorithms of the literature. A major advantage of the proposed approach is its capacity in successfully synthesizing textures in many situations where traditional algorithms fail to reproduce the large-scale patterns.

The structure/texture synthesis approach is extended to color texture synthesis. 3D texture synthesis is then addressed and finally, an extension to the synthesis of specified form textures using an imposed texture is carried out, showing the capacity of the approach in generating textures of arbitrary forms while preserving the input texture characteristics.

Keywords:

Texture synthesis, structure tensor, color texture, volumetric texture, constraint tensor field, multi-scale synthesis.

Remerciements

Les travaux présentés dans ce mémoire ont été menés au sein du groupe Signal et Image du laboratoire d'Intégration du Matériau au Système (IMS) que je remercie pour m'avoir offert un bon environnement de travail.

Le Professeur Germain a dirigé ma thèse. Je tiens à le remercier de m'avoir guidé tout au long de mon parcours, pour la cohérence qu'il a apporté à mes travaux de recherche, pour sa confiance en moi, pour son hospitalité lors de mes séjours à Bordeaux, et pour ses efforts à organiser la soutenance de thèse avec un jury d'experts.

J'adresse aussi mes remerciements les plus sincères à Dr. Jean-Pierre Da Costa, Dr. Marc Donias et Dr. Charles Yaacoub pour avoir encadré de façon continue mes activités de recherche, pour la patience qu'ils ont manifestée à mon égard et pour leur aide et leur soutien durant cette thèse.

Mes remerciements vont également aux rapporteurs, Prof. Christine Guillemot et Prof. Agnès Desolneux pour le temps qu'ils ont accordé à la lecture de ce mémoire et pour leur participation au jury de soutenance. Je remercie de même Prof. Philippe Bolon et Dr. Bruno Galerne de m'avoir fait l'honneur de participer au jury.

Je tiens à exprimer ma gratitude à la Faculté d'Ingénierie de l'Université Saint-Esprit de Kaslik (USEK). Je remercie particulièrement son Doyen, Ing. Pascal Damien, pour son aide et son soutien pendant ces trois ans. Je voudrais également remercier P. Marwan Azar pour sa bonté et pour la confiance qu'il m'a témoignée.

Je remercie vivement le Laboratoire des Composites Thermo Structuraux (LCTS) pour le soutien et les images, et Dr. Radu-Dragos Urs et Mlle Ioana Ilea pour les discussions techniques.

Mes travaux ont, pour partie, été menés dans le cadre du projet PyroMaN, financé par l'Agence Nationale de la Recherche Française (ANR) et Aerospace Valley, pôle de compétitivité mondial en Aéronautique, Espace et Systèmes Embarqués, associant les régions Midi-Pyrénées et Aquitaine. Ils ont également bénéficié d'un accompagnement financier du Centre Supérieur de la Recherche (CSR) de l'Université Saint-Esprit de Kaslik et

de l'Agence Universitaire de la Francophonie (AUF). Pour cela, je leur témoigne toute ma gratitude.

Je ne voudrais conclure ces remerciements sans mentionner mes collègues, mes amis, ma famille, mais plus particulièrement mes parents, pour leur bonté, leur encouragement et leur foi en moi. Sans leur soutien infini tout au long de mes longues années d'études, mon parcours n'aurait pu aboutir à cette thèse.

Table des matières

Remerciements ..... **iv**
Table de symboles et notations ..... **ix**
1 Introduction **1**
Eléments de contexte : 1
Objectif et contributions originales de la thèse : 2
2 Synthèse de texture **5**
2.1 Introduction 7
2.2 Notions fondamentales 7
2.2.1 Définition d'une texture 7
2.2.2 Caractéristiques d'une texture 9
2.2.3 Usage des textures en traitement d'image 11
2.3 Synthèse de texture 12
2.3.1 Principe de la synthèse de texture 12
2.3.2 Applications de la synthèse de texture 12
2.3.3 Typologie des méthodes de synthèse de textures 14
2.3.4 Synthèse à partir d'un exemple : état de l'art 17
2.3.4.1 Méthodes de synthèse 2D-2D 17
2.3.4.2 L'algorithme de Wei et Levoy 21
2.3.4.3 La synthèse 3D 23
2.4 Motivation 24
2.4.1 Limite des méthodes existantes 24
2.4.2 Approche de synthèse proposée : vue globale 25
2.5 Conclusion 25
3 Synthèse du champ de tenseurs de structure **27**
3.1 Introduction 29
3.2 Le tenseur de structure 30
3.3 Stratégie de synthèse du champ de tenseurs de structure 36
3.3.1 Principe 36
3.3.2 Métriques de comparaison de tenseurs 37
3.3.3 Système de voisinage 39
3.3.4 Approche multi-résolution 40
3.3.5 Accélération par interpolation 43
3.3.6 Traitement spécifique du champ de tenseurs d'entrée 45
3.3.7 Traitement des bords du champ de tenseurs de sortie 46

3.4 Résultats 46
3.4.1 Comparaison des métriques de l'espace tensoriel 46
3.4.1.1 Qualité des images synthétisées 46
3.4.1.2 Complexité calculatoire 49
3.4.2 Influence de la forme du voisinage, du type de parcours et du nombre d'itérations 50
3.4.3 Influence de la taille du voisinage 54
3.4.4 Réduction de la complexité calculatoire par interpolation 56
3.5 Conclusion 59
4 Approche de synthèse structure/texture 61
4.1 Introduction 63
4.2 Synthèse de texture contrainte par champ de tenseurs 63
4.2.1 Algorithme de synthèse 63
4.2.2 Synthèse multi-échelle 66
4.3 Résultats 68
4.3.1 Efficacité de la synthèse de la couche de structure et influence de la pondération structure/texture 68
4.3.2 Evaluation objective de la qualité de synthèse 70
4.3.3 Influence de la taille des voisinages de synthèse 71
4.3.4 Comparaison de la méthode proposée avec les approches existantes 73
4.4 Conclusion 83
5 Les extensions de l'algorithme de synthèse proposé 84
5.1 Introduction 86
5.2 Synthèse de texture couleur 86
5.2.1 Extension de l'approche structure/texture 87
5.2.2 Résultats 88
5.3 Synthèse de texture 3D 93
5.3.1 Synthèse de texture 3D à partir d'un exemplaire 3D 93
5.3.2 Résultats 95
5.3.3 Inférence 2D/3D du tenseur de structure 98
5.4 Synthèse de texture contrainte par champ de structure arbitraire 102
5.4.1 Construction du dictionnaire de recherche 103
5.4.2 Construction de la contrainte et processus de synthèse 104
5.4.3 Résultats 108
5.4.3.1 Pertinence du champ de tenseurs contrainte 108
5.4.3.2 Effet du pas angulaire du dictionnaire de recherche 110
5.4.3.3 Exemples de résultats 111
5.5 Conclusion 115

6 Conclusion **117**
Annexes **120**
Annexe A 120
Annexe B 122
Annexe C 124
Annexe D 126
Annexe E 127
Annexe F 128
Annexe G 129
Annexe H 130
Annexe I 135
Annexe J 137
Annexe K 139
Annexe L 141
Annexe M 142
Bibliographie **145**
Liste des publications de l'auteur **154**

Table de symboles et notations

Symbole	Signification
I	Image
S	Champ de tenseurs de structure
S_{xx} , S_{xy} , S_{yy}	Composantes d'un champ de tenseurs de structure S
$S(z)$, $I(z)$	Tenseur de structure et intensité au site z
$S_{xx}(z)$, $S_{xy}(z)$, $S_{yy}(z)$	Composantes d'un tenseur de structure $S(z)$ au site z
O , C	Champ d'orientation et champ de cohérence
$N_K(z)$	Ensemble de K sites connexes à z
$O(z)$, $C(z)$	Orientation et cohérence au site z
$VS(z)$, $VI(z)$	Voisinage de tenseur de structure et voisinage à valeurs d'intensité au site z
PS , PI	Pyramide de tenseurs de structure et pyramide de texture
PS^L , PI^L	Niveau L de la pyramide de tenseurs de structure et de la pyramide de texture
NL	Nombre d'échelles de la pyramide
$VS_L^{mr}(z)$, $VI_L^{mr}(z)$	Voisinages multi-résolution de tenseurs de structure et à valeurs d'intensité au site z du niveau L
$\nabla I = [I_x\ I_y]$	Champs de gradients horizontale et verticale
x , y	Coordonnées horizontale et verticale d'un site
$*$	Produit de convolution
D_x , D_y	Noyaux de dérivées Gaussiennes horizontale et verticale
σ_D	Ecart-type des noyaux de dérivées Gaussiennes D_x et D_y
$W = G_\sigma$	Noyau de lissage Gaussien
σ	Ecart-type du noyau de lissage Gaussien G_σ
G_{σ_L}	Noyau de lissage Gaussien utilisé pour construire une pyramide Gaussienne
σ_L	Ecart-type du noyau de lissage Gaussien G_{σ_L}
$2\downarrow$	Echantillonnage avec un facteur 2:1
$\lambda_1(z)$, $\lambda_2(z)$	Première et deuxième valeur propre d'un tenseur de structure $S(z)$
$e_1(z)$, $e_2(z)$	Premier et deuxième vecteur propre d'un tenseur de structure $S(z)$
$[\,]^t$	Transposée d'une matrice
det	Déterminant d'une matrice
I_d	Matrice identité
$trace$	Trace d'une matrice
$\widehat{[\,]}$	Logarithme d'une matrice
M_{SED}, $M_{SO,\alpha}$, M_{FN} , M_{LE}	Métriques de comparaison de tenseurs : Euclidienne, *Shape-Orientation*, Norme de Frobenius et Log-Euclidienne
SSTD	Somme des valeurs de dissimilarités des tenseurs de structure (*Sum of Structure Tensor Dissimilarity*)
SSD	Somme des erreurs quadratiques (*Sum Square Distance*)

$[\,]_{in}$, $[\,]_{out}$	Image, champ de tenseurs, pyramide d'image, pyramide de tenseurs, champ de cohérence, champ d'orientation, voisinage de tenseurs ou voisinage à valeurs d'intensité d'entrée et de sortie
W&L, P&S, E-L, Tar, Gal	Méthodes de Wei et Levoy, Portilla et Simoncelli, Efros et Leung, Tartavel et al., et Galerne et al.
γ	Métrique pour mesurer la ressemblance des voisinages des images de sortie et des images d'entrée
p	Facteur de pondération tenseur/intensité
Z_u	Formulation complexe de l'orientation
V_u	Champ aléatoire vectoriel
$\emptyset_z(w)$	Autocorrélation selon un déplacement w
$\emptyset_V(w)$	Formulation vectorielle de l'autocorrélation selon un déplacement w
$\|\,.\,\|$	Module d'un nombre complexe
$\langle\,.\,\rangle$	Moyenne
$\bar{S}^C$	Champ de tenseurs de structure couleur
$\bar{S}^C_{xx}$, $\bar{S}^C_{xy}$, $\bar{S}^C_{yy}$	Composantes d'un champ de tenseurs de structure couleur
S^R_{ab} , S^G_{ab} , S^B_{ab} ; $ab \in \{xx, xy, yy\}$	Composantes d'un champ de tenseurs de structure obtenues sur les trois canaux *R*, *G* et *B* d'une texture couleur
CSSD	Somme des erreurs quadratiques calculée sur des données couleur (*Color Sum of Squares Distance*)
γ^C	Métrique pour mesurer la ressemblance des voisinages des images de sortie et des images d'entrée couleur
VI^R , VI^G , VI^B	Voisinages à valeurs d'intensité dans les plans *R*, *G* et *B*
VS^{3D}	Voisinage de tenseurs de structure 3D
$[\,]^f$, $[\,]^d$, $[\,]^h$	Champs ou voisinages de tenseurs de structure de la vue de face, de droite et de haut
$D_M(I_{in})$	Dictionnaire de recherche de tenseurs de structure et à valeurs d'intensité
φ_m , M	Pas angulaire et nombre de rotation utilisés dans la procédure de construction du dictionnaire $D_M(I_{in})$
$V_k = (VI_k, VS_k)$	$k^{\text{ème}}$ voisinage à valeurs d'intensité et de tenseurs de structure dans le dictionnaire $D_M(I_{in})$
I_{arb} , S_{arb} , O_{arb}	Texture arbitraire, son champ de tenseurs de structure et son champ d'orientation
S_{con}	Champ de tenseurs de structure *contrainte*

O_{csg}	Champ d'orientation *consigne*
$D_M(S_{in})_{\mathrm{Agl}(z')}$	Champ de tenseurs du dictionnaire $D_M(I_{in})$ tourné selon l'angle $O_{csg}(z')$ imposé par la carte d'orientation consigne O_{csg}
$\mathrm{Agl}(z')$	Angle de recherche imposé par la carte d'orientation *consigne* O_{csg}
A_FS	Méthode d'utilisation du champ de tenseurs de la texture arbitraire comme contrainte à la synthèse de la texture
M_FS	Méthode d'instauration du champ de tenseurs *contrainte* par modulation et de synthèse de la texture
CT_LS	Méthode de synthèse du champ de tenseurs *contrainte* et de la texture
Z_C^L , T_C^L	Couronne (et sa largeur) inutilisée dans la procédure de recherche de la meilleure ressemblance à l'étage L de la pyramide
W^L, H^L	Largeur et hauteur de l'étage L d'une pyramide
W_u^L , H_u^L	Largeur et hauteur de l'étage L d'une pyramide utilisée dans la procédure de recherche de la meilleure ressemblance
D , G , $Liss$	Tailles des champs non utilisées dans la recherche de la meilleure ressemblance et dérivées par le calcul du gradient, par le lissage du tenseur et par le lissage Gaussien utilisé pour construire les étages supérieures d'une pyramide
[.]	Arrondi en entier par excès
S_{out_tor}	Champ de tenseurs périodique
$PS_{out_tor}^L$	Champ de tenseurs périodique à l'étage L de la pyramide PS_{out_tor}

Chapitre 1
Introduction

Eléments de contexte :

La notion de texture est au cœur de différentes méthodes de traitement et d'analyse d'images. Les applications impliquant les textures sont variées. Parmi elles, la synthèse de texture consiste à générer, à partir d'une texture modèle, un nouvel exemplaire de taille quelconque qui donne à un observateur une même impression visuelle.

La synthèse de texture constitue un champ de recherche actif. Elle joue un rôle important en traitement d'image, en infographie, et dans des domaines d'applications tels que le *mapping* de textures, l'extrapolation, la complétion d'images, la retouche ou la restauration d'images, l'animation et la synthèse de vidéos, la compression d'images ou la simulation de matériaux virtuels [Urs14].

Les différentes situations d'utilisation des textures [For03] ont conduit au développement de plusieurs algorithmes de synthèse paramétriques et non-paramétriques. Ces méthodes, généralement élaborées pour créer des images 2D, ont parfois été adaptées aux textures volumiques pour des applications, comme la reconstruction de carbones pyrolytiques observés par microscopie électronique [Ley09].

La plupart des méthodes existantes ont montré leur capacité à traiter avec succès une grande variété de textures, qu'elles soient stochastiques ou structurées. Cependant, dans des situations difficiles telles que les textures à grandes variations directionnelles, la plupart de ces approches ont tendance à produire des textures plus régulières que la texture modèle. Elles ont également des difficultés à synthétiser des textures exhibant des variations d'orientation non stationnaires. D'autres cas critiques concernent les textures avec des motifs multi-échelles de grandes dimensions que les approches classiques ne parviennent pas à bien reproduire. Ces différentes situations sont analysées dans [Urs12].

Il a été démontré dans [Pey09] que la synthèse préalable d'une couche géométrique peut aider à la synthèse de la couche de texture. Cette approche en deux étapes consiste à produire tout d'abord un champ d'orientation synthétique à partir du champ d'orientation de

l'exemplaire original, qui serait, dans un second temps, utilisé pour contraindre la synthèse de la texture elle-même, à l'aide des statistiques de niveaux de gris analysées à partir de l'exemplaire. L'algorithme de synthèse complet fonctionne très bien sur les textures turbulentes localement parallèles, mais ne parvient pas à reproduire les textures à géométries complexes avec des motifs croisés considérés comme des singularités du flux d'orientation.

Objectif et contributions originales de la thèse :

L'objectif de cette thèse est d'explorer les solutions algorithmiques pertinentes pour une synthèse assurant une reproduction aussi fidèle que possible des structures et des différentes variations d'orientations de la texture de départ. Un algorithme de synthèse à deux étapes structure/texture est proposé. Il consiste à produire la couche de structure dans une première étape, et de l'utiliser pour contraindre la synthèse de la texture, dans la deuxième étape.

La couche de structure est représentée par le champ de tenseurs de structure sans se limiter au flux d'orientation, mais en tenant compte à la fois de l'orientation locale, du degré d'anisotropie et de l'énergie [Kaa87]. L'étape de synthèse de la structure consiste à synthétiser le champ de tenseurs de structure. Le choix a été fait d'appuyer cette synthèse sur l'approche non-paramétrique de Wei et Levoy [Wei00], qui compte parmi les plus populaires, et qui a été conçue pour la synthèse de texture à partir d'un exemple. Sa mise en œuvre nécessite au préalable de l'adapter aux spécificités des valeurs tensorielles des images et de lui associer des métriques tensorielles adaptées.

La couche de structure synthétisée est utilisée dans une seconde étape pour contraindre la synthèse de la couche de texture. Pour cette seconde étape, l'algorithme de synthèse contrainte par champ de tenseurs de structure est aussi inspiré de l'algorithme de Wei et Levoy dont il sera une adaptation. Ainsi, les voisinages de synthèse combinent les voisinages à valeurs d'intensités appartenant aux images, et les voisinages de tenseurs de structure appartenant aux champs de tenseurs de structure avec un poids affecté à chacune de ces deux composantes.

Pour remédier aux difficultés liées à la complexité calculatoire, une synthèse multi-échelle basée sur la décomposition des images et des champs de tenseurs de structure en pyramides Gaussiennes est utilisée. Dans l'étape de synthèse du champ de tenseurs de structure, une méthode d'accélération qui consiste à remplacer la synthèse de l'étage de plus

grande taille de la pyramide par une interpolation bilinéaire, est proposée. Une accélération logicielle par parallélisation au moyen d'une carte graphique (GPU) a également été mise en œuvre.

Afin de s'assurer de l'intérêt de la méthode proposée, l'effet des différents paramètres de la synthèse du champ de tenseurs de structure et de la synthèse de la texture doit être étudié. Au-delà de l'évaluation visuelle des champs de tenseurs et textures de synthèse, une procédure d'évaluation quantitative est donc proposée. Cette étude permet, d'une part, d'identifier les stratégies les plus pertinentes pour la synthèse et, d'autre part, de comparer objectivement les algorithmes proposés à certaines approches paramétriques et non-paramétriques de la littérature.

Plusieurs extensions de la méthode proposées sont également développées. Une première extension de l'algorithme vise la synthèse de texture couleur. Elle s'appuie sur deux méthodes variantes de calcul du tenseur de structure d'une image couleur. La démarche est également étendue à la synthèse de textures volumiques. Enfin, la dernière variante de l'algorithme proposé consiste à contraindre la synthèse de la texture par un champ de tenseurs de structure arbitraire dans l'objectif de générer des textures de formes spécifiées. Cette dernière extension procède en deux étapes. Tout d'abord un champ de consigne d'orientation est construit à partir d'une image arbitraire ou par synthèse directe, puis est utilisé dans une seconde étape pour contraindre la synthèse de la texture.

Le mémoire s'organise de la manière suivante :

Le deuxième chapitre présentera les notions relatives aux textures, à leur synthèse et à ses applications. Il dressera l'état de l'art des méthodes de synthèse de texture 2D et de quelques extensions 3D, et détaillera l'algorithme de synthèse de Wei et Levoy. Les limites des approches de synthèse existantes seront discutées et une vue globale de l'algorithme de synthèse à deux étapes proposé sera finalement présentée.

Le troisième chapitre abordera la synthèse du champ de tenseurs de structure. Après une brève présentation de la notion de tenseur de structure, un algorithme non-paramétrique de synthèse sera proposé. Il s'appuie sur une mesure de similarité entre deux tenseurs dont le calcul repose sur quatre métriques différentes. La réduction du temps de calcul est fondée sur l'utilisation

de pyramides Gaussiennes et sur une interpolation bilinéaire. Une étude approfondie sur l'influence des différents paramètres de synthèse sera mise en œuvre.

Le quatrième chapitre porte sur l'approche de synthèse de texture contrainte par champ de structure. L'influence des paramètres de synthèse sera analysée et l'algorithme proposé sera évalué et comparé à cinq méthodes de synthèse de la littérature, de manière qualitative mais également quantitative, à l'aide de métriques objectives. Notons que dans les troisième et quatrième chapitres, les textures utilisées sont des textures en niveaux de gris.

Le cinquième chapitre sera consacré aux extensions de l'algorithme de synthèse : synthèse de texture couleur, synthèse de texture volumique et synthèse de texture de forme arbitraire.

Enfin, le chapitre six conclura ce manuscrit en dressant un bilan des contributions présentées et des résultats obtenus, et évoquera quelques perspectives quant à la poursuite des approches proposées.

Chapitre 2

Synthèse de texture

Sommaire

2.1 Introduction 7
2.2 Notions fondamentales 7
2.2.1 Définition d'une texture 7
2.2.2 Caractéristiques d'une texture 9
2.2.3 Usage des textures en traitement d'image 11
2.3 Synthèse de texture 12
2.3.1 Principe de la synthèse de texture 12
2.3.2 Applications de la synthèse de texture 12
2.3.3 Typologie des méthodes de synthèse de textures 14
2.3.4 Synthèse à partir d'un exemple : état de l'art 17
2.3.4.1 Méthodes de synthèse 2D-2D 17
2.3.4.2 L'algorithme de Wei et Levoy 21
2.3.4.3 La synthèse 3D 23
2.4 Motivation 24
2.4.1 Limite des méthodes existantes 24
2.4.2 Approche de synthèse proposée : vue globale 25
2.5 Conclusion 25

2.1 Introduction

La notion de texture est directement liée aux arrangements spatiaux des niveaux de gris ou couleurs des pixels d'une image. Elle est au cœur de nombreuses méthodes d'analyse et de traitement des images numériques. La diversité et l'efficacité des applications des textures n'ont plus besoin d'être démontrées. En particulier, la synthèse de texture, sujet de ce mémoire, représente un champ de recherche très actif car même si plusieurs approches de synthèse se sont montrées particulièrement performantes, il reste des cas critiques pour lesquels les algorithmes existants n'aboutissent pas à des résultats satisfaisants.

Dans un premier temps, ce chapitre dressera des notions fondamentales sur les textures. Les principes de la synthèse de texture et ses applications seront abordés, et une typologie des méthodes de synthèse sera proposée. Un état de l'art des nombreuses méthodes existantes viendra appuyer l'importance de la synthèse de texture en traitement d'image. Ceci conduira à la présentation des limites des méthodes existantes et ouvrira la voie vers la proposition d'un algorithme de synthèse original exploitant la couche de structure.

2.2 Notions fondamentales

2.2.1 Définition d'une texture

En traitement d'image, il est difficile de donner une définition universelle de la notion de texture du fait de la diversité des types de textures [For03] et de la variété des applications impliquant des textures.

Dans *Merriam-Webster's Collegiate Dictionary* [Merr], une texture est définie comme « une structure obtenue par l'entrecroisement des fils de tissage » et « l'aspect et les caractéristiques tactiles et visuelles d'un objet ». Il s'agit là de deux définitions distinctes, l'une spécifique à une application (le tissage) et ayant un caractère directionnel implicite et l'autre, uniquement perceptuelle.

D'un point de vue plus centré sur l'analyse d'images numériques, une texture peut être définie comme une fonction de la variation spatiale en niveaux de gris des pixels décrivant une large gamme de phénomènes naturels ayant des motifs répétitifs avec un caractère aléatoire [Con80] [Har92]. On notera que cette définition écarte les textures déterministes.

Dans [Har79], Haralick décrit les textures comme des phénomènes ayant deux composantes; les primitives (i.e. des éléments de base formant la texture) et l'organisation spatiale des motifs de la texture.

Unser [Uns84] propose une définition de la texture, inspirée des travaux de Gagalowicz [Gag81] : « Une texture est une région d'une image pour laquelle il est possible de définir une fenêtre de dimensions minimales, telle qu'une observation au travers de celle-ci se traduit par une perception (impression) visuelle identique pour toutes les translations possibles de cette fenêtre à l'intérieur de la région considérée ».

Germain [Ger97] propose une définition visant à prendre en compte le caractère déterministe ou stochastique des motifs et de l'arrangement, ainsi que le caractère multi-échelle et le principe de stationnarité. Autrement dit, une texture se veut être une combinaison hiérarchique de motifs déterministes ou stochastiques arrangés selon des règles elles-mêmes déterministes ou stochastiques. Observées au sein d'une fenêtre de taille limitée, elle fournit une impression visuelle similaire quelle que soit la position de cette fenêtre d'observation.
La définition devient alors : « Une texture, définie à une échelle donnée n, est une règle d'arrangement spatial des niveaux de gris des pixels qui, appliquée à une région R de taille supérieure à l'échelle n, se traduit par l'obtention de résultats identiques lors de la réalisation d'une expérience (visuelle ou mathématique) limitée à n'importe quelle fenêtre de taille n incluse dans la région R ». Cette définition se comprend aussi bien en 2D qu'en 3D.

Notons enfin qu'en infographie, une texture désigne souvent une image numérique plaquée à la surface d'un modèle maillé tridimensionnel [Hec86] afin de lui donner un aspect visuel plus réaliste.

Ces définitions d'une texture sont adaptées à un contexte sous-jacent. Certains auteurs considèrent qu'une texture ne peut être que locale (i.e. chaque pixel est prévisible à partir des quelques pixels de son voisinage) et stationnaire (i.e. ses statistiques spatiales sont invariantes par translation). Ainsi il a été démontré que de nombreuses textures peuvent être modélisées par un champ de Markov aléatoire [Pop93], qui lui-même leur associe les propriétés de localité et de stationnarité. D'autres auteurs définissent une texture comme une image comprenant des motifs répétitifs avec une variation aléatoire en niveaux de gris, sans tenir compte de la non-localité et/ou non-stationnarité [Har92].

Ce mémoire étant centré sur l'analyse/synthèse, une **texture** y désignera une combinaison de motifs déterministes ou stochastiques vue comme une variation spatiale de l'intensité ou, de façon plus générale de la couleur, peu importe son caractère local et/ou stationnaire. Par ailleurs, si plusieurs processus générateurs sont juxtaposés de façon à produire des régions perçues comme des textures, le résultat sera considéré comme une **image texturée**. La Figure 1 montre un exemple d'une texture locale (i.e. la couleur de chaque pixel ne dépend que de la couleur des pixels de son voisinage) et stationnaire (les différentes régions de la texture, surlignées en rouge, sont similaires et homogènes au sens de la structure et des couleurs), une texture de forme concentrique locale mais non-stationnaire, et une image non-stationnaire comprenant des régions localement texturées.

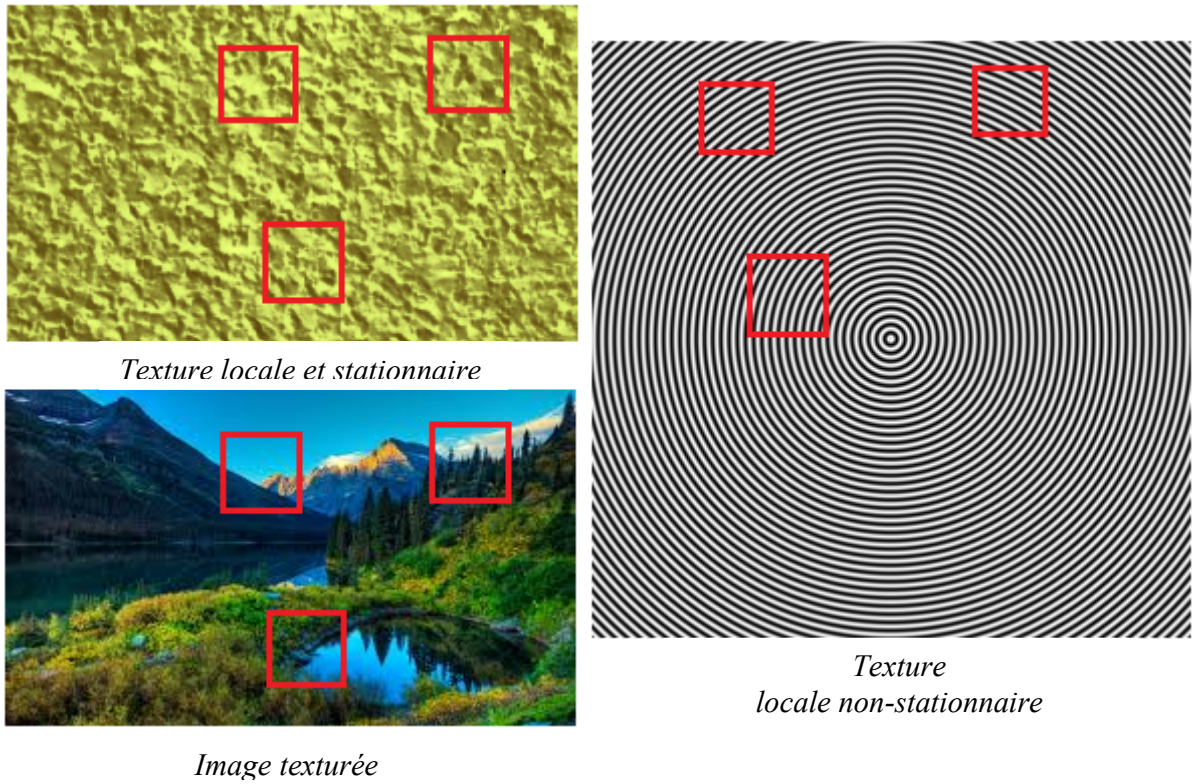


Figure 1 – *Exemple d'une texture locale (la couleur de chaque pixel ne dépend que de la couleur des pixels de son voisinage) et stationnaire (les régions sélectionnées en rouge sont perçues comme similaires), une texture locale et non-stationnaire et une image texturée dont les différentes régions produisent des impressions visuelles différentes.*

2.2.2 Caractéristiques d'une texture

Plutôt que de définir la notion de texture, Rao s'est attaché à leur description (e.g. granularité, directionnalité, anisotropie, etc.) et à leur classification [Rao90]. La Figure 2 illustre quelques exemples de textures. Les textures (a_1), (a_2) et (c_1) sont issues de l'ouvrage

de Brodatz [Bro66], (b_1) est une image sismique et (b_2) représente un matériau céramique [Mun08], (c_2) est un exemple d'une texture de matériaux carbonés servant de base à la simulation de matériaux virtuels [Ley09], (d_1) et (d_2) sont, respectivement, des images de sable et de bois. (a_1) peut être qualifiée comme granuleuse ou grossière tandis que (a_2) est directionnelle, régulière et périodique. (b_1) et (b_2) sont des textures laminaires localement anisotropes. (c_1) est une texture structurée présentant différentes échelles. (c_2), (d_1) et (d_2) illustrent des exemples de textures naturelles localement anisotropes.

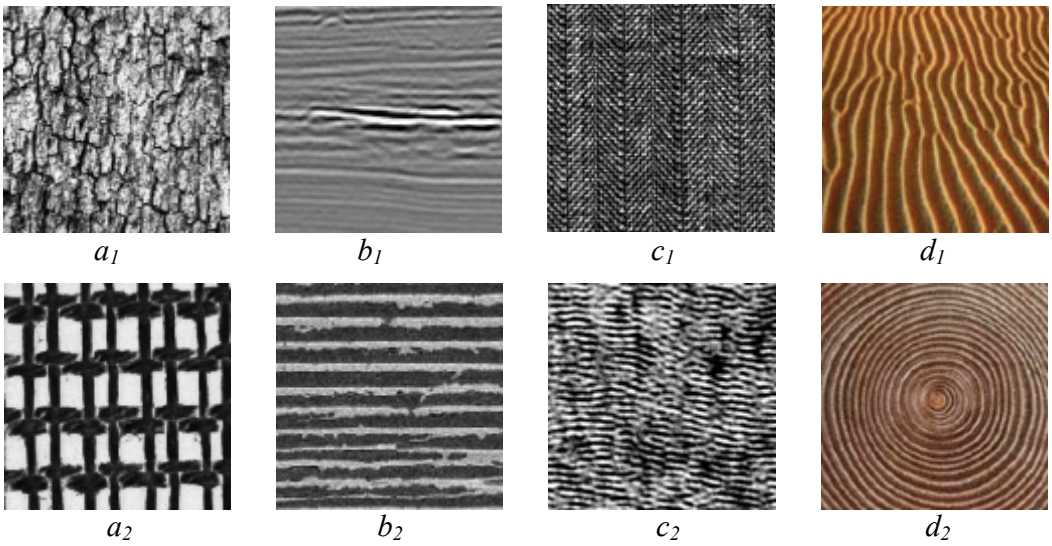
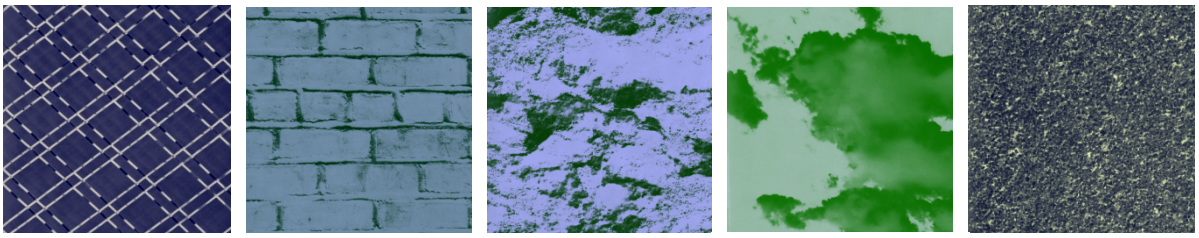


Figure 2 - *Exemples de textures. (a_1) : texture granuleuse et (a_2) : texture régulière directionnelle* [Bro66]. *(b_1) : donnée sismique et (b_2) : matériau céramique* [Mun08]. *(c_1) : texture présentant différentes échelles* [Bro66] *et (c_2) : matériaux carbonés* [Ley09]. *(d_1) et (d_2) : textures naturelles localement anisotropes.*

Les textures sont souvent décrites par leur degré de régularité [Lin04], allant des textures régulières aux textures stochastiques. La Figure 3 montre un exemple de cinq textures issues de la base de données couleur de Brodatz [Bro] et arrangées selon leur degré de régularité.

Les textures régulières désignent des motifs réguliers et structurés qui se répètent périodiquement, tandis que les textures stochastiques ressemblent à un bruit ou encore des points d'intensités différentes distribués aléatoirement sur toute l'image. Beaucoup de textures issues d'artefacts sont quasi-régulières i.e. des modifications de textures régulières à différents degrés (e.g. les tissus, les briques d'un mur, etc.). Les textures issues de la nature sont souvent plus aléatoires (fourrure, sable, marbre,…).



régulière *quasi-régulière* *irrégulière* *quasi-stochastique* *stochastique*

Figure 3 – *Exemple de textures* [Bro] *arrangées selon leur degré de régularité.*

2.2.3 Usage des textures en traitement d'image

De nombreuses applications en traitement d'image exploitent la notion de texture. Par exemple, la reconnaissance de formes, la classification et la segmentation des images [Che98]. Le rehaussement et la compression des images et la détection de défauts [Bee96] [Sun10] sont également des outils génériques s'appuyant sur la notion de texture.

Les approches relatives au traitement et à l'analyse des images texturées ont été déclinées dans différents champs d'applications, comme par exemple la sécurité ou le diagnostic en imagerie médicale (les empreintes digitales, la représentation de tissus organiques, etc.) [Dun00] et l'étude de la structure des différents tissus ou matériaux (bois, alliages, carbures, etc.). L'analyse de ces textures consiste à trouver des attributs descriptifs qui quantifient des notions comme la granularité, la directionnalité ou encore la finesse, qui sont utilisées pour qualifier les textures [Dac01]. Une autre application typique consiste à retrouver des régions similaires ou bien des régions d'intérêt dans les images sismiques ou les images de télédétection (nuages, scènes agricoles ou environnementales) [Skl78] [Wik01]. D'autres applications graphiques utilisent des textures couleur représentant des images du monde réel observées sous différentes conditions d'éclairage [Bar06]. Tous ces exemples ont montré depuis plusieurs décennies, la pertinence des textures en analyse d'image dans des contextes variés.

En outre, les textures naturelles peuvent être modélisées et reproduites de manière efficace en utilisant les techniques de synthèse d'image [Har92]. La synthèse de texture, quant à elle, peut être vue comme un moyen empirique de tester la généralité et la pertinence des modèles de texture. Par rapport aux approches visant la classification ou la segmentation de textures, la synthèse constitue parfois un défi encore plus grand car elle nécessite une description plus détaillée et très fidèle (d'un point de vue perceptuel) de la texture.

2.3 Synthèse de texture

2.3.1 Principe de la synthèse de texture

Depuis une quinzaine d'années, la synthèse de texture a été un sujet de recherche actif en vision par ordinateur. La synthèse d'une texture consiste ici à remplir une région plane ou un volume à partir de la texture observée sur un échantillon modèle. Lorsque l'échantillon est trop petit pour recouvrir la région considérée, il doit être étendu. Si l'on essaie d'obtenir une nouvelle image, plus grande que l'originale, en copiant la texture de départ, il est très probable que l'image de sortie aura un aspect artificiel de pavage régulier. C'est ici qu'intervient la synthèse de texture.

L'idée est alors de créer une texture de taille quelconque et qui ressemble à l'échantillon original. La texture créée ou *texture de sortie* devrait idéalement être perçue comme une partie d'un matériau homogène duquel aurait également été extraite la texture d'entrée. Elle ne doit contenir ni répétitions non souhaitées ni de défauts visibles. D'un point de vue perceptuel, la texture synthétisée doit sembler avoir été créée par le même processus que celui ayant engendré l'échantillon d'entrée [Wei00] comme illustré à la Figure 4.

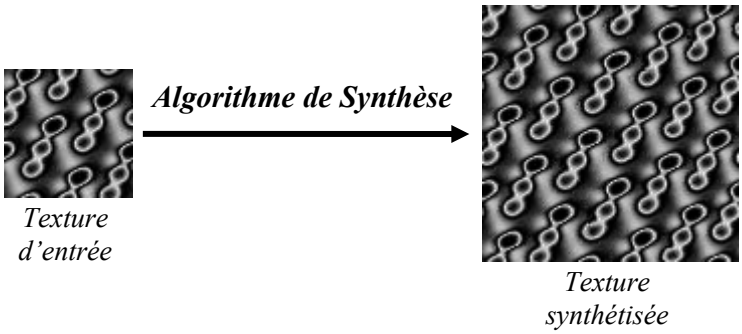


Figure 4 - *Illustration du processus de synthèse de texture : à partir d'un exemplaire (à gauche), il s'agit de synthétiser une nouvelle texture (à droite) perçue comme ayant été créée par le même processus que celui de l'exemplaire.*

2.3.2 Applications de la synthèse de texture

Des applications classiques de la synthèse de texture sont par exemple :

- Le *mapping* de texture [Hec86] :

En infographie, la synthèse de texture est une technique courante pour créer généralement de grandes textures à partir de petits échantillons. Les textures de synthèse ainsi créées peuvent alors être utilisées pour « habiller » la surface d'objets bi ou tridimensionnels.

- La retouche, la restauration ou la complétion d'images (*image inpainting*) [Ige97] [Efr99] [Ber00] [Yam03] :

Au fil du temps ou en raison d'une mauvaise manipulation, les photographies ou films se détériorent (rayures, taches, etc.). Ils peuvent également nécessiter des modifications liées à la prise de vue (occultation par des objets, présence d'objets ou personnes indésirables, etc.). Ces différents cas peuvent être corrigés par synthèse de texture (Figure 5).

a_1 a_2 b_1 b_2

Figure 5 – *Exemples de retouche d'images* [Ige97]. *(a_1) et (b_1) sont les images originales présentant des objets indésirables, (a_2) et (b_2) sont les images corrigées par synthèse.*

- Animation et synthèse vidéo [Szu96] [Sch00] [Kwa03] [Dub12] [Köp12] :

La notion de texture, concerne le plus souvent des images 2D, mais peut être adaptée aux structures volumiques telles que celles obtenues par des techniques d'imagerie 3D (e.g. imagerie médicale, tomographie, imagerie sismique, etc.). L'extension à 3 dimensions, nativement dédiée à la synthèse de textures volumiques, peut également concerner les textures temporelles (2D+T) ayant une étendue indéterminée à la fois dans l'espace et dans le temps, i.e. des régions homogènes sans réelles limites dont les arrangements spatiaux évoluent dans le temps, représentant, par exemple, du feu, de la fumée ou des vagues [Wei00]. D'autres exemples de textures temporelles concernent les animations générées par ordinateur qui peuvent représenter des mouvements d'arrière-plan plus ou moins cycliques (houle, chutes d'eau, mouvements de feuillages). La notion de texture temporelle permet ainsi d'étendre les outils de retouche à la synthèse vidéo.

- La compression d'images [Bee96] [Sun10] :

Les images montrant des scènes naturelles contiennent souvent de grandes régions texturées. Les algorithmes standards de compression d'image peuvent parfois détériorer ces textures. Une étape préalable de segmentation et de modélisation des textures présentes dans l'image,

permet d'obtenir une représentation compacte de ces textures qui peut, à son tour, être utilisée pour reproduire les régions texturées lors de la phase ultérieure de décompression des images.

- Simulation de matériaux composites [Ley09] [Cha14] :

La synthèse de matériaux virtuels joue un rôle croissant en sciences des matériaux, tout particulièrement pour les matériaux composites. La première étape de la conception d'un matériau virtuel s'appuie de plus en souvent sur des techniques d'imagerie industrielle pour la caractérisation initiale de la structure microscopique ou nanométrique du matériau d'intérêt. Ensuite, la synthèse du matériau peut faire appel à des techniques de synthèse de texture pour générer *in silico* de nouveaux matériaux aux caractéristiques désirées.
Dans certains cas toutefois, l'utilisation de techniques d'imagerie 3D n'est pas envisageable [Wei01]. Par exemple, les structures tridimensionnelles constitutives des matériaux sont souvent imagées en 2D par microscopie optique ou électronique à des résolutions spatiales que les tomographes n'atteignent pas en 3D. Dans ces conditions, toute méthode stéréologique permettant l'extrapolation en 3D d'une information 2D est susceptible d'améliorer la compréhension de la réalité physique du matériau. En d'autres termes, une approche de synthèse volumique par inférence 2D/3D est parfois la seule possibilité d'accéder à une structure tridimensionnelle probable de ce matériau [Dac10].

2.3.3 Typologie des méthodes de synthèse de textures

Les travaux menés en synthèse de texture au cours des deux dernières décennies ont conduit à la mise au point de nombreuses techniques dont il est possible de dresser une typologie en trois grandes familles: les méthodes procédurales, les approches de synthèse à partir d'un exemple et les méthodes de synthèse sur la base d'un modèle.

Il est important de noter que les frontières de cette classification sont perméables puisque certaines approches peuvent tout à fait être à cheval sur différentes catégories. Par exemple, on peut synthétiser à partir d'un exemple, en utilisant une approche procédurale, sur la base d'une analyse fondée sur un modèle (e.g. synthèse par bruits de Gabor [Lag09] [Gal11 bis]).

- Les méthodes procédurales :

Les méthodes de synthèse de textures procédurales consistent à produire des textures à l'aide de fonctions mathématiques ou bien d'algorithmes qui peuvent être exécutés avec un coût calculatoire fixe. Il s'agit donc d'outils très bien adaptés pour générer la texture d'objets dans des environnements virtuels, comme ceux des jeux vidéo par exemple [Ebe02]. Différents

auteurs ont essayé de reproduire des phénomènes du monde réel, tels qu'un flux d'eau, la corrosion ou la distribution des particules dans les matériaux, en utilisant des propriétés physiques ou chimiques des procédés observés [Urs13]. Ces approches sont basées sur la transformation de certains signaux prédéfinis en une texture désirée. Elles sont généralement utilisées pour la synthèse de textures très structurées (e.g. un pavé) ou à l'inverse pour produire certains types de textures complètement non structurées, telles que le bruit de Worley [Wor96]. D'autres fonctions de bruit ont été proposées, telles que les bruits de Gabor [Lag09] [Gal11 bis].

- Les approches de synthèse à partir d'un exemple :

Ces approches nécessitent une ou plusieurs textures exemples comme entrées. La plupart d'entre elles synthétisent les textures par copie directe des pixels ou de *patches* (sous-images) de l'image d'entrée. Ces approches consistent donc à générer une nouvelle texture qui ressemble le plus possible au modèle ou exemple (l'exemplaire). Elles peuvent synthétiser une très grande variété de textures et comportent trois familles principales :

- Synthèse de texture par approches paramétriques ou par analyse [Hee95] [DeB97] [Por00] [Rab10] :

Cette famille d'approches est basée sur la modélisation statistique de la texture exemple. Une nouvelle texture est donc synthétisée de sorte qu'un ensemble de contraintes statistiques (signature texturale ou paramètres) estimées à partir de l'image d'entrée soit imposé sur la texture de sortie. Le principe de synthèse de texture par analyse est résumé à la Figure 6. Les caractéristiques de l'échantillon d'entrée sont extraites pour obtenir les différents paramètres utilisés dans le processus de synthèse. La texture synthétique est ensuite comparée à l'exemplaire pour s'assurer que les deux textures ont un aspect similaire.

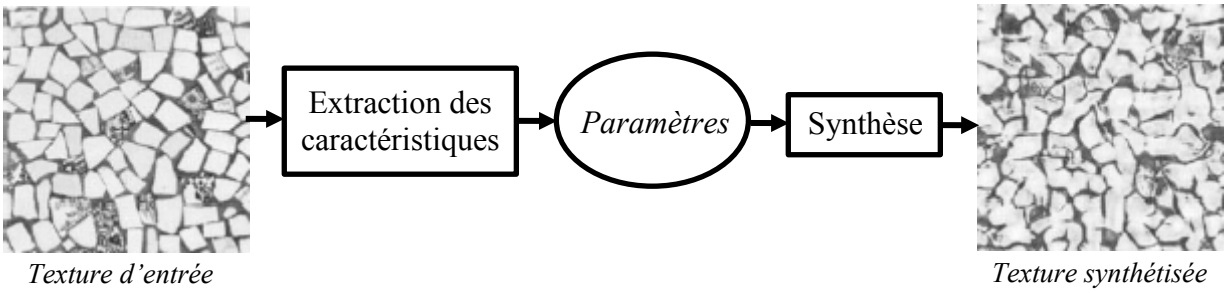


Figure 6 – *Principe des méthodes de synthèse de texture par analyse* [Por00].

- Synthèse de texture pixel par pixel [Pop93] [Pag98] [Efr99] [Wei00] [Ash01] [Her01] [Zel02] [Ton02] :

Ces méthodes de synthèse génèrent la texture pixel par pixel, en maintenant la cohérence de la texture locale avec son voisinage. Elles sont généralement basées sur la théorie des champs de Markov qui modélise la texture comme une réalisation d'un processus local et stationnaire. Pour assurer un coût calculatoire raisonnable, la plupart de ces approches impliquent des algorithmes de recherche efficaces combinés à une mise en œuvre multi-échelle capable d'exprimer des modèles à différentes échelles, sans augmenter significativement la charge de calcul.

- Synthèse de texture par *patch* [XuG00] [Efr01] :

La synthèse de texture par *patch* est apparue comme une amélioration des approches à base de pixels en termes de complexité, car ces dernières méthodes souffrent d'un coût calculatoire élevé lorsqu'il s'agit de reproduire des images très structurées. L'approche générale consiste à choisir le *patch* (morceau) de l'exemplaire le plus ressemblant au voisinage en cours de synthèse dans la texture de sortie, et à « rabouter » les *patchs* choisis entre eux à l'aide de techniques d'optimisation pour réduire les artefacts ou défauts de contour.

Notons que la catégorie de synthèse à partir d'un exemple englobe la plupart des techniques de synthèse.

- Les méthodes de synthèse sur la base d'un modèle :

Ces méthodes consistent à construire un modèle d'image qui peut être utilisé pour décrire la texture, mais aussi à en faire la synthèse. Les paramètres du modèle doivent capturer les caractéristiques visuelles importantes de la texture. De nombreux modèles existent dans la littérature, tels que [Cro83] [Gem84] [Che85] [Tur86] [Cla87] [Pei88] [Fra93] [Che93]. Notons que le modèle peut être obtenu par apprentissage à partir d'une image exemple, ce qui rapproche alors cette catégorie de méthodes de la précédente.

Dans les paragraphes suivants, nous nous concentrerons sur les approches de synthèse à partir d'un exemple dont nous dresserons un état de l'art et sur lesquelles nous nous appuierons, dans les chapitres 3 à 5, pour proposer de nouveaux algorithmes.

2.3.4 Synthèse à partir d'un exemple : état de l'art

2.3.4.1 Méthodes de synthèse 2D-2D

- Les méthodes non-paramétriques :

Dans les dernières décennies, plusieurs algorithmes non-paramétriques ont été proposés pour l'analyse et la synthèse de texture. En particulier, les approches basées sur les champs de Markov aléatoires et les approches par *patch* ont prouvé leur efficacité en synthèse de textures structurées.

Une des premières méthodes, classée comme un algorithme de synthèse non-paramétrique par champ de Markov, consiste à ordonner les pixels selon un parcours particulier, par exemple lexicographique, puis à synthétiser chaque pixel à partir d'une représentation non-paramétrique de la fonction de probabilité conditionnelle. Cette fonction de probabilité est estimée à partir d'échantillons de la texture d'entrée [Pop93], en utilisant un modèle de mélange Gaussien (*Gaussian Mixture Model*). Les principaux problèmes posés sont la taille limitée de l'ordre du voisinage et la causalité artificielle induite par l'existence d'un parcours prédéfini. Des améliorations ont été apportées à l'aide de l'estimateur de densité non-paramétrique de type «fenêtre de Parzen» [Sil86] [Pag98]. Cette approche est capable de synthétiser des textures complexes et structurées en modélisant la texture par un champ de Markov aléatoire. La méthode capture les caractéristiques visuelles de la texture dans un modèle statistique unique qui décrit les interactions entre les valeurs des pixels.

La synthèse de texture par échantillonnage non-paramétrique proposée par Efros et Leung [Efr99] [Agu13] ne nécessite aucun estimateur de densité de probabilité, mais utilise simplement un algorithme de recherche du plus proche voisin pour générer une texture en échantillonnant directement de nouvelles valeurs à partir de l'échantillon d'entrée. Cette méthode est efficace dans la synthèse de textures très structurées. La Figure 7 illustre l'approche non-paramétrique d'Efros et Leung [Efr99]. La texture est synthétisée pixel par pixel en identifiant tous les voisinages causaux dans l'échantillon d'entrée (les carrés à gauche) qui sont similaires au voisinage du pixel en cours de synthèse (carré à droite) et en choisissant ensuite, aléatoirement, un de ces voisinages dont le pixel central est recopié.

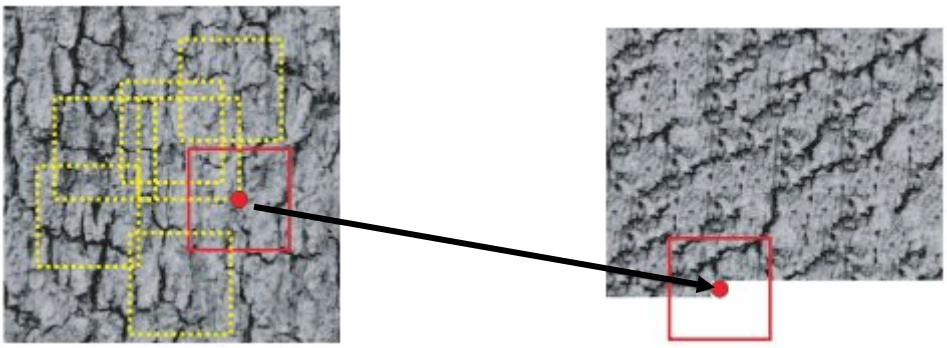


Figure 7 – *Principe de synthèse d'Efros-Leung* [Efr99] *: un des voisinages de l'exemplaire, qui sont similaires au voisinage du pixel courant, est choisi aléatoirement et son pixel central est copié à la position cible en sortie.*

L'algorithme d'Efros et Leung qui n'est pas multi-échelle a été le précurseur de la méthode proposée par Wei et Levoy [Wei00] qui est l'une des approches les plus polyvalentes. Celle-ci repose également sur les hypothèses d'un champ de Markov aléatoire et synthétise la texture de sortie pixel par pixel selon un ordre de balayage prédéfini. Etant à la base de l'algorithme de synthèse que nous allons proposer ultérieurement, l'algorithme de Wei et Levoy [Wei00] sera plus particulièrement détaillé dans le paragraphe suivant.

Pour accélérer la recherche du meilleur voisinage, un principe efficace consiste à réduire, pour chaque pixel à synthétiser, le nombre de voisinages à comparer [Ash01]. Ces voisinages sont définis comme les voisinages de la texture échantillon qui ont déjà été retenus pour la synthèse des pixels voisins au pixel auquel on s'intéresse dans la texture de sortie.

Un autre type de méthodes utilisant des analogies est proposé dans [Her01]. Cet algorithme combine les idées de [Wei00] et [Ash01] et les étend au cas de paires d'images. D'autres approches ont pour but d'éviter les comparaisons aux plus proches voisins qui sont couteuses en termes de complexité calculatoire [Zel02].

Dans le but de réduire la complexité calculatoire, [Ton02] a introduit une méthode basée sur la *k*-cohérence. Cette méthode est capable de synthétiser des fonctions de texture qui décrivent l'apparence d'une surface du monde réel en fonction de l'éclairage et des directions d'observation. Le principe consiste à identifier préalablement, pour chaque pixel de la texture d'entrée, l'ensemble des *k* plus proches correspondances dans la texture d'entrée, i.e. l'ensemble des *k* pixels dont les voisinages sont les plus ressemblants au voisinage du pixel considéré, au sens d'une distance entre voisinages. Lors de la phase de synthèse, les pixels voisins du pixel d'intérêt sont utilisés pour définir un ensemble de candidats à partir duquel la

valeur du meilleur pixel est déterminée et copiée. La valeur de *k* dépend du type de texture à synthétiser : une valeur élevée est généralement utilisée tandis qu'une valeur faible est préférable pour les textures comportant des composantes hautes fréquences.

Une des premières approches de synthèse de texture par *patch* consiste à sélectionner, dans l'image d'entrée, des *patches* ressemblants, au sens d'une certaine mesure, à des *patches* environnants dans l'image de sortie, puis à réduire les artefacts ou défauts de contour [Efr01]. La synthèse de texture à base de *patch* a également été réalisée sous la forme de mosaïque de chaos [XuG00]. Ceci consiste en une distribution aléatoire des patches de la texture d'entrée sur la sortie et un lissage des frontières entre *patches* qui se chevauchent. L'inconvénient est que la synthèse produit de grandes régions où la texture d'entrée est copiée sans aucune variation à partir de l'exemple, ce qui n'est pas visuellement satisfaisant. Un algorithme de recherche rapide basé sur une structure pyramidale par arbres quaternaires (*quad-tree*) de la texture d'entrée a été introduit. Cela garantit une technique d'ajustement séquentiel optimale en fixant un *patch* après l'autre [Lia01]. Une version améliorée a été proposée par une technique de coupure de graphe appelé *min-cut* ou *max-flow* [Kwa03], permettant les réévaluations des coupures précédentes par rapport à d'autres.

- Les méthodes paramétriques :

Les approches paramétriques constituent une autre famille d'algorithmes de synthèse de texture. La première approche paramétrique remarquable a été proposée par Heeger et Bergen [Hee95]. Cette méthode contraint la synthèse en utilisant les histogrammes des réponses des filtres à différentes échelles et orientations d'une pyramide orientée. C'est l'une des premières méthodes de synthèse de textures couleurs. L'approche est itérative, procédant tour à tour à une décomposition de l'image en une collection de sous-bandes fréquentielles, à un ajustement d'histogramme des sous-bandes et à une reconstruction de l'image. La méthode proposée dans [DeB97] est plus performante que le processus itératif. Elle consiste à modifier l'entrée de sorte à restreindre la distribution conditionnelle de la sortie du filtre sur de multiples résolutions.

Une amélioration substantielle a été réalisée par Portilla et Simoncelli [Por00] en utilisant la transformée en ondelettes pour paramétrer le modèle par un ensemble de descripteurs statistiques calculés à différentes orientations et échelles. Cette méthode étend la modélisation statistique au second ordre et réalise la mise à jour de la réponse du filtre, non pas par égalisation d'histogramme, mais par un algorithme de descente de gradient. Plusieurs

améliorations de cette méthode ont été proposées [Rab10]. Cependant, ces types d'approches conduisent à des formulations complexes qui sont difficiles à optimiser.

Les méthodes stochastiques ont également démontré leur capacité à donner des résultats impressionnants avec les micro-textures stochastiques. Par exemple, le *spot noise*, ou encore le *shot-noise* bidimensionnel a été introduit par van Wijk dans [Wij91] pour synthétiser de nouvelles textures à partir des images de *spot* générées en assemblant un grand nombre de fonctions d'intensité à des positions aléatoires dans le plan. Dans cette approche, l'asymptote du *spot noise* discret (*Discrete Spot Noise - DSN*) est obtenue par blanchiment des phases de tous les coefficients de la transformée de Fourier. Galerne et al. [Gal11] proposent un algorithme de bruit de phase aléatoire (*Random Phase Noise - RPN*) basé sur un simple tirage aléatoire de la phase de la transformée de Fourier. Cet algorithme est capable de reproduire des textures qui sont caractérisées par le module de leur transformée de Fourier, plus précisément des micro-textures stochastiques. " Les textures à phase aléatoire supposent une superposition linéaire. Lorsque la superposition linéaire n'est pas adaptée, comme c'est le cas pour la formation de certaines images naturelles (e.g. imagerie satellite), plusieurs auteurs lui préfèrent l'occlusion qui a abouti au modèle stochastique *dead leaves* [Ser82] [Bor06] ". Tartavel et al. [Tar14] ont récemment proposé une approche intégrant des méthodes par *patch* et des méthodes stochastiques. Ils proposent " un cadre unifié qui impose des contraintes statistiques de premier ordre sur l'utilisation d'atomes d'un dictionnaire adaptatif, ainsi que des contraintes du second ordre sur les valeurs de pixels ".

Comme mentionné précédemment, les méthodes de synthèse de textures fondées sur un modèle paramétrique consistent à construire un modèle décrivant l'image qui pourra ensuite être utilisé pour analyser la texture et aussi pour la synthétiser.
Les modèles autorégressifs (AR) supposent une interaction locale entre pixels de texture. L'intensité d'un pixel est décrite par une somme pondérée des intensités de pixels voisins [Che85]. Notons que les méthodes basées sur ce type de modèles rencontrent des difficultés à générer des textures naturelles [Hai91].
Le modèle de Wold est une méthode plus récente utilisée pour modéliser les textures. Ce modèle décompose la texture en trois composantes orthogonales [Fra93] : une composante purement aléatoire, une composante harmonique et une composante évanescente c'est-à-dire aléatoire dans une direction et déterministe dans la direction orthogonale. Ils offrent ainsi une

plus grande généralité et une plus grande souplesse que les modèles AR, mais leurs paramètres sont souvent difficiles à estimer [Sti07].

Les modèles fractals [Che93] sont exploités en infographie pour créer des textures d'images réalistes. Cela est possible parce que la dimension fractale est relativement insensible à l'échelle des images, mais plus important encore, parce qu'ils montrent une forte corrélation avec l'impression visuelle provoquée par la rugosité des surfaces. Les fractales sont notamment capable de décrire des objets ayant un degré élevé d'irrégularité. Un certain nombre de méthodes de synthèse de textures basées sur les fractales ont été proposées, s'appuyant sur un *mean shift* et sur un filtrage de Fourier [Pei88]. Les méthodes fractales représentent fidèlement des textures naturelles aléatoires, mais ne parviennent pas à décrire les structures locales de l'image par leur manque de sélectivité en termes d'orientation.

Plusieurs autres modèles ont été également proposés, comme les champs aléatoires de Markov et de Gibbs [Cro83] [Gem84], les modèles de Gabor et d'ondelettes [Tur86] [Cla87] [Mal89].

Dans ces approches, la texture est modélisée par un modèle de probabilité ou par une combinaison linéaire d'un ensemble de fonctions de base. Les principales difficultés résident dans le choix du modèle approprié pour la texture ciblée, qui entraine une perte de généricité, et dans l'estimation de ses paramètres.

2.3.4.2 L'algorithme de Wei et Levoy

L'algorithme non-paramétrique de Wei et Levoy (W&L) est basé sur une hypothèse Markovienne. Les Champs de Markov ont démontré leur capacité à modéliser une large variété de textures en les représentant comme des processus aléatoires locaux (i.e. chaque pixel est prévisible à partir des quelques pixels de son voisinage) et stationnaires (i.e. les statistiques spatiales sont invariantes par translation), cf. paragraphe 2.2.1 [Wei00]. En d'autres termes, il s'agit d'un processus probabiliste au sein duquel toutes les interactions sont locales, des interactions à plus grande échelle pouvant également se reproduire par effet de propagation.

La méthode vise à obtenir une texture de sortie localement semblable à une partie de l'échantillon d'entrée. Par conséquent, chaque pixel synthétisé est déterminé de telle sorte que la ressemblance locale est reproduite. L'algorithme prend en entrée un échantillon I_{in} et part d'une image de sortie I_{out} initialisée par un bruit blanc aléatoire. Le processus de synthèse consiste à modifier progressivement l'image de sortie afin de la faire ressembler à l'échantillon

d'entrée [Wei03]. La texture est synthétisée pixel par pixel, de manière itérative, et selon un parcours prédéfini. Pour chaque pixel de sortie à la position z' = (x, y), son voisinage $VI_{out}(z')$ est capturé, le voisinage le plus similaire $VI_{in}(z)$ est recherché dans l'exemplaire d'entrée (I_{in}) et le pixel correspondant $I_{in}(z_{min})$ est copié à la position cible dans la texture de sortie (I_{out}), comme illustré à la Figure 8. La distance Euclidienne est utilisée pour mesurer la ressemblance entre deux voisinages, exprimés sous forme vectorielle. La Figure 9 résume l'algorithme de synthèse de W&L.

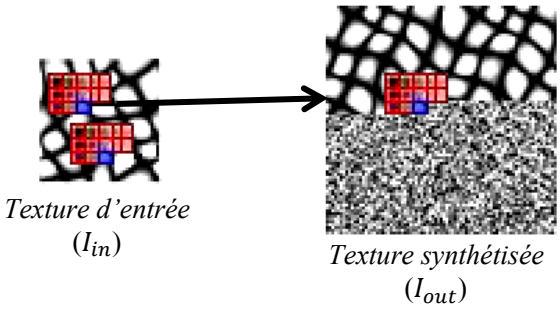


Figure 8 - *Représentation schématique du processus de synthèse de Wei et Levoy (W&L)* [Wei00] *en utilisant un voisinage causal et un parcours lexicographique.*

La synthèse est généralement effectuée en utilisant un voisinage causal (en forme de « L ») avec un parcours lexicographique (balayage en ligne). La texture de sortie est considérée périodique de sorte que la prise en considération de voisinages sur les bords de l'image ne soit pas problématique. On notera que la synthèse d'un nouveau pixel ne dépend donc que des pixels précédemment synthétisés (Figure 8). Bien que le voisinage considéré puisse contenir des pixels de bruit pour les premières lignes de la texture de sortie, dès que l'algorithme progresse, les voisinages ne contiennent que des pixels déjà synthétisés, c'est-à-dire pour lesquels le bruit a été remplacé par des valeurs de niveaux de gris plus pertinentes. En d'autres termes, le bruit ne sert que pour la synthèse des premières lignes de la texture de sortie, donnant ainsi un caractère aléatoire au résultat de synthèse, et est par la suite ignoré [Wei00]. Un voisinage non-causal avec un parcours complètement aléatoire peut également être utilisé permettant au pixel synthétisé de s'affranchir de son « passé », ce qui peut par conséquent multiplier les configurations possibles.

Pour éviter l'utilisation de larges voisinages, et donc réduire la complexité calculatoire, une synthèse multi-échelle est utilisée. Elle consiste à construire des pyramides de textures à multi-résolution pour capturer les structures d'une manière plus compacte au moyen de moins de pixels dans les échelles de la pyramide à résolution inférieure. La synthèse commence à

l'échelle ayant la plus basse résolution et s'arrête à l'échelle ayant la plus haute résolution de sorte que chaque échelle est synthétisée à partir de l'échelle supérieure.

Entrée: I_{in} , I_{out}

pour chaque position $z' = (x, y)$ de I_{out}

Extraire $VI_{out}(z')$

$z_{min} \leftarrow argmin_z\{DistanceEuclidienne(VI_{out}(z')\ vs.\ VI_{in}(z))\}$

$I_{out}(z') \leftarrow I_{in}(z_{min})$

fin pour

Sortie: I_{out}

Figure 9 – *Pseudocode de l'algorithme de synthèse de W&L.*

2.3.4.3 La synthèse 3D

Les techniques de synthèse de texture, souvent proposées dans un contexte 2D, demandent à être adaptées en vue de la modélisation de structures volumiques denses telles que celles obtenues par des techniques d'imagerie 3D, comme l'imagerie médicale, la tomographie ou l'imagerie sismique par exemple. Une autre application originale concerne la simulation de la structure de matériaux composites [Cha14].

Différentes extensions des algorithmes de synthèse de texture 2D à la synthèse de texture volumique ont été proposées, parmi lesquelles figure l'approche non-paramétrique de Kopf et al. [Kop07].

Une décomposition de Wold 3D a été proposée récemment pour la modélisation d'une texture 3D homogène sous la forme d'une somme unique de quatre composantes orthogonales [Sti07]. D'autres extensions 2D/3D d'approches paramétriques [Dac10] ou non-paramétriques [Wei03] [Urs13] se sont avérées efficaces dans de nombreux cas, y compris la conception de matériaux virtuels [Ley09] [Ley12] [Far14].

2.4 Motivation

2.4.1 Limite des méthodes existantes

Les méthodes existantes ont montré leur capacité à traiter avec succès une grande variété de textures et à fournir des résultats très satisfaisants dans de nombreux domaines d'applications. Cependant, la plupart d'entre elles échouent dans des situations difficiles telles que les textures à grandes variations directionnelles, comme cela est montré dans [Urs12] et [Urs13], par exemple, avec des textures anisotropes structurées d'échantillons de matériaux. Dans de telles situations, la plupart des approches existantes ont tendance à produire des textures plus régulières que l'exemplaire. Elles peinent notamment à synthétiser des textures ayant des structures laminaires complexes.

D'autres cas critiques concernent les textures présentant des motifs multi-échelles que les approches classiques ne parviennent pas à reproduire. Ceci est illustré à la Figure 10 sur la texture '*canvas*' [Bro66] qui présente des mailles horizontales et verticales (à l'échelle la plus fine), des lignes orientées à 45° ou 135° (échelle intermédiaire) et enfin une structure en chevron (à l'échelle la plus large). Sur une telle texture, les méthodes testées [Wei00] [Por00] [Agu13] [Tar14] [Gal11] réussissent à synthétiser les échelles les plus fines mais ne parviennent pas à reproduire les bandes périodiques de chevrons à large échelle. Ces exemples mettent en évidence la pertinence de prendre en considération des informations structurelles (géométriques) lors de la synthèse de la texture.

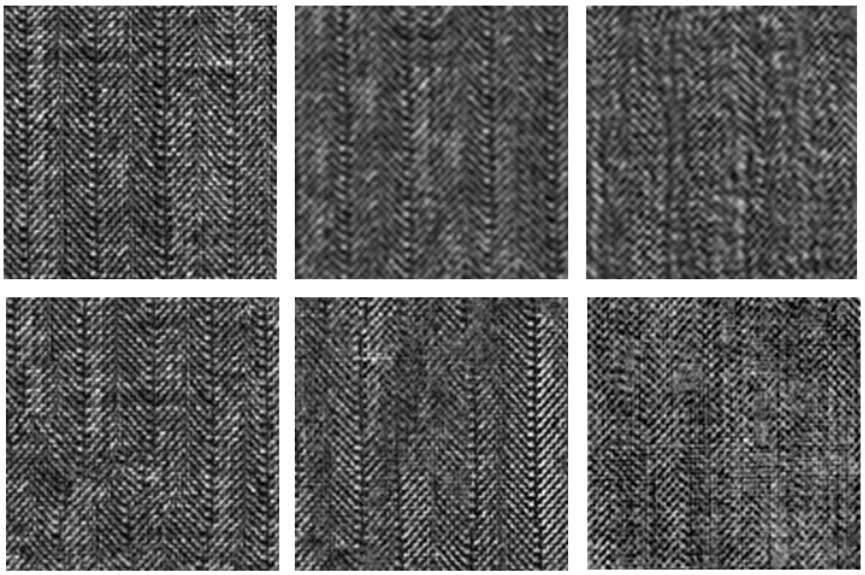


Figure 10 - *Exemple d'échecs de différents algorithmes à reproduire l'information complexe de l'exemplaire. Du haut-gauche au bas-droite : l'échantillon d'entrée et les textures synthétisées par les méthodes* [Wei00], [Por00], [Agu13], [Tar14] *et* [Gal11].

2.4.2 Approche de synthèse proposée : vue globale

Comme indiqué par Peyré [Pey09], la synthèse a priori d'une couche géométrique peut faciliter la synthèse de la texture. L'approche de synthèse en deux étapes qu'il propose consiste à produire tout d'abord un flux d'orientation synthétique qui sera utilisé, dans une deuxième étape, pour contraindre la synthèse de la texture elle-même. En effet, la première étape consiste à synthétiser une carte d'orientation à partir de la carte d'orientation de la texture exemplaire. La carte d'orientation est déduite de l'analyse du tenseur de structure local de l'exemplaire [Big87] [Knu89]. L'algorithme de synthèse constitue une variante de l'algorithme de Heeger et Bergen [Hee95]. Celui-ci reproduit les distributions marginales des coefficients issus de la décomposition multi-échelle de la carte d'orientation. La deuxième étape consiste à ajuster les distributions marginales des coefficients d'une décomposition en *grouplets* conditionnée par la carte d'orientation. La procédure de synthèse complète fonctionne très bien sur les textures turbulentes localement parallèles avec des flux d'orientation turbulents mais ne parvient pas à reproduire les textures à géométries complexes comportant des motifs croisés considérés comme des singularités du flux d'orientation.

Dans la suite du mémoire, un nouvel algorithme de synthèse à deux étapes est proposé. Il consiste à produire la couche de structure dans une première étape, et à l'utiliser pour contraindre la synthèse de la texture dans une deuxième étape. La couche de structure d'une image est analysée et représentée par son champ de tenseurs de structure local (*Local Structure Tensor - LST*) sans se limiter au flux d'orientation, mais en tenant compte à la fois de l'orientation locale, de l'énergie et du degré d'anisotropie [Big87] [Kaa87]. Dans cet objectif, l'algorithme non-paramétrique de Wei et Levoy (W&L) [Wei00] est adapté aux spécificités des champs de tenseurs de structure (chapitre 3). Cet algorithme sera également utilisé comme base de l'algorithme de synthèse de la texture contrainte par la structure (chapitre 4).

2.5 Conclusion

Dans ce chapitre, nous avons rappelé les principes fondamentaux de la synthèse de texture, de ses domaines d'applications et des différents types d'algorithmes proposés dans la littérature. L'étude bibliographique réalisée a montré que la synthèse de texture est un champ particulièrement dynamique, preuve de l'intérêt de la communauté scientifique pour ce domaine.

Toutefois, les méthodes de la littérature peinent à synthétiser certaines textures présentant une structure complexe telles que, par exemple, les textures montrant des variations directionnelles, des motifs multi-échelles voire des non stationnarités. Or il a été mis en évidence que la synthèse a priori de la couche de structure pouvait aider à une meilleure reproduction des motifs de l'exemplaire lors de la synthèse de la texture. Aussi une nouvelle approche est proposée, consistant à synthétiser dans une première étape, la couche de structure représentée par le champ de tenseurs de structure local, et de l'utiliser dans une seconde étape pour contraindre la synthèse de la texture. Cette approche est présentée au chapitre suivant.

Chapitre 3

Synthèse du champ de tenseurs de structure

Sommaire

3.1 Introduction 29
3.2 Le tenseur de structure 30
3.3 Stratégie de synthèse du champ de tenseurs de structure 36
3.3.1 Principe 36
3.3.2 Métriques de comparaison de tenseurs 37
3.3.3 Système de voisinage 39
3.3.4 Approche multi-résolution 40
3.3.5 Accélération par interpolation 43
3.3.6 Traitement spécifique du champ de tenseurs d'entrée 45
3.3.7 Traitement des bords du champ de tenseurs de sortie 46
3.4 Résultats 46
3.4.1 Comparaison des métriques de l'espace tensoriel 46
3.4.1.1 Qualité des images synthétisées 46
3.4.1.2 Complexité calculatoire 49
3.4.2 Influence de la forme du voisinage, du type de parcours et du nombre d'itérations 50
3.4.3 Influence de la taille du voisinage 54
3.4.4 Réduction de la complexité calculatoire par interpolation 56
3.5 Conclusion 59

3.1 Introduction

Dans ce chapitre, nous présentons l'algorithme de synthèse de la couche de structure. Comme indiqué au chapitre 2, la couche de structure de la texture est représentée par son champ de tenseurs de structure local (*Local Structure Tensor - LST*) qui exprime ses orientations dominantes, son degré d'anisotropie locale et son énergie [Big87] [Kaa87].

La synthèse de la couche de structure consiste à construire un nouveau champ de tenseurs de structure local d'après le champ de tenseurs de l'exemplaire. La méthode prend sa source dans l'adaptation de l'approche non-paramétrique de Wei et Levoy [Wei00], qui fonctionne généralement avec des données scalaires ou vectorielles (comme des données couleur), aux spécificités des champs de tenseurs de structure. Cette adaptation concerne le choix de métriques et de décompositions multi-échelles pertinentes. L'approche proposée dans ce chapitre permet de reproduire les statistiques marginales et spatiales du champ de tenseurs de structure. Notons que cet algorithme est exploité ici en raison de son adaptabilité. De plus, son principe de fonctionnement fondé sur des *patchs* permet la synthèse d'une grande variété de textures. Cependant, d'autres algorithmes non-paramétriques, e.g. [Pag98] [Efr99], pourraient s'y substituer.

Le champ de tenseurs de structure dont on extrait les informations d'orientation et de cohérence est introduit. Quatre métriques différentes (la métrique Euclidienne, la métrique *Shape-Orientation* [Tou10], la norme de Frobenius et la métrique Log-Euclidienne [Fil05]) sont utilisées pour mesurer la ressemblance entre deux tenseurs. Une comparaison des champs de tenseurs de structure synthétisés en utilisant ces différentes métriques et des temps de calcul associés à ces métriques, sans et avec parallélisation, est également proposée.

La qualité du champ de tenseurs synthétisé est liée à la forme du voisinage de recherche de la meilleure ressemblance, ainsi qu'au type de parcours adéquat. Un voisinage causal associé à un parcours lexicographique, ainsi qu'un voisinage carré avec un parcours complètement aléatoire sont comparés en utilisant différents échantillons d'entrée.

Dans le cas de textures contenant des structures à grande échelle, l'utilisation de larges voisinages dans le processus de synthèse est nécessaire mais coûteuse en termes de complexité calculatoire, ce qui légitime l'utilisation de décompositions multi-échelles dans différentes approches de la littérature [Bur83] [Sim92] [Pop93] [Hee95] [Jac11]. Nous montrons, dans le cas de champs de tenseurs de structure, qu'il est également possible de

mettre en œuvre une version de l'algorithme de synthèse s'appuyant sur une décomposition en pyramide Gaussienne. De plus, une interpolation bilinéaire est substituée à la synthèse de l'étage à plus haute résolution (naturellement de plus grande taille) de la pyramide Gaussienne. Finalement, la nécessité d'un voisinage de taille adéquate est prouvée, et la capacité de l'étape d'interpolation à réduire la complexité calculatoire sans nuire à la qualité de la synthèse est montrée.

3.2 Le tenseur de structure

En géométrie différentielle, le tenseur de structure, (en anglais, *structure tensor* ou *second-moment matrix*) est une matrice dérivée du gradient d'une fonction. Le gradient pointe dans la direction de la plus grande variation du champ scalaire. Le tenseur de structure permet d'analyser l'anisotropie locale autour d'un point donné en résumant les directions prédominantes du gradient dans le voisinage de ce point, et en indiquant dans quelle mesure ces directions sont cohérentes [Big87] [Knu89]. En traitement d'image, le tenseur de structure est couramment utilisé pour représenter les informations de contour et décrire des motifs locaux [Jäh93] [Ars06].

Le champ de tenseurs de structure S d'une image I est généralement défini comme le champ des matrices de covariances locales des dérivées premières partielles de I [Big87] [Kaa87] [Knu89] [Tou10]. Il est construit à partir des champs de gradients $\nabla I = [I_x\ I_y]$ préalablement estimés par convolution linéaire :

$$I_x = I * D_x \quad , \quad I_y = I * D_y \ , \tag{1}$$

où le symbole (*) désigne la convolution, D_x et D_y sont des filtres dérivatifs, en général à réponse impulsionnelle finie.

Bien que des noyaux de convolution locaux tels que les opérateurs de Prewitt [Pre70] ou de Sobel [Lyv88] puissent être choisis pour des raisons de coût calculatoire, les dérivées partielles dans le domaine discret sont généralement estimées en utilisant des noyaux construits à partir des dérivées de Gaussiennes [Per90] [Rao91] :

$$D_x(x,y) = -\frac{x}{2\pi\sigma_D^4}\, e^{-\frac{x^2+y^2}{2\sigma_D^2}} \quad , \quad D_y(x,y) = -\frac{y}{2\pi\sigma_D^4}\, e^{-\frac{x^2+y^2}{2\sigma_D^2}} \ , \tag{2}$$

où σ_D est l'écart-type d'une Gaussienne isotrope en dimension 2, tronquée à l'intervalle $[-3\sigma_D\ , 3\sigma_D]$.

Le champ de tenseurs de structure noté :

$$S = \begin{bmatrix} S_{xx} & S_{xy} \\ S_{xy} & S_{yy} \end{bmatrix}, \tag{3}$$

est calculé par convolution spatiale :

$$S = W * \nabla I \nabla I^t = W * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} W * I_x^2 & W * I_x I_y \\ W * I_x I_y & W * I_y^2 \end{bmatrix} \tag{4}$$

Un noyau de pondération Gaussien $W = G_\sigma$ – ayant un écart-type σ – est généralement considéré. Comme indiqué dans [Bro06], le lissage spatial a deux objectifs. D'une part, il rend le tenseur de structure plus robuste au bruit et, d'autre part, il permet de distribuer spatialement l'information d'orientation, principalement disponible sur les contours, c'est-à-dire sur les régions de fort gradient.
Notons que le champ de tenseurs de structure S a la même taille que l'image I, i.e. même largeur et même hauteur (abstraction faite des bords de l'image où le calcul du tenseur par convolution n'est pas possible). Il s'agit en revanche d'un champ à valeurs matricielles, $S(z)$ au site z étant une matrice symétrique semi-définie positive.

Le choix de σ est crucial pour obtenir une analyse locale pertinente de l'image. Une valeur élevée de σ implique un champ de tenseurs de structure plus lisse et une estimation d'orientation moins locale mais plus robuste au bruit. Au contraire, un faible σ assure une analyse locale précise mais avec une grande sensibilité au bruit [Tou10]. Dans la suite du mémoire, σ est choisi en fonction de la taille des motifs de la texture sous-jacente, de façon à assurer que la fenêtre de lissage Gaussienne est suffisamment large pour révéler fidèlement l'information de structure.

En chaque site z, une décomposition en éléments propres de $S(z)$ donne deux valeurs propres non-négatives $\lambda_1(z)$ et $\lambda_2(z)$ liées aux variations d'intensité dans la direction principale des gradients au voisinage de z et dans sa direction orthogonale, directions fournies respectivement par le premier et le second vecteur propre $e_1(z) = [e_{1x}(z) \quad e_{1y}(z)]^t$ et $e_2(z) = [e_{2x}(z) \quad e_{2y}(z)]^t$.

Il est courant de représenter le tenseur de structure $S(z)$ sous la forme d'une ellipse dont l'orientation $O(z)$ est calculée à partir du vecteur propre $e_1(z)$ associé à $\lambda_1(z)$ et varie entre -π/2 et π/2 :

$$O(z) = \tan^{-1}\left(\frac{e_{1y}(z)}{e_{1x}(z)}\right) \quad (5)$$

et dont l'ellipticité est donnée par le rapport des valeurs propres, le petit et le grand axe de l'ellipse étant données par les racines des valeurs propres [Dry09].

Un indicateur pertinent basé sur le tenseur de structure est le facteur de cohérence [Rao89], également appelé facteur de confiance ou indicateur de dispersion, qui est obtenu à partir des valeurs propres du tenseur de structure et varie entre 0 et 1 :

$$C(z) = \frac{\lambda_1(z) - \lambda_2(z)}{\lambda_1(z) + \lambda_2(z)} \quad (6)$$

Il est considéré comme une mesure de l'anisotropie locale et caractérise la dispersion de l'orientation du gradient i.e. la variabilité locale de la géométrie de l'image.

Dans un contexte d'estimation d'orientation, la confiance dans l'orientation estimée par le tenseur de structure est directement liée au facteur de cohérence : plus le facteur de cohérence est élevé, plus la confiance dans l'estimation de l'orientation est forte. La Figure 11 présente une représentation elliptique d'un tenseur de structure 2D ayant respectivement un faible (à gauche) et un fort (à droite) facteur de cohérence.

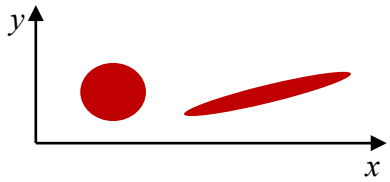


Figure 11 - *Représentation elliptique d'un tenseur de structure 2D. A gauche :* $\lambda_1(z) \approx \lambda_2(z)$ *A droite :* $\lambda_1(z) \gg \lambda_2(z)$.

L'énergie d'un tenseur de structure, définie comme la somme de ses valeurs propres (i.e. l'espérance du module du gradient au carré), est également un facteur essentiel qui caractérise la dynamique. En d'autres termes, l'énergie du tenseur en un point exprime le contraste local au voisinage de ce point [Ang14].

Dans la suite, la couche de structure est représentée par le champ de tenseurs de structure S qui associe un tenseur de structure $S(z)$ à chaque site z. Les orientations locales et la cohérence peuvent également être utilisées pour former un champ d'orientation O et un champ de cohérence C, qui peuvent être représentés comme des images.

La Figure 12 présente les images d'orientation et de cohérence du champ de tenseurs de structure de trois textures différentes. Chaque ligne montre, de gauche à droite, la texture d'entrée, l'orientation et la cohérence du champ de tenseurs de structure. La partie droite de la figure présente la palette utilisée pour représenter les images d'orientation et de cohérence. Notons que cette palette est utilisée pour tous les résultats qui suivent. La texture de la première ligne est de taille 256×256 tandis que les deux autres textures sont de taille 128×128. Pour les trois résultats, des écarts-types de 1 et de 1.5 sont utilisés pour les noyaux Gaussiens employés pour le calcul du gradient et du tenseur de structure.

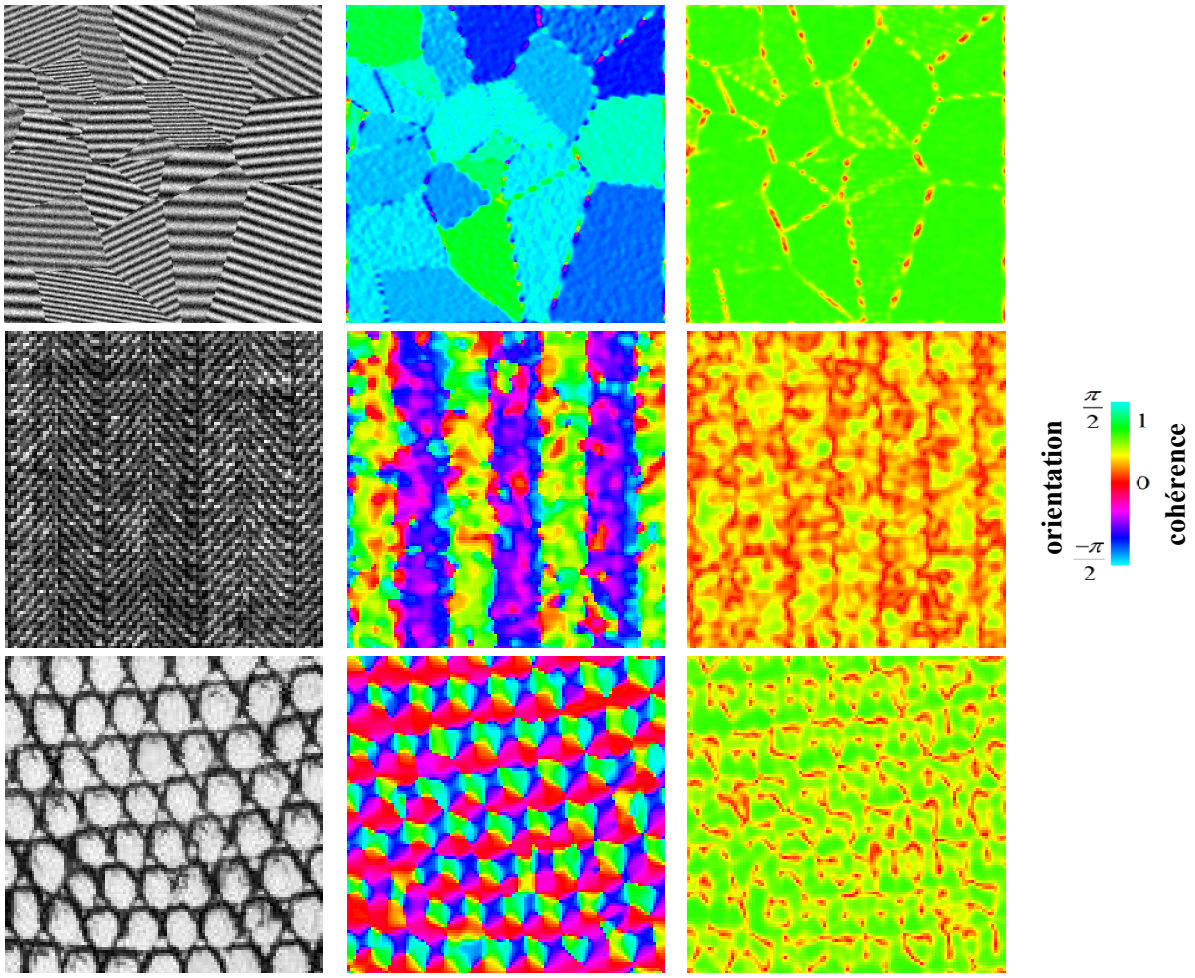


Figure 12 – *Exemples de champs de tenseurs de structure. 1ère, 2ème et 3ème ligne (de gauche à droite) : l'échantillon d'entrée, l'orientation et la cohérence du champ de tenseurs de structure. Au milieu à droite : palette utilisée pour les images d'orientation et de cohérence. La texture de la 1ère ligne est de taille 256×256. Celles de la 2ème et 3ème lignes sont de taille 128×128. Des valeurs de 1 et 1.5 sont utilisées pour les écarts-types de la Gaussienne du gradient et du noyau de pondération Gaussien, respectivement.*

Si l'on observe la première ligne de la Figure 12 (pavage constitué de régions présentant différentes orientations, différents niveaux de bruit et différentes amplitudes), on peut clairement remarquer d'après l'image d'orientation que le champ de tenseurs de structure de la texture rend compte des changements d'orientations. Des fluctuations d'orientation sont

observées dans les régions ayant un niveau de bruit élevé. Une augmentation de la taille du noyau de pondération Gaussien aurait ici permis un lissage du champ d'orientation. En outre, l'image de cohérence montre que la cohérence est élevée à l'intérieur des régions, indépendamment du contraste (amplitude des motifs), tandis que les frontières des régions, marquées par un changement d'orientation des motifs, présentent une cohérence plus faible. Les lignes orientées à 45° ou 135° de la deuxième texture (texture '*canvas*' [Bro66]) sont de même détectées par le tenseur de structure révélant la structure en chevrons à l'échelle la plus large. En revanche, les mailles horizontales et verticales de l'échelle la plus fine sont plus difficiles à mettre en évidence du fait du lissage inhérent à l'estimation du tenseur. Finalement, les variations d'orientations et de niveaux de cohérences régulières et répétitives de la troisième texture structurée montrent également la capacité du tenseur à bien représenter la structure de l'image texturée.

La Figure 13 illustre un nouvel exemple de champ de tenseurs de structure. De haut en bas et de gauche à droite sont respectivement représentés une texture synthétique de taille 256×256, son champ de tenseurs de structure sous la forme d'un champ d'ellipses, les images d'orientation et de cohérence du champ de tenseurs. Des valeurs de 1 et 1.5 sont utilisées pour l'écart-type de la Gaussienne du gradient et l'écart-type du noyau de pondération Gaussien. Une grille ou pas d'affichage 10:1 est utilisée pour le champ d'ellipses (i.e. une ellipse est affichée tous les 10 pixels de l'exemplaire). L'ellipse en un site z représente le tenseur de structure $S(z)$ ayant son orientation locale $O(z)$ et son facteur de cohérence $C(z)$. On peut clairement remarquer que le champ de tenseurs affiché comme champ d'ellipses représente finalement la structure de la texture exemplaire. La carte d'orientation du champ de tenseurs exprime fidèlement les orientations dominantes. Finalement, l'image de cohérence issue du champ de tenseurs de structure de l'exemplaire montre que ce dernier présente une forte cohérence.

La Figure 14 montre un exemple de calcul de champ de tenseurs de structure avec différentes valeurs de l'écart-type du noyau Gaussien employé pour le calcul du tenseur. La première colonne montre une version bruitée de l'exemplaire utilisé dans la Figure 13. Les deuxième à quatrième colonnes montrent les orientations (en haut) et les cohérences (en bas) des champs de tenseurs de structure avec des écarts-types de pondération Gaussienne respectivement égaux à 0.5, 2 et 3.5. L'écart-type de la Gaussienne du gradient est choisi égal à 1.

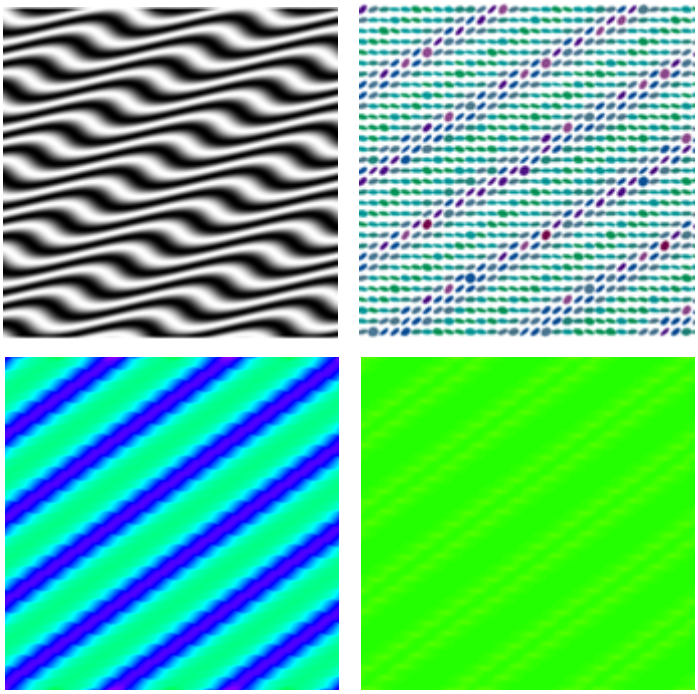


Figure 13 – *Illustration d'un champ de tenseurs de structure. Haut-gauche au bas-droite : l'exemplaire (taille 256×256), son champ de tenseurs représenté sous la forme d'un champ d'ellipses (grille d'affichage 10:1), l'orientation et la cohérence du champ de tenseurs. Des écarts-types de 1 et de 1.5 sont utilisés pour la Gaussienne du gradient et pour le noyau de pondération Gaussien, respectivement.*

Si l'on observe la deuxième colonne de la Figure 14, on peut remarquer que le noyau Gaussien d'écart type 0.5 révèle une forte sensibilité au bruit. D'autre part, un écart-type très élevé (3.5) conduit à un lissage excessif de la structure (voir quatrième colonne de la Figure 14). En revanche, une valeur intermédiaire (2) constitue un compromis entre robustesse au bruit et estimation locale de la structure. Cet exemple montre que le choix de l'écart-type du noyau de pondération Gaussien est crucial pour obtenir une analyse locale pertinente de la texture.

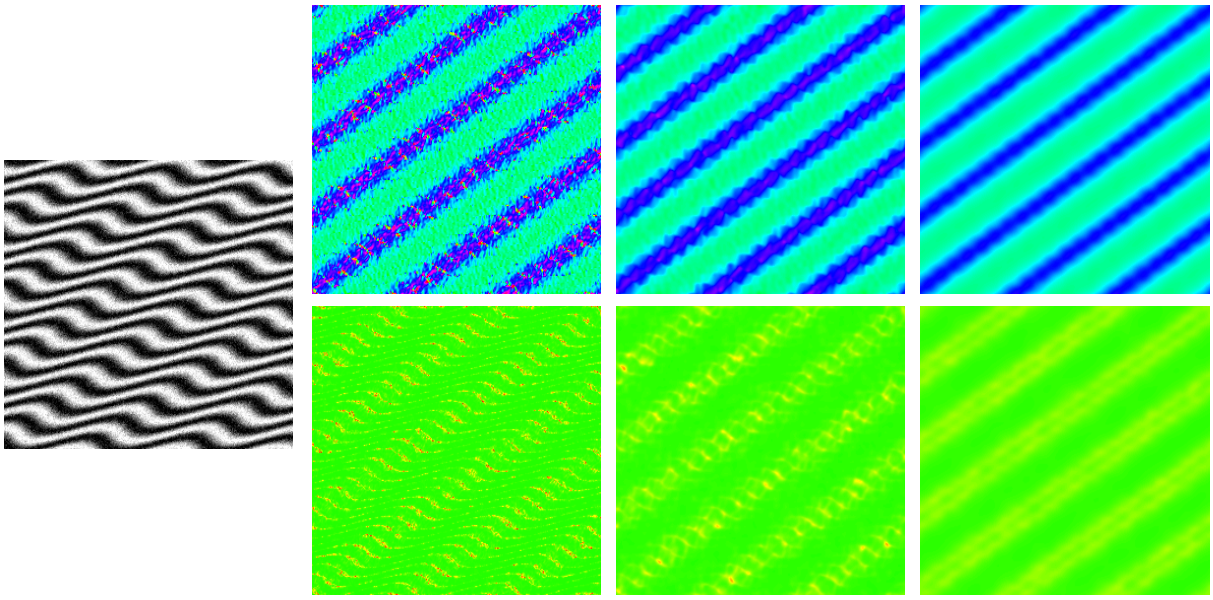


Figure 14 – *Exemple de calcul de champs de tenseurs de structure avec différentes valeurs de l'écart-type σ du noyau de pondération Gaussien. 1ère colonne : version bruitée de l'exemplaire de la Figure 13. 2ème, 3ème et 4ème colonne : orientations (en haut) et cohérences (en bas) des champs de tenseurs de structure avec σ = 0.5, 2 et 3.5, respectivement.*

3.3 Stratégie de synthèse du champ de tenseurs de structure

3.3.1 Principe

L'algorithme de Wei et Levoy (W&L) [Wei00] permet initialement de traiter des données scalaires ou bien des données vectorielles (des données couleur notamment). L'utilisation de cet algorithme pour des champs de tenseurs de structure requiert donc une adaptation aux champs matriciels et en particulier aux spécificités des matrices symétriques semi-définies positives. Cette adaptation réside principalement dans le choix de la métrique utilisée pour la comparaison des tenseurs de la texture d'entrée et de sortie. Mise à part le choix de la métrique, le processus de synthèse est similaire à celui de Wei et Levoy [Wei00]. L'algorithme commence par le calcul du champ de tenseurs de structure d'entrée S_{in} à partir de l'exemplaire I_{in}. Ensuite, un champ de tenseurs de sortie S_{out} est initialisé par un bruit de tenseurs de structure (i.e. tenseurs choisis aléatoirement dans le champ de tenseurs de structure d'entrée). L'algorithme de synthèse consiste à modifier S_{out} itérativement pour qu'il ressemble à S_{in}.

Soit z un site dans le champ de tenseurs de structure d'entrée S_{in} et z' un site dans le champ de tenseurs de structure de sortie S_{out}. En chaque site z' du champ de tenseurs de structure de sortie, les tenseurs de structure dans le voisinage de z' sont organisés sous la forme d'un vecteur de tenseurs de structure. Ce vecteur, que nous appellerons par abus de langage

voisinage du tenseur $S_{out}(z')$, est alors comparé à tous les voisinages des tenseurs $S_{in}(z)$ de l'image d'entrée. La similarité est mesurée au sens d'une métrique dont plusieurs exemples seront proposés au paragraphe 3.3.2. Le site z_{min} du voisinage le plus similaire est alors retenu. Enfin le tenseur de structure $S_{in}(z_{min})$ est copié à la position cible z' dans S_{out} : $S_{out}(z') \leftarrow S_{in}(z_{min})$. Tous les sites du champ de sortie sont traités séquentiellement jusqu'à ce que tous les tenseurs soient déterminés. Enfin, l'ensemble de cette procédure de synthèse peut être itéré plusieurs fois. Le nombre d'itérations nécessaire pour aboutir à une qualité de synthèse acceptable dépend de la forme du voisinage et du parcours de synthèse, comme il sera montré ultérieurement. Cette méthode de synthèse vise à assurer que le tenseur choisi préservera la similarité locale entre les champs de tenseurs d'entrée et de sortie.

3.3.2 Métriques de comparaison de tenseurs

Concernant les spécificités du cas des tenseurs de structure, la similarité entre deux tenseurs $S_{in}(z)$ et $S_{out}(z')$ peut être mesurée au moyen de différentes métriques. En plus du carré de la distance Euclidienne (*Square Euclidean Distance - SED*) définie par :

$$M_{SED}\big(S_{in}(z), S_{out}(z')\big) = [(S_{in,xx}(z) - S_{out,xx}(z'))^2 + (S_{in,yy}(z) - S_{out,yy}(z'))^2 + 2(S_{in,xy}(z) - S_{out,xy}(z'))^2] \ , \tag{7}$$

trois autres métriques sont considérées : la métrique *Shape-Orientation* (*SO*), la norme de Frobenius (*Frobenius Norm - FN*), et la métrique Log-Euclidienne (*LE*).

La métrique *Shape-Orientation* mesure la dissimilarité entre deux tenseurs en termes de propriétés géométriques (facteurs d'orientation et de cohérence) comme indiqué dans [Tou10] :

$$M_{SO,\alpha}\big(S_{in}(z), S_{out}(z')\big) = M_{or}\big(S_{in}(z), S_{out}(z')\big)^{\alpha} \cdot M_{coh}\big(S_{in}(z), S_{out}(z')\big)^{1-\alpha} \ , \tag{8}$$

où α est un facteur de pondération entre 0 et 1. M_{or} est la mesure de dissimilarité d'orientation calculée comme suit :

$$M_{or}\big(S_{in}(z), S_{out}(z')\big) = \left|\sin\big(O_{in}(z) - O_{out}(z')\big)\right| \cdot min\big(C_{in}(z), C_{out}(z')\big) \ , \tag{9}$$

et M_{coh} est la mesure de dissimilarité de cohérence exprimée par :

$$M_{coh}\big(S_{in}(z), S_{out}(z')\big) = max\left(\frac{C_{in}(z)}{C_{out}(z')}, \frac{C_{out}(z')}{C_{in}(z)}\right). \quad (10)$$

Il apparait que la métrique $M_{SO,\alpha}$ n'utilise pas l'ensemble de l'information véhiculée par le tenseur de structure mais s'appuie uniquement sur l'orientation et la cohérence. Elle a toutefois l'avantage de permettre d'ajuster l'importance accordée à l'une ou l'autre de ces deux composantes.

La métrique *FN* [Fil05] est donnée par :

$$M_{FN}\big(S_{in}(z), S_{out}(z')\big) = [trace([S_{in}(z) - S_{out}(z')]^t[S_{in}(z) - S_{out}(z')])]^{1/2} \ , \quad (11)$$

où l'opérateur $[\ ]^t$désigne la transposée d'une matrice.

La métrique Log-Euclidienne (*LE*) [Fil05], aussi appelée Riemannienne, s'exprime par :

$$M_{LE}\big(S_{in}(z), S_{out}(z')\big) = \Big[trace\left(\big[\hat{S}_{in}(z) - \hat{S}_{out}(z')\big]^t\big[\hat{S}_{in}(z) - \hat{S}_{out}(z')\big]\right)\Big]^{1/2} \ , \quad (12)$$

où $\hat{S}$ est le logarithme de la matrice S.

Notons que d'autres métriques Riemanniennes ont été proposées dans la littérature, dans des contextes variés, comme la métrique de Cholesky [Wan04], la divergence root Stein [Sra12], la distance affine-invariant [Pen06] ou la distance power-Euclidean [Jay13].

Soit $N_K(z)$ le voisinage d'un site *z* : $N_K(z) = \{t_k\ , k = 1\ to\ K\}$ un ensemble de *K* sites t_k (par exemple carré ou en forme de « L »). Par abus de langage mais dans un souci de concision, on appelle « voisinage de tenseurs de structure » au site *z* d'une image tensorielle *S*, l'ensemble $VS(z) = \{S(t), t \in N_K(z)\}$.
Deux voisinages de tenseurs de structure $VS_{in}(z)$ et $VS_{out}(z')$ des sites *z* et *z'* peuvent être comparés en calculant la somme des valeurs de dissimilarité des tenseurs de structure (*Sum of Structure Tensor Dissimilarity - SSTD*) comme suit :

$$SSTD\big(VS_{in}(z), VS_{out}(z')\big) = \sum_{k=1}^{K} M(S_{in}(t_k), S_{out}(t'_k)) \ , \quad (13)$$

où $t_k \in N_K(z)$, $t'_k \in N_K(z')$, *K* est le nombre de tenseurs de structure dans chaque voisinage, $S_{in}(t_k)$ (resp. $S_{out}(t'_k)$) désigne le $k^{\text{ème}}$ tenseur de structure du voisinage $VS_{in}(z)$ (resp. $VS_{out}(z')$) et *M* est l'une des quatre métriques : M_{SED}, $M_{SO,\alpha}$, M_{FN} et M_{LE}.

La procédure de synthèse du champ de tenseurs est résumée à la Figure 15.

Entrée: I_{in}

$S_{in} \leftarrow$ CHAMPTENSEURSDESTRUCTURE (I_{in})

$S_{out} \leftarrow$ INITIALISATIONBRUITTENSEURSDESTRUCTURE (S_{in})

Tant que iter ≤ nbr_itérations

pour chaque position $z' = (x, y)$ de S_{out}

Extraire $VS_{out}(z')$

$z_{min} \leftarrow argmin_z\{SSTD(VS_{in}(z), VS_{out}(z'))\}$

$S_{out}(z') \leftarrow S_{in}(z_{min})$

fin pour

Incrémenter iter

Fin tant que

Sortie: S_{out}

Figure 15 – *Pseudocode de l'approche de synthèse du champ de tenseurs de structure.*

3.3.3 Système de voisinage

Une des tâches les plus importantes dans le processus de synthèse du champ de tenseurs est le choix du système de voisinage (la forme du voisinage et le type de parcours adéquat) qui influe directement sur l'image tensorielle synthétisée. Une solution courante consiste à utiliser un voisinage causal en forme de « L », avec un parcours lexicographique (Figure 16 – colonne gauche) [Wei00] ou encore un voisinage carré non-causal associé à un parcours complètement aléatoire (Figure 16 – colonne droite) [Urs13, Urs14]. Des parcours alternatifs comme les parcours en Z ou les parcours de Hilbert ont également été proposés en synthèse de texture [Urs12] sans pour autant se démarquer de façon claire.

Dans le cas d'un voisinage causal, la synthèse d'un pixel à valeur tensorielle dépend totalement des tenseurs précédemment traités ce qui peut aboutir à un haut degré de régularité du champ de tenseurs synthétisé. Le bruit de tenseur a un effet lors de la synthèse des premières lignes et colonnes du champ mais est ensuite ignoré. Au contraire, le parcours aléatoire permet au tenseur synthétisé de s'affranchir de son « passé » et par conséquent de multiplier les configurations possibles.

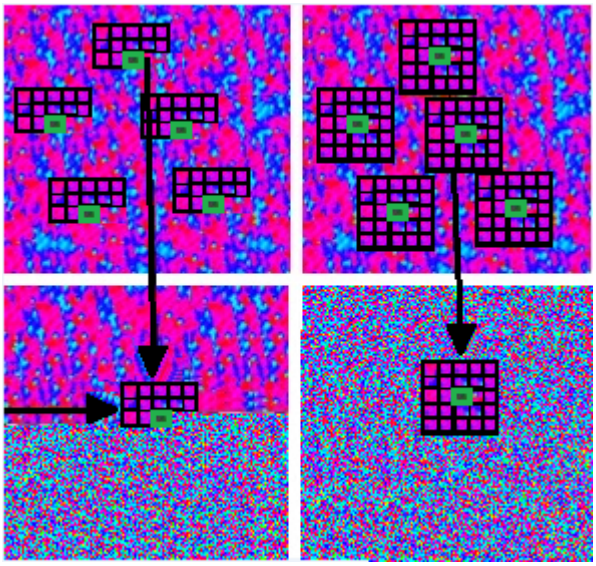


Figure 16 – *Voisinage causal, en forme de « L », adapté à un parcours lexicographique (colonne gauche) et voisinage carré non-causal utilisé avec un parcours aléatoire (colonne droite). Notons que les images montrées sont les orientations d'un champ de tenseurs de structure.*

3.3.4 Approche multi-résolution

Le voisinage local étant utilisé comme modèle fondamental dans l'algorithme de synthèse, la qualité des résultats synthétisés dépend de sa taille. Par conséquent, la taille du voisinage doit être choisie de manière appropriée dans le but de préserver la structure de l'échantillon d'entrée. Intuitivement, elle doit être déterminée en fonction de l'échelle du plus grand motif observé au risque de perdre la structure [Wei00].

Pour cette raison, une texture contenant des structures à grande échelle nécessite un voisinage de grande taille ce qui est coûteux en termes de calculs. Afin d'éviter l'utilisation de larges voisinages, des pyramides à multi-résolution sont utilisées pour capturer les structures de façon plus compacte au moyen de moins de tenseurs dans les niveaux de la pyramide à résolution inférieure. Le nombre de niveaux de la pyramide a autant d'influence que la taille du voisinage.

Dans notre algorithme de synthèse multi-échelle, des pyramides de tenseurs de structure à multi-résolution sont mises en œuvre. Les pyramides Laplaciennes [Bur83] [Hee95], les pyramides Gaussiennes [Pop93] [Jac11] et les pyramides orientables (*Steerable pyramids*) [Sim92] [Hee95] sont toutes des approches éligibles. Dans ce mémoire, les pyramides Gaussiennes [Pop93] sont utilisées; chaque niveau de la pyramide est obtenu en lissant l'image tensorielle du niveau inférieur avec un noyau Gaussien puis en sous-échantillonnant avec un facteur 2:1. Ainsi, pour passer d'un niveau L au niveau suivant

(supérieur) *L+1*, l'algorithme multi-résolution filtre et réduit le nombre de tenseurs (Figure 17) [Urs13].

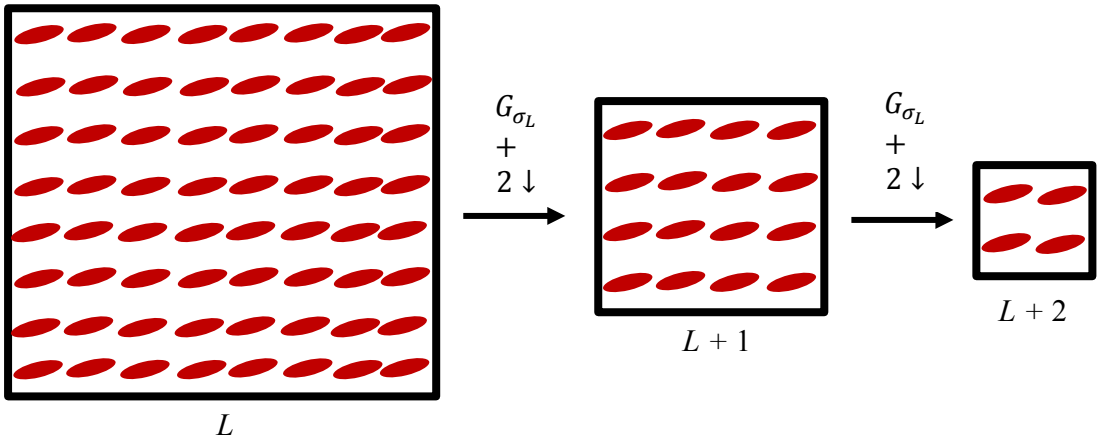


Figure 17 – *Principe de construction d'une pyramide Gaussienne de tenseurs de structure : pour passer d'un niveau L au niveau supérieur L+1, l'algorithme filtre avec un noyau Gaussien* G_{σ_L} *et réduit le nombre de tenseurs* (2 ↓).

L'algorithme de synthèse du champ de tenseurs de structure opère comme suit : les pyramides de tenseurs de structure d'entrée (PS_{in}) et de sortie (PS_{out}) sont créées à partir du champ de tenseurs de structure d'entrée S_{in} et le champ de bruit de tenseurs de sortie S_{out}, respectivement. La synthèse commence au niveau le plus élevé de la pyramide (PS_{out}^{NL-1} ; *NL* est le nombre d'échelles dans la pyramide) ayant la plus basse résolution, et se termine à la résolution la plus élevée, c'est-à-dire au niveau le plus bas de la pyramide (PS_{out}^{0}), de sorte que chaque niveau est construit à partir du niveau de résolution inférieure déjà synthétisé (Figure 18 – gauche) [Wei00].

Plus précisément, pour s'assurer que les détails à haute fréquence ajoutés sont compatibles avec les structures basses fréquences déjà synthétisées, le voisinage multi-résolution de tenseurs de structure $VS_L^{mr}(z)$ au site $z = (x, y)$ du niveau L, contient son voisinage $VS_L(z)$ au même niveau ainsi que le voisinage $VS_{L+1}(z_\downarrow)$ du site correspondant $z_\downarrow=(x/2, y/2)$ au niveau $L+1$ précédemment synthétisé (Figure 18 – droite) [Wei00] :

$$VS_L^{mr}(z) = VS_L(z) \cup VS_{L+1}(z_\downarrow) \tag{14}$$

où $VS_L(z) = \{S_L(t), t \in N_L(z)\}$ et $VS_{L+1}(z_\downarrow) = \{S_{L+1}(t_\downarrow), t_\downarrow \in N_{L+1}(z_\downarrow)\}$.

Notons que même si un voisinage $VS_L(z)$ causal est considéré au niveau L, un voisinage $VS_{L+1}(z_\downarrow)$ carré est utilisé au niveau $L + 1$ puisque ce dernier est complètement synthétisé

avant de passer au niveau L. La Figure 19 résume l'algorithme de synthèse multi-échelle du champ de tenseurs de structure.

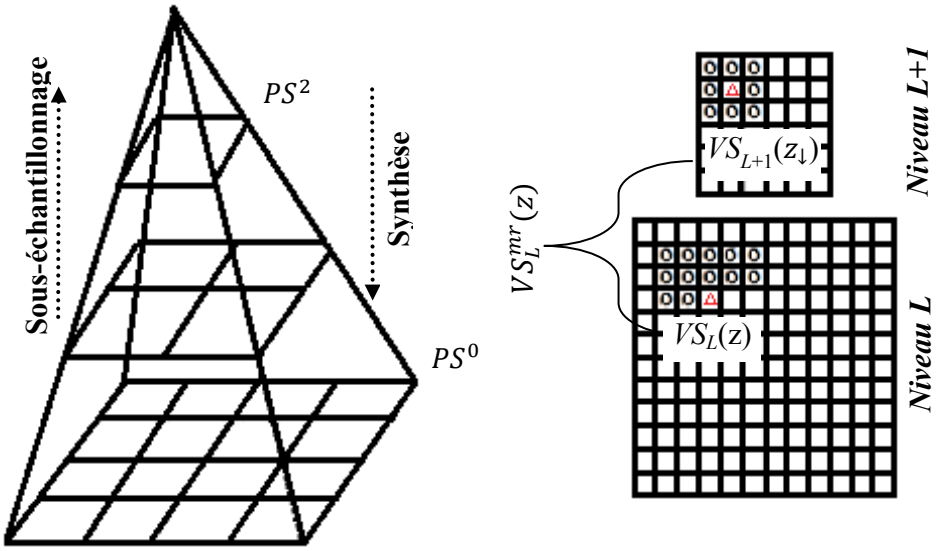


Figure 18 – *Illustration multi-résolution : pyramide Gaussienne à trois niveaux (à gauche) avec un exemple de voisinage causal de taille 5×5 (à droite)* [Urs13].

D'autres solutions ont été proposées pour gérer l'aspect multi-échelle. Par exemple, celle de Paget et Longstaff [Pag98] reprise dans [Urs13] consiste à synthétiser à la résolution la plus faible puis à sur-échantillonner et synthétiser successivement afin de construire les autres niveaux de la pyramide.

Entrée: I_{in}

S_{in} ← CHAMPTENSEURSDESTRUCTURE (I_{in})

PS_{in} ← DECOMPOSITIONPYRAMIDEGAUSSIENNE (S_{in})

S_{out} ← INITIALISATIONBRUITTENSEURSDESTRUCTURE (S_{in})

PS_{out} ← DECOMPOSITIONPYRAMIDEGAUSSIENNE (S_{out})

Tant que iter ≤ nbr_itérations

  pour chaque niveau L de $NL - 1$ à 0

    pour chaque position $z' = (x, y)$ de PS_{out}^{L}

      Extraire $VS_{L}^{mr}(z')_{out}$

      $z_{min} \leftarrow argmin_{z}\{SSTD(VS_{L}^{mr}(z)_{in}, VS_{L}^{mr}(z')_{out})\}$

      $PS_{out}^{L}(z') \leftarrow PS_{in}^{L}(z_{min})$

    fin pour

  fin pour

  Incrémenter iter

Fin tant que

Sortie: PS_{out}^{0}

Figure 19 – *Pseudocode de l'approche de synthèse multi-échelle du champ de tenseurs de structure.*

3.3.5 Accélération par interpolation

En pratique, durant le processus de synthèse multi-échelle, la charge de calcul réside principalement dans la synthèse du niveau le plus bas (à plus haute résolution) de la pyramide Gaussienne. En outre, pour la plupart des textures, les informations de structure dans le niveau le plus bas de la pyramide diffèrent très peu de celles existant dans le niveau supérieur.

Par conséquent, ayant une pyramide de tenseurs de structure de sortie PS_{out} à NL niveaux, où PS_{out}^{NL-1} est le plus haut niveau ayant la plus basse résolution, et PS_{out}^{0} est le niveau le plus bas ayant la plus haute résolution, l'algorithme de synthèse accéléré opère comme suit :

- La synthèse des champs de tenseurs de structure commence au niveau NL-1, comme dans l'algorithme classique, mais elle s'arrête à un niveau Ls.

- Le niveau de la pyramide PS_{out}^{Ls} étant synthétisé, le niveau PS_{out}^{Ls-1} est construit à partir de PS_{out}^{Ls} à l'aide d'une interpolation bilinéaire (Figure 20).
- Le même principe s'applique pour construire le niveau PS_{out}^{Ls-2} à partir de PS_{out}^{Ls-1}, et ainsi de suite jusqu' au niveau le plus bas PS_{out}^{0} interpolé à partir de PS_{out}^{1}.

Ceci peut être considéré comme une alternative à l'algorithme de synthèse multi-échelle qui consiste à synthétiser tous les étages de la pyramide Gaussienne.

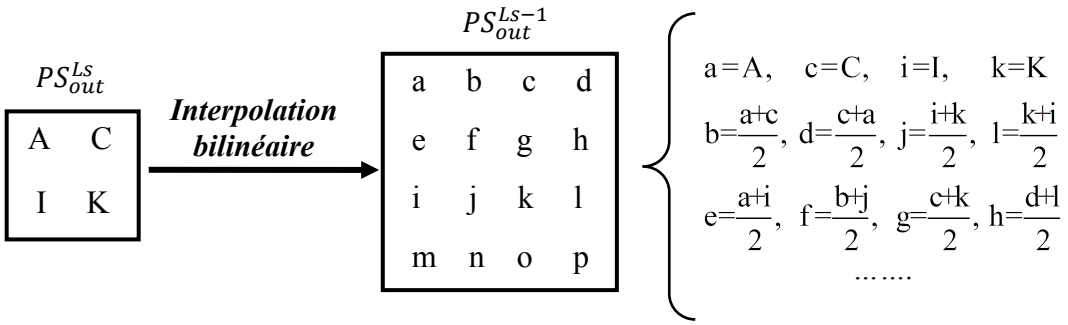


Figure 20 – *Illustration de l'interpolation bilinéaire utilisée pour l'accélération de l'algorithme de synthèse du champ de tenseurs de structure.*

La Figure 21 présente le pseudocode de l'algorithme de synthèse accéléré par interpolation bilinéaire.

Bien qu'il s'avère difficile de trouver une valeur optimale du paramètre *Ls* qui indique le dernier niveau synthétisé et à partir duquel la construction par interpolation commence, il sera montré dans le paragraphe 3.4.4 qu'une valeur de *Ls* égale à 1 conduit pour de nombreuses textures à des résultats satisfaisants.

Notons que l'interpolation bilinéaire est exploitée dans cette approche en raison de sa faible complexité calculatoire. De plus, elle a montré sa capacité à aboutir à de bons résultats sur une grande variété de textures. Cependant, d'autres méthodes d'interpolation pourraient s'y substituer comme par exemple l'interpolation fréquentielle par *zero-padding*. Son implémentation par *FFT* (*Fast Fourier Transform*) permet de gérer la périodicité de l'image, tout en garantissant une complexité calculatoire raisonnable.

En outre, d'autres méthodes de recherche de la meilleure ressemblance pourraient se substituer à l'argument du minimum (*argmin*) utilisé dans cet algorithme (Figure 21), comme celle de l'approche de Efros et Leung [Efr99].

Entrée: I_{in}

$S_{in} \leftarrow$ CHAMPTENSEURSDESTRUCTURE (I_{in})

$PS_{in} \leftarrow$ DECOMPOSITIONPYRAMIDEGAUSSIENNE (S_{in})

$S_{out} \leftarrow$ INITIALISATIONBRUITTENSEURSDESTRUCTURE (S_{in})

$PS_{out} \leftarrow$ DECOMPOSITIONPYRAMIDEGAUSSIENNE (S_{out})

Tant que iter ≤ nbr_itérations

pour chaque niveau L de $NL - 1$ à Ls

pour chaque position $z' = (x, y)$ de PS_{out}^{L}

Extraire $VS_{L}^{mr}(z')_{out}$

$z_{min} \leftarrow argmin_{z}\{SSTD(VS_{L}^{mr}(z)_{in}, VS_{L}^{mr}(z')_{out})\}$

$PS_{out}^{L}(z') \leftarrow PS_{in}^{L}(z_{min})$

fin pour

fin pour

pour chaque niveau L de $Ls - 1$ à 0

$PS_{out}^{L} \leftarrow$ INTERPOLATIONBILINEAIRE (PS_{out}^{L+1})

fin pour

Incrémenter iter

Fin tant que

Sortie: PS_{out}^{0}

Figure 21 – *Pseudocode de l'approche de synthèse du champ de tenseurs de structure accélérée.*

3.3.6 Traitement spécifique du champ de tenseurs d'entrée

Comme mentionné précédemment, le champ de tenseurs de structure de l'échantillon d'entrée est construit à partir des champs de gradients estimés à l'aide des noyaux de dérivées Gaussiennes [Per90] [Rao91] (cf. équation (2)). Etant donné que ces fenêtres Gaussiennes conduisent à considérer des sites en dehors des frontières de l'exemplaire, un traitement spécifique aux bords est nécessaire. Le calcul du gradient se limite donc à l'intérieur de l'exemplaire laissant une zone inutilisée dans la procédure de recherche de la meilleure ressemblance. En fait, la largeur de cette zone inutilisée, qui sera qualifiée d'inutile par la

suite, dépend de la taille des noyaux de dérivées Gaussiennes et de la taille du noyau de lissage Gaussien du tenseur (cf. équation (4)). En outre, dans l'approche de synthèse multi-résolution, la largeur de la zone inutile de chaque étage de la pyramide, dépend également de la taille de la fenêtre de lissage Gaussienne utilisée pour construire les différents étages de la pyramide (cf. paragraphe 3.3.4). Le calcul de la largeur de la zone inutile est développé en Annexe A.

3.3.7 Traitement des bords du champ de tenseurs de sortie

Etant donné que dans le processus de synthèse du champ de tenseurs de structure, le voisinage de synthèse dépasse les bords extérieurs du champ de tenseurs de sortie, les bords doivent subir un traitement spécifique. La méthode la plus simple est de ne pas considérer les tenseurs aux bords et de faire la synthèse strictement à l'intérieur du champ de tenseurs. Un traitement plus approprié des tenseurs de bords consiste à gérer le champ de tenseurs de sortie comme une donnée toroïdale (i.e. périodique) [Wei00]. Ainsi, dans le cas d'une synthèse multi-échelle, tous les étages de la pyramide de tenseurs de structure sont traités périodiquement de la même manière que dans l'approche mono-échelle. La construction d'un champ de tenseurs périodique est détaillée en Annexe B. Notons que la périodicité du champ ainsi créé permet très facilement de produire des champs de plus grande dimension par simple tuilage horizontal et vertical. Cependant, des distorsions aux bords peuvent apparaître dans le cas où la taille des motifs n'est pas compatible avec la périodicité du champ.

3.4 Résultats

Dans cette partie, les résultats de synthèse de champs de tenseurs de structure obtenus avec les quatre métriques tensorielles sont comparés. L'influence du système de voisinage, du nombre d'itérations, et de la taille du voisinage est ensuite étudiée. Enfin, le gain obtenu par la méthode d'accélération proposée, en termes de complexité calculatoire, est observé en comparant les temps de calcul nécessaires pour synthétiser différents champs de tenseurs de structure sans et avec interpolation. Par souci de concision et pour éviter les redondances, un sous-ensemble de résultats est présenté.

3.4.1 Comparaison des métriques de l'espace tensoriel

3.4.1.1 Qualité des images synthétisées

Dans ce paragraphe, les quatre métriques de l'espace tensoriel présentées au paragraphe 3.3.2 sont comparées. La Figure 22 montre plusieurs résultats de synthèse de

champs de tenseurs de structure représentés par leurs images d'orientations (la figure comportant les images de cohérence est donnée en Annexe C). Dans les première, deuxième et troisième colonnes, les images d'entrée et de sortie sont de taille 64×64 et 128×128, respectivement. Dans la quatrième colonne, les images d'entrée et de sortie sont de taille 256×256 et 512×512, respectivement. Des pyramides Gaussiennes à deux échelles sont utilisées avec un voisinage causal, un parcours lexicographique et trois itérations. Des voisinages de taille 11×11, 15×15, 13×13 et 9×9 sont utilisés pour les textures (*a*), (*b*), (*c*) et (*d*) respectivement. La deuxième ligne (*Oa*-*Od*) présente les champs d'orientation des tenseurs de structure calculés sur les textures de la première ligne (*a*-*d*).

La troisième ligne (Oa_M_{SED}-Od_M_{SED}) présente les orientations des champs de tenseurs de structure qui sont synthétisés en se basant sur le carré de la distance Euclidienne. Cette métrique fonctionne bien sur les textures (*a*) (*b*) et (*d*) alors que le champ d'orientation dans Oc_M_{SED} semble plus régulier que l'entrée : des motifs de même taille répétés sont visibles. L'utilisation d'un voisinage non-causal avec un parcours aléatoire pour multiplier les configurations éventuelles serait une solution possible dans ce cas. Les quatre images d'orientation de la quatrième ligne ($Oa_M_{SO,\alpha}$-$Od_M_{SO,\alpha}$) obtenues avec la métrique *Shape-Orientation* sont assez similaires à ceux de l'entrée. Les images d'orientation Oa_M_{FN}, Ob_M_{FN} et Od_M_{FN} obtenues avec la norme de Frobenius sont de bonne qualité. Cependant, des artefacts indésirables peuvent être observés avec la troisième texture dans Oc_M_{FN}. Les résultats Oa_M_{LE}, Ob_M_{LE} et Od_M_{LE} montrent la capacité de la métrique Log-Euclidienne à capturer les caractéristiques visuelles des champs de tenseurs de structure d'entrée et à conserver les structures de la texture. Toutefois, de même que Oc_M_{SED}, Oc_M_{LE} semble plus régulière que *Oc*.

On peut remarquer que la plupart des cartes d'orientation synthétiques sont très similaires aux cartes d'orientation d'entrée et donnent l'impression d'avoir été générées par le même processus sous-jacent. En outre, pour certaines textures, le changement de la métrique ne donne pas de variations notables dans le champ de tenseurs de structure de sortie, comme montré pour les textures (*a*), (*b*) et (*d*) alors que pour d'autres textures, un choix approprié de la métrique peut aboutir à un résultat mieux synthétisé comme c'est le cas pour la texture (*c*), où seule la métrique *Shape-Orientation* a conduit à une synthèse satisfaisante. Par conséquent, on peut conclure que l'effet apporté par chaque métrique reste dépendant de la texture d'entrée.

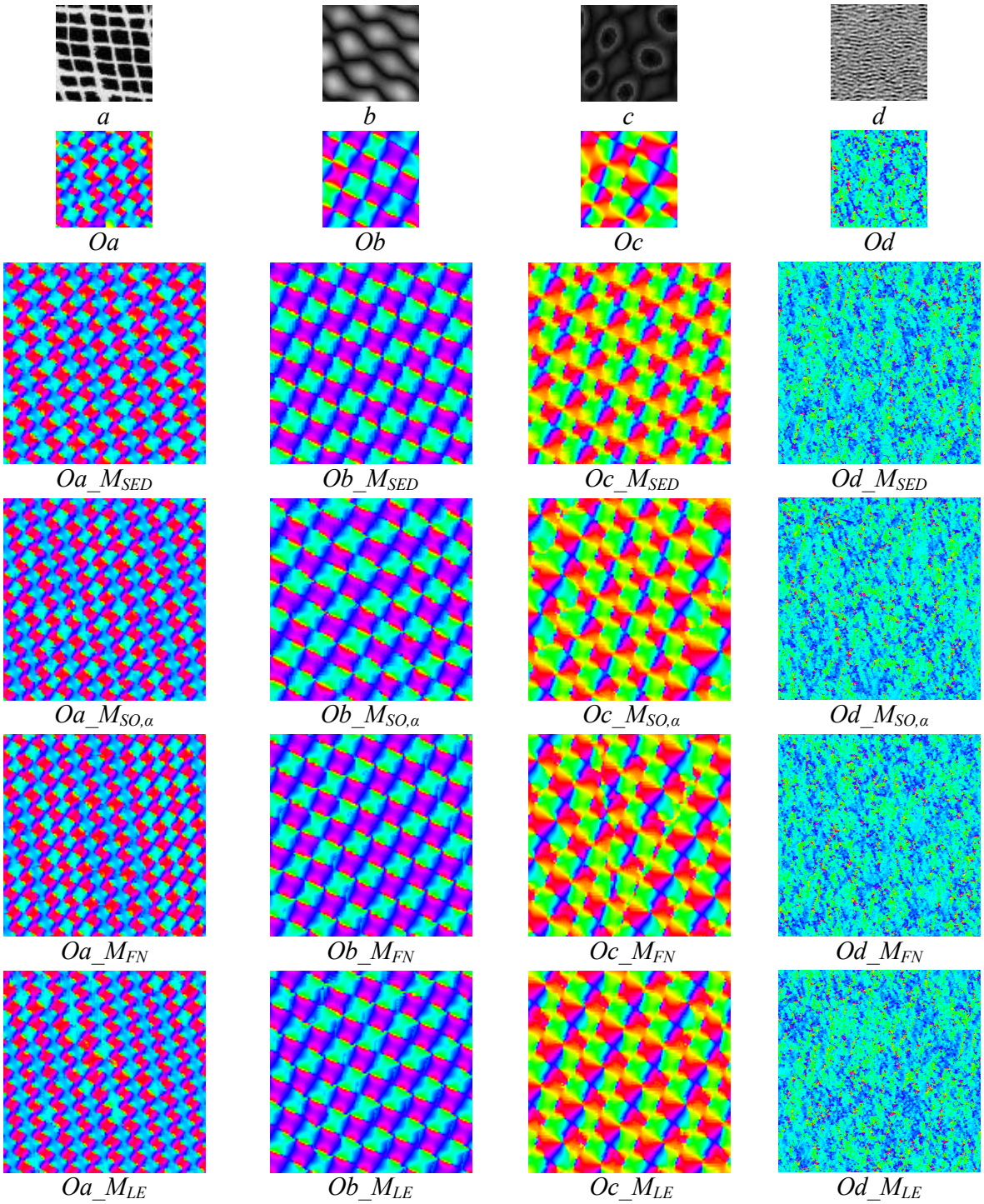


Figure 22 – *Résultats de synthèse du champ de tenseurs de structure avec différentes métriques de l'espace tensoriel. Chaque colonne montre (de haut en bas) : l'échantillon d'entrée, l'image d'orientation du champ de tenseurs de structure d'entrée, l'orientation du champ de tenseurs de structure obtenue en utilisant le carré de la distance Euclidienne, la métrique Shape-Orientation, la norme de Frobenius et la métrique Log-Euclidienne. Les images d'entrée a, b et c sont de taille 64×64 tandis que l'image d'entrée d est de taille 256×256. Les images de sortie de la 1ère, 2ème et 3ème colonne sont de taille 128×128 tandis que les images de sortie de la 4ème colonne sont de taille 512×512. Des pyramides Gaussiennes à 2 échelles sont utilisées avec un voisinage causal, un parcours lexicographique et 3 itérations.*

3.4.1.2 Complexité calculatoire

La métrique de l'espace tensoriel est le paramètre principal qui influe sur la complexité calculatoire et sur le temps total d'exécution du processus de synthèse (au moins 95% du temps de calcul complet). Aussi, nous présentons dans ce paragraphe une comparaison des temps de calcul correspondant aux différentes métriques tensorielles. Afin de réduire la charge de calcul, une accélération logicielle par parallélisation au moyen d'une carte graphique (GPU) a été mise en œuvre. Le Tableau 1 montre les temps de calcul observés pour la comparaison d'un couple de tenseurs au moyen des différentes métriques, avec (GPU, '*NVIDIA GeForce GTX690-4GD5, 1536 cores'*) ou sans parallélisation (CPU, '*Intel Core i7-4771, 3.50 GHz*'). La dernière ligne du tableau montre le gain apporté par le calcul GPU par rapport au calcul CPU.

Tableau 1 – *Temps d'exécution (en microsecondes) du CPU ou du GPU pour la comparaison de deux tenseurs en fonction de différentes métriques (Euclidienne, Shape-Orientation, Frobénius et Log-Euclidienne).*

	M_{SED}	$M_{SO,\alpha}$	M_{FN} / M_{LE}
CPU	169	2113	517
GPU	12.4	24.6	12
Gain	13.6	85.9	43.1

On peut remarquer dans le Tableau 1 que les coûts calculatoires de la métrique de Frobenius et de la métrique Log-Euclidienne sont toujours égaux. Ceci s'explique par le fait que le calcul du logarithme qui intervient dans la seconde métrique est effectué en amont du processus de synthèse et, par conséquent, est en pratique rendu négligeable du fait du nombre important de comparaisons et du nombre d'itérations effectuées ainsi que de la taille des voisinages utilisés. Les résultats obtenus par la version CPU montrent que la complexité calculatoire de la métrique *Shape-Orientation* est beaucoup plus importante que celle des autres. Dans le cas de calculs par la version GPU, il est à noter que la complexité calculatoire est relativement constante quelle que soit la métrique utilisée. Ceci s'explique par le fait que la gestion de la mémoire au sein du GPU (latences, accès concurrentiels, etc.) induite par la parallélisation consomme davantage de temps d'exécution que les calculs proprement dits. Dans tous les cas, la complexité calculatoire est très fortement réduite (de 13,6 à 85,9) de sorte que la version GPU conduit à un temps d'exécution au moins divisé par 10 pour le processus complet de synthèse.

3.4.2 Influence de la forme du voisinage, du type de parcours et du nombre d'itérations

Dans ce paragraphe, les influences de la forme du voisinage, du type du parcours choisi et du nombre d'itérations utilisés dans le processus de synthèse du champ de tenseurs sont étudiées en utilisant différents échantillons d'entrée.

Trois résultats obtenus avec trois textures différentes sont montrés aux Figures 23, 24 et 25, où les champs de tenseurs de structure des textures d'entrée (*a*), (*b*) et (*c*) sont représentés par leurs images d'orientations (*Oa*, *Ob* et *Oc*) et images de cohérence (*Ca*, *Cb* et *Cc*). Les images de la troisième ligne sont obtenues après une seule itération, les images de la quatrième ligne sont obtenues après trois itérations et ceux de la cinquième ligne sont obtenus après vingt itérations. Les champs d'orientation et de cohérence obtenus en utilisant un voisinage causal en forme de « L » associé à un parcours lexicographique sont à gauche, et ceux qui sont obtenus par un voisinage carré avec un parcours complètement aléatoire, sont à droite. La texture (*a*) est de taille 70×70, les textures (*b*) et (*c*) sont de taille 64×64 et les champs de tenseurs de structure de sortie sont de taille 128×128. Des voisinages de taille 15×15 sont utilisés avec des pyramides Gaussiennes à deux étages.

L'analyse des résultats de synthèse en utilisant la texture régulière structurée (*a*) à la Figure 23 montre qu'en traitant les premières lignes et colonnes du champ de tenseurs de structure de sortie, le voisinage causal contient des tenseurs bruités, ce qui explique la présence de motifs déformés dans la partie supérieure des images de sortie obtenues avec une seule itération (les images du côté gauche de *Oa1* et *Ca1*). Cependant, tous les voisinages restants de l'image de sortie sont sans bruit et comprennent entièrement des tenseurs déjà synthétisés. Par conséquent, trois itérations sont en général suffisantes pour fournir un champ de tenseurs bien synthétisé qui semble presque similaire à celui de l'entrée (images du côté gauche de *Oa3* et *Ca3*).

Le voisinage non-causal associé à un parcours complètement aléatoire nécessite un nombre d'itérations significativement plus élevé que le voisinage causal avec un balayage lexicographique, pour une qualité de résultat à peu près similaire; les images du côté droit de *Oa3* et *Ca3*, obtenues après trois itérations, présentent encore des motifs déformés et ceux de *Oa20* et *Ca20,* obtenues après vingt itérations, sont de bonne qualité. Ceci s'explique par le fait que la plupart des voisinages peuvent inclure des tenseurs bruités.

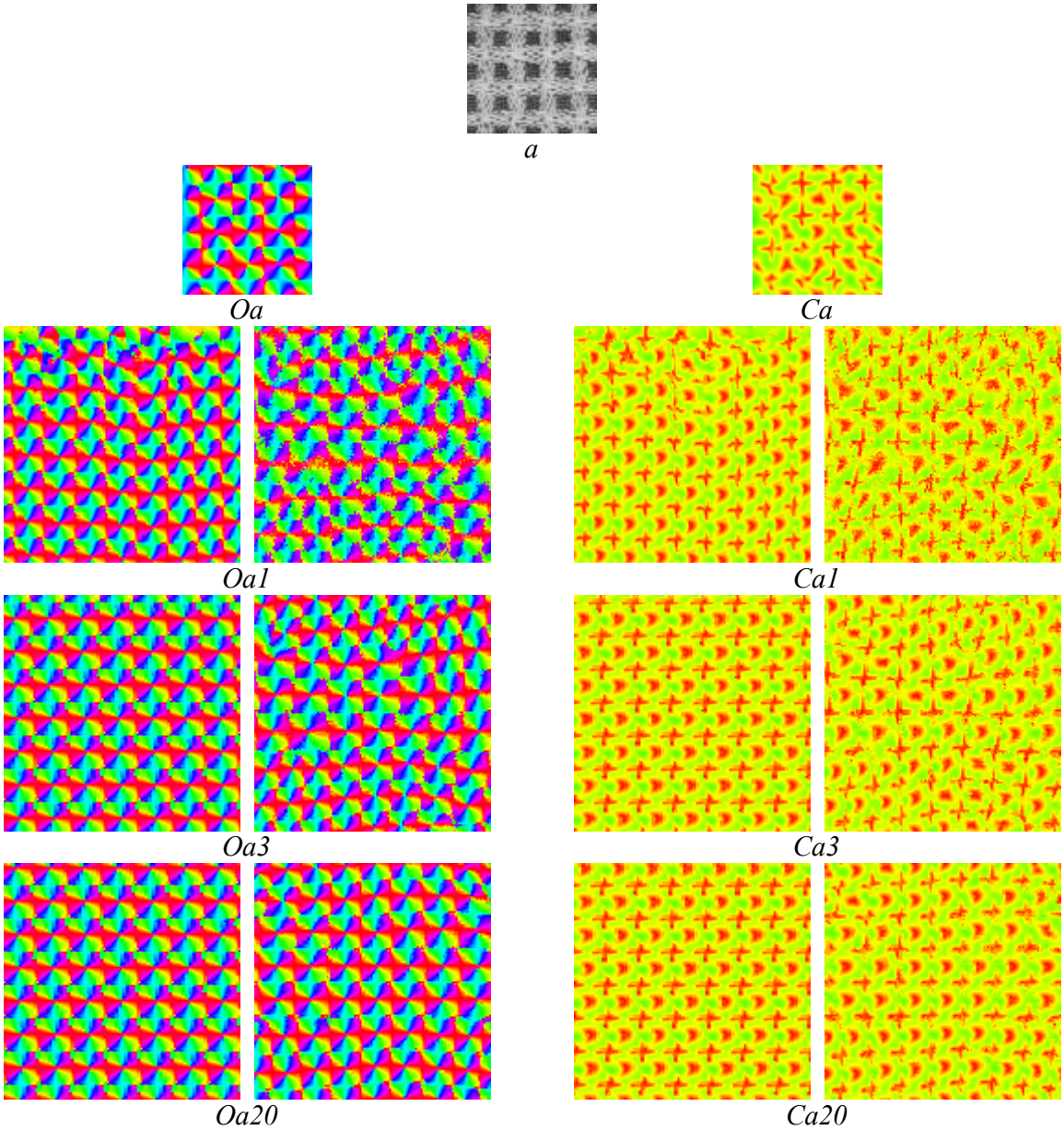


Figure 23 – *Influences de la forme du voisinage, du type de parcours et du nombre d'itérations sur la synthèse du champ de tenseurs de structure de la texture (a) représentée par ses images d'orientation (Oa) et de cohérence (Ca). La 3ème ligne est obtenue après 1 itération, la 4ème ligne est obtenue après 3 itérations et la 5ème ligne est obtenue après 20 itérations. Les images obtenues par un voisinage en forme de « L » avec un parcours lexicographique sont à gauche et ceux qui sont obtenues avec un voisinage carré et parcours aléatoire sont à droite. La texture (a) est de taille 70×70 et les champs de tenseurs de sortie sont de taille 128×128. Un voisinage de taille 15×15 est utilisé avec des pyramides Gaussiennes à 2 étages.*

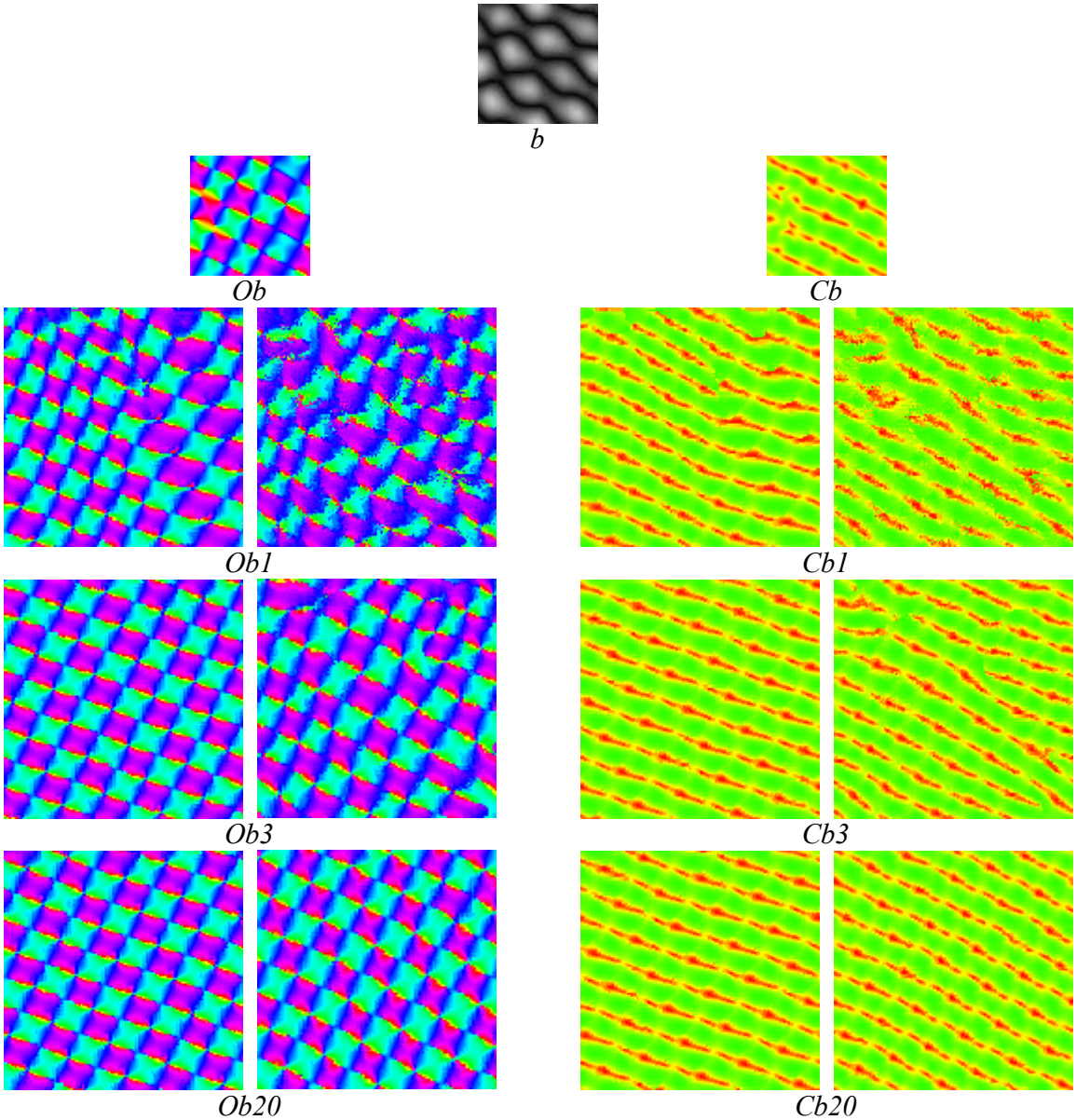


Figure 24 – *Influences de la forme du voisinage, du type de parcours et du nombre d'itérations sur la synthèse du champ de tenseurs de structure de la texture (b) représentée par ses images d'orientation (Ob) et de cohérence (Cb). La 3ème ligne est obtenue après 1 itération, la 4ème ligne est obtenue après 3 itérations et la 5ème ligne est obtenue après 20 itérations. Les images obtenues par un voisinage en forme de « L » avec un parcours lexicographique sont à gauche et ceux qui sont obtenues avec un voisinage carré et parcours aléatoire sont à droite. La texture (b) est de taille 64×64 et les champs de tenseurs de sortie sont de taille 128×128. Un voisinage de taille 15×15 est utilisé avec des pyramides Gaussienne à 2 étages.*

La même analyse s'applique aux résultats obtenus avec la texture (*b*), régulière et structurée (Figure 24), où trois itérations avec un voisinage causal associé à un parcours lexicographique donnent des images presque similaires à celles obtenues après vingt itérations avec un voisinage carré et un parcours aléatoire. On peut en conclure que le voisinage causal en forme de « L » est plus approprié pour la synthèse de textures régulières structurées.

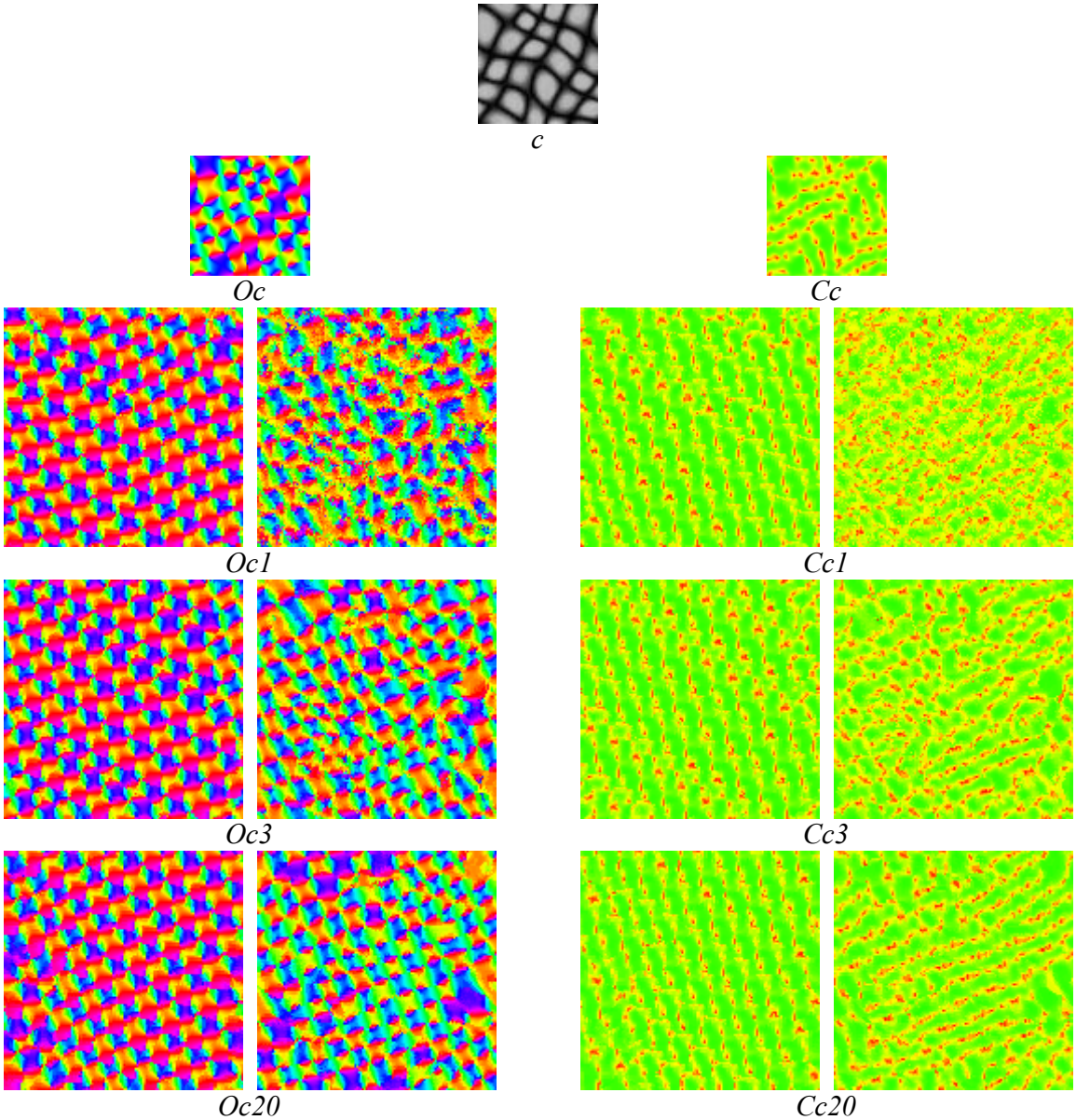


Figure 25 – *Influences de la forme du voisinage, du type de parcours et du nombre d'itérations sur la synthèse du champ de tenseurs de structure de la texture (c) représentée par ses images d'orientation (Oc) et de cohérence (Cc). La 3ème ligne est obtenue après 1 itération, la 4ème ligne est obtenue après 3 itérations et la 5ème ligne est obtenue après 20 itérations. Les images obtenues par un voisinage en forme de « L » avec un parcours lexicographique sont à gauche et ceux qui sont obtenues avec un voisinage carré et parcours aléatoire sont à droite. La texture (c) est de taille 64×64 et les champs de tenseurs de sortie sont de taille 128×128. Un voisinage de taille 15×15 est utilisé avec des pyramides Gaussienne à 2 étages.*

D'autre part, les résultats de la texture (*c*) (Figure 25) qui est moins régulière que les textures (*a*) et (*b*), montrent que le voisinage carré non-causal utilisé avec un parcours aléatoire converge vers une sortie de meilleure qualité que le voisinage causal en forme de « L » utilisé avec un balayage lexicographique (images du côté droit de *Oc20* et *Cc20*). Ce dernier ne parvient pas à reproduire les variations d'orientations présentes dans l'exemplaire et, par suite, aboutit à un champ de tenseurs synthétique plus régulier que le champ de

tenseurs d'entrée (images du côté gauche de *Oc20* et *Cc20*) et présentant des copies *verbatim* [XuG00] [Urs13], i.e. des répétitions de motifs quasi identiques.

3.4.3 Influence de la taille du voisinage

Outre le système de voisinage, les résultats de synthèse sont directement liés à la taille des voisinages utilisés qui doit être suffisamment grande pour capturer les motifs de la texture. Comme déjà mentionné, la synthèse multi-échelle permet d'éviter l'utilisation de grands voisinages et le nombre de niveaux de la pyramide a le même effet que la taille du voisinage. Par conséquent, cette partie présente l'influence de la taille du voisinage causal sur les champs de tenseurs de structure synthétisés.

Dans la Figure 26, les champs de tenseurs de structure des textures (*a*) et (*b*) sont représentés par leurs images d'orientations (*Oa* et *Ob*) et images de cohérence (*Ca* et *Cb*). Les résultats de synthèse du champ de tenseurs de structure de la texture (*a*) avec un voisinage causal de taille 5×5, 9×9 et 13×13 sont montrés à la troisième, quatrième et cinquième ligne, respectivement. De même pour la texture (*b*) en utilisant un voisinage causal de taille 17×17, 19×19 et 21×21. Les textures (*a*) et (*b*) sont de taille 64×64 et 96×96 respectivement et les champs de tenseurs de structure de sortie sont de taille 128×128. Pour les deux textures, l'approche mono-échelle est utilisée et les résultats sont obtenus après trois itérations. Pour chaque résultat, les images d'orientation et de cohérence sont montrées respectivement à gauche et à droite.

On peut observer dans la Figure 26 que le voisinage de tenseurs de structure de taille 5×5 n'est pas suffisamment large pour capturer les motifs de la structure de la texture (*a*), ce qui perturbe le champ de tenseurs de structure synthétique (voir les images *Oa_VS5* et *Ca_VS5*). Le même commentaire s'applique aux résultats obtenus en utilisant un voisinage de taille 9×9 avec toutefois une meilleure performance (voir *Oa_VS9* et *Ca_VS9*). En revanche, un voisinage de taille 13×13 est suffisant pour capturer les motifs de la structure et aboutit à des images de sortie satisfaisantes (*Oa_VS13* et *Ca_VS13*).

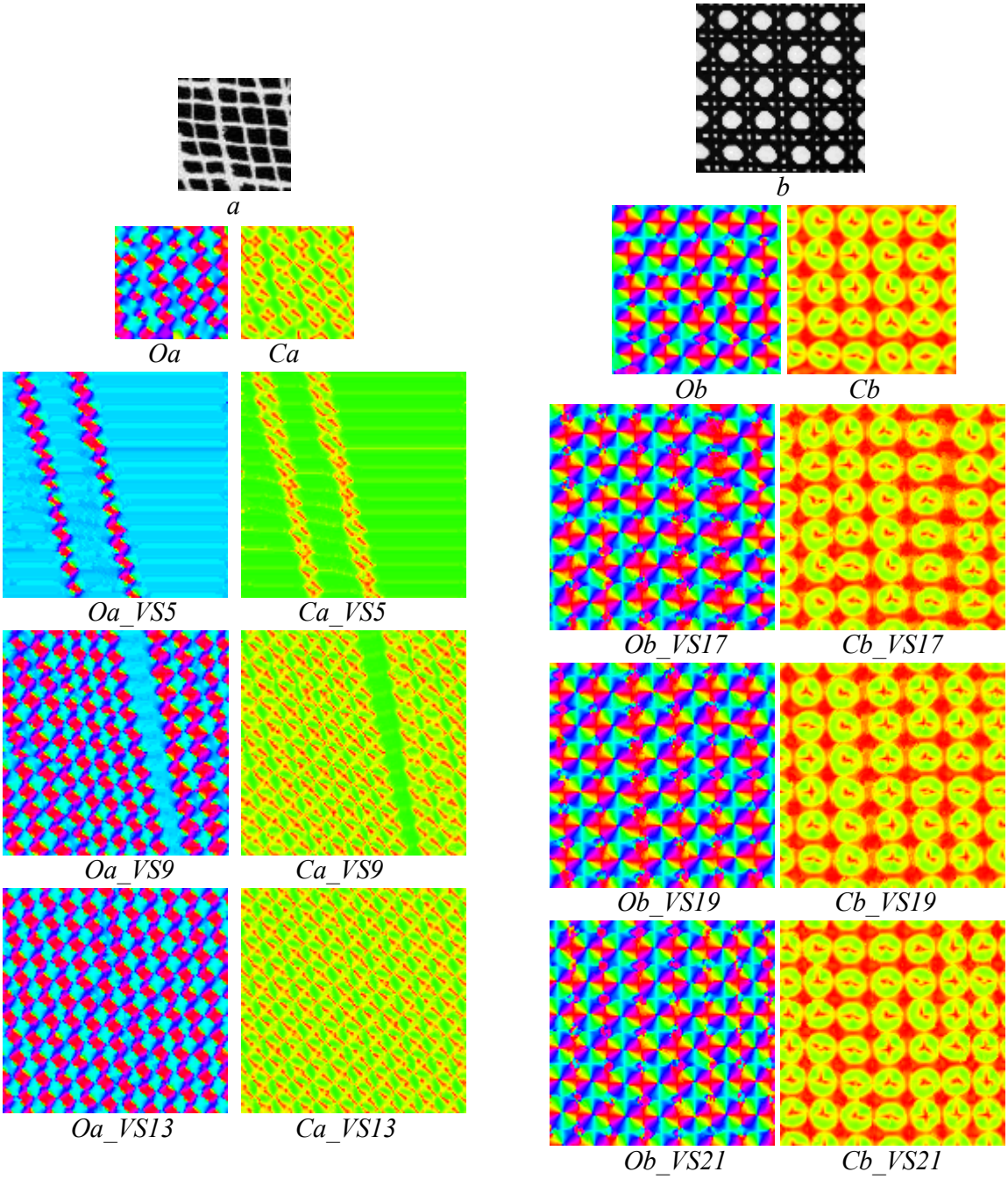


Figure 26 – *Influence de la taille du voisinage sur la synthèse de la couche de structure. Les champs de tenseurs des textures (a) et (b) sont représentés par leurs images d'orientations (Oa et Ob) et de cohérence (Ca et Cb). Les résultats de synthèse du champ de tenseurs de la texture (a) avec un voisinage de taille 5×5, 9×9 et 13×13 sont montrés à la 3ème, 4ème et 5ème ligne, respectivement. De même pour la texture (b) en utilisant un voisinage de taille 17×17, 19×19 et 21×21. (a) et (b) sont de taille 64×64 et 96×96 respectivement et les images de sortie sont de taille 128×128. Dans les deux résultats, des écarts-types de 1 sont utilisés pour la Gaussienne du gradient. Des écarts-types de 1.5 et 2 sont utilisés pour le noyau de pondération Gaussien avec la texture a et b, respectivement. L'approche mono-échelle est utilisée avec 3 itérations. Les images d'orientation sont montrées à gauche et les images de cohérence sont à droite.*

En outre, les images *Ob_VS17* et *Cb_VS17* montrent qu'un voisinage de taille 17×17 n'est pas suffisant pour pouvoir préserver la structure de la texture (*b*) car l'échelle de ces motifs est plus grande que celle de la texture (*a*). Les images obtenues avec un voisinage de

taille 19×19 ressemblent davantage aux images d'entrées, mais des motifs indésirables (des traces jaunes) existent toujours au milieu de l'image de cohérence (voir *Cb_VS19*). Dans ce cas, un voisinage de taille 21×21 est nécessaire pour bien reproduire la structure (voir *Ob_VS21* et *Cb_VS21*).

3.4.4 Réduction de la complexité calculatoire par interpolation

L'accélération de l'algorithme de synthèse du champ de tenseurs proposée consiste à commencer la synthèse de la pyramide par l'échelle à plus basse résolution et à synthétiser tous les étages de la pyramide jusqu'à l'étage *Ls* à partir duquel tous les autres étages sont construits par interpolation bilinéaire comme expliqué au paragraphe 3.3.5.

La Figure 27 montre des résultats de synthèse de différents champs de tenseurs de structure obtenus sans et avec accélération par interpolation bilinéaire. Chacun des résultats *a*, *b*, *c* et *d* montre, de haut en bas : les images d'orientation et de cohérence du champ de tenseurs de structure d'entrée, les orientations et cohérences du champ de tenseurs de structure obtenu par synthèse complète (i.e. sans accélération), et les orientations et cohérences du champ de tenseurs synthétisé en utilisant l'accélération par interpolation bilinéaire avec $Ls = 1$. Les images d'orientation sont montrées à gauche et les images de cohérences sont à droite. Des pyramides Gaussiennes à deux étages sont utilisées avec un voisinage causal de taille 9×9 pour les résultats *b* et *c* et un voisinage de taille 11×11 pour le résultat *a*. Des pyramides Gaussiennes à trois étages sont utilisées avec un voisinage de taille 9×9 pour le résultat *d*. Les champs de tenseurs d'entrée sont de taille 64×64 pour les résultats *a* et *c*, 70×70 et 128×128 pour les résultats *b* et *d*, respectivement. Les champs synthétisés dans *a*, *b*, et *c* sont de taille 128×128, tandis que le champ synthétisé dans *d* est de taille 256×256. Dans tous les résultats, la métrique Euclidienne (*SED*) est utilisée avec trois itérations.

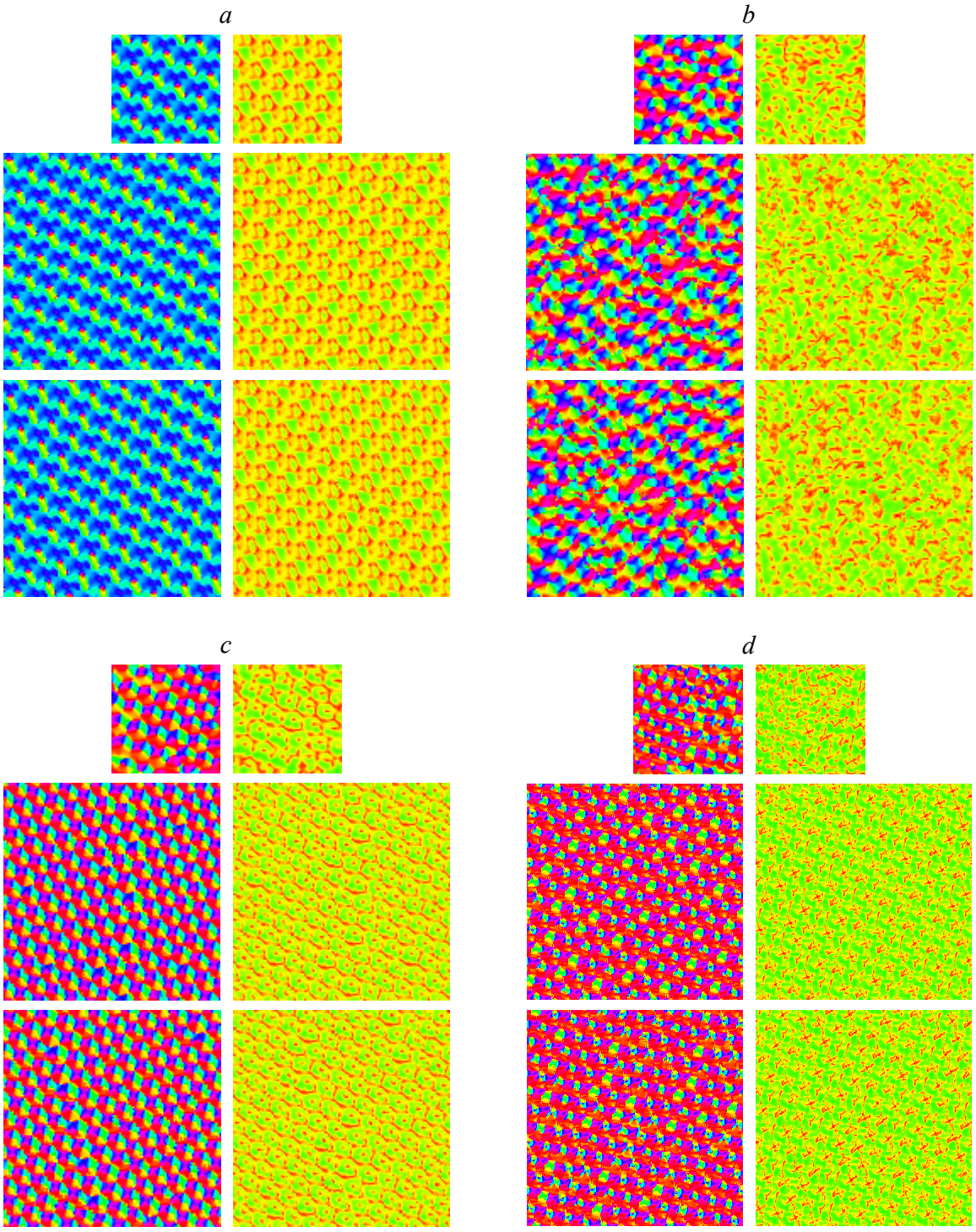


Figure 27 – *Résultats de synthèse de champs de tenseurs de structure sans et avec accélération par interpolation. Chaque résultat (de haut en bas) : les images d'orientation et de cohérence du champ de tenseurs d'entrée, les orientations et cohérences du champ de tenseurs obtenu par synthèse complète, et les orientations et cohérences du champ de tenseurs synthétisé en utilisant l'accélération par interpolation bilinéaire avec Ls = 1. Les images d'orientation sont montrées à gauche et les images de cohérences sont à droite. Des pyramides Gaussiennes à deux étages sont utilisées avec un voisinage de taille 9×9 pour les résultats b et c et un voisinage de taille 11×11 pour le résultat a. Des pyramides Gaussiennes à trois étages sont utilisées avec un voisinage de taille 9×9 pour le résultat d.*

La Figure 28 montre les mêmes résultats que ceux de la Figure 27.*d* mais obtenus en arrêtant la synthèse au niveau le plus haut (ayant la plus basse résolution) de la pyramide, dans l'algorithme accéléré. De haut en bas : les images d'orientation et de cohérence du champ de tenseurs d'entrée, les orientations et cohérences du champ de tenseurs obtenu par synthèse complète, et les orientations et cohérences du champ de tenseurs synthétisé en utilisant l'accélération par interpolation bilinéaire avec $Ls = 2$.

d'

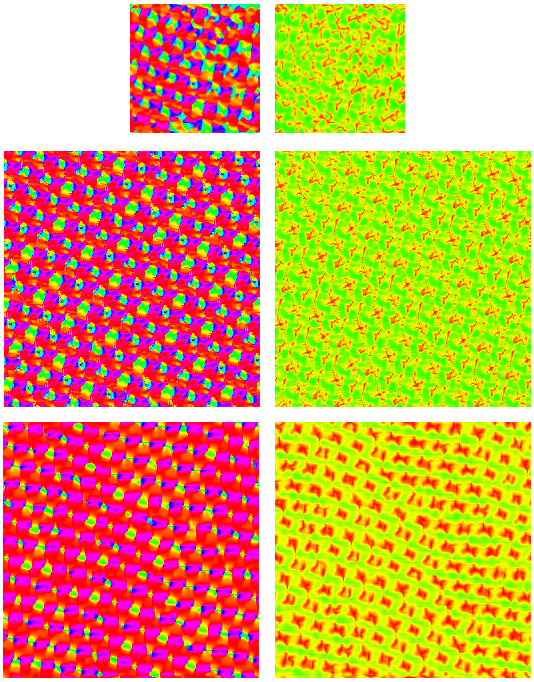


Figure 28 – *Résultat de synthèse d'un champ de tenseurs de structure avec accélération par interpolation. De haut en bas : les images d'orientation et de cohérence du champ de tenseurs d'entrée, les orientations et cohérences du champ de tenseurs obtenu par synthèse complète, et les orientations et cohérences du champ de tenseurs synthétisé en utilisant l'accélération par interpolation bilinéaire avec Ls = 2. Des pyramides Gaussiennes à trois étages sont utilisées avec un voisinage de taille 9×9.*

Le Tableau 2 présente le temps de calcul des résultats des Figures 27 et 28. La deuxième ligne montre le temps de calcul en utilisant l'approche de synthèse du champ de tenseurs de structure classique tandis que la troisième ligne montre le temps de calcul de l'approche de synthèse accélérée par interpolation bilinéaire, en utilisant un processeur '*Intel Core i7-2670QM, 2.20 GHz*', sans parallélisation.

Tableau 2 – *Temps de simulation (en secondes) pour les images des Figures 27 et 28.*

Résultat:	*a*	*b*	*c*	*d*	*d'*
Synthèse complète sans accélération	32	38	24	1013	1013
Synthèse accélérée par interpolation	12	16	10	385	45

On peut voir clairement, dans tous les résultats de la Figure 27, que le champ de tenseurs de structure synthétique obtenu par l'algorithme de synthèse classique est de bonne qualité et ressemble au champ de tenseurs d'entrée. En outre, les images obtenues en utilisant l'interpolation bilinéaire avec $Ls = 1$ sont à peu près similaires à celles générées par l'approche non-accélérée. Le tenseur de structure interpolé reste de bonne qualité en termes de structure et de préservation de la dynamique, tout en réduisant le temps de simulation (Tableau 2). En effet, bien que la qualité des images synthétiques reste intacte, les temps de calcul associés sont énormément réduits. Par exemple l'algorithme classique de synthèse prend 1013 secondes pour générer le champ de tenseurs de structure dans le résultat *d*, tandis que l'algorithme accéléré avec $Ls = 1$ n'exige que 385 secondes, ce qui indique une réduction de la complexité calculatoire de plus que 60%. On peut remarquer d'après la 6ème colonne du Tableau 2 qu'une réduction de la complexité calculatoire de 95% peut être atteinte en limitant la synthèse au plus haut niveau de la pyramide et en appliquant ensuite deux interpolations successives. Cependant, ceci vient au détriment de la qualité des champs de tenseurs obtenus (voir 3ème ligne de la Figure 28). Pour cette texture, la synthèse avec $Ls = 2$ peine notamment à respecter les structures du champ de tenseurs d'entrée, ce qui s'explique sans doute par les lissages successifs appliqués pour construire la pyramide Gaussienne. En d'autres termes, les informations de structure ne sont pas facilement reproduites par interpolations successives.

3.5 Conclusion

Comme illustré au paragraphe 2.4.2, la première étape de l'approche de synthèse structure/texture proposée consiste à synthétiser la couche de structure représentée par le champ de tenseurs de structure issu de la texture exemplaire. Dans un premier temps, ce chapitre a introduit le champ de tenseurs de structure et les informations d'orientation locale et de cohérence qui en sont extraits. La stratégie de synthèse du champ de tenseurs de structure par adaptation de l'algorithme de Wei et Levoy aux spécificités des champs de tenseurs de structure est détaillée dans un second temps.

Pour la comparaison de deux tenseurs, nous avons choisi d'utiliser quatre métriques différentes, dont trois (la métrique Euclidienne, la norme de Frobenius et la métrique Log-Euclidienne [Fil05]) sont calculées directement sur les composantes du tenseur alors que la quatrième métrique (*Shape-Orientation* [Tou10]) est basée sur la géométrie (orientation et cohérence) du tenseur. Concernant le système de voisinage, un voisinage causal en forme de « L » associé à un parcours lexicographique ainsi qu'un voisinage carré non-causal avec un parcours complètement aléatoire sont mis en œuvre. Afin d'éviter l'utilisation de larges voisinages, couteuse en termes de complexité calculatoire, l'approche multi-résolution par décomposition en pyramides Gaussiennes est utilisée. En outre, l'interpolation bilinéaire est associée à cette approche multi-échelle pour remplacer la synthèse exhaustive de l'étage avec la plus haute résolution de la pyramide.

Dans le cadre des résultats de synthèse du champ de tenseurs, une comparaison des différentes métriques de dissimilarité de tenseurs a montré qu'aucune métrique ne se détache clairement. En outre, par calcul GPU, ces différentes métriques aboutissent presque à la même complexité calculatoire. Une étude comparative sur le système de voisinage et le nombre d'itérations de synthèse a également montré que le voisinage causal associé à un balayage lexicographique est plus approprié pour la synthèse de textures régulières structurées. Le voisinage carré avec un parcours aléatoire peut toutefois remédier à l'excès de régularité des champs synthétiques obtenu en utilisant le voisinage causal. La nécessité d'un voisinage de taille suffisante à capturer les motifs de la structure est également établie. Finalement, l'interpolation bilinéaire a montré son aptitude à réduire le temps de calcul, sans affecter la qualité du champ de tenseurs synthétisé.

Chapitre 4

Approche de synthèse structure/texture

Sommaire

4.1 Introduction 63
4.2 Synthèse de texture contrainte par champ de tenseurs 63
4.2.1 Algorithme de synthèse 63
4.2.2 Synthèse multi-échelle 66
4.3 Résultats 68
4.3.1 Efficacité de la synthèse de la couche de structure et influence de la pondération structure/texture 68
4.3.2 Evaluation objective de la qualité de synthèse 70
4.3.3 Influence de la taille des voisinages de synthèse 71
4.3.4 Comparaison de la méthode proposée avec les approches existantes 73
4.4 Conclusion 83

4.1 Introduction

Dans ce chapitre, nous présentons une approche pour la synthèse d'une texture sous la contrainte d'un champ de tenseurs de structure. Il s'agit de la deuxième étape de l'algorithme de synthèse structure/texture, évoqué au paragraphe 2.4.2, consistant à synthétiser dans un premier temps un champ de tenseurs de structure à partir de celui de l'exemplaire (cf. chapitre 3), puis à s'appuyer sur ce champ pour synthétiser la texture finale.
Pour cette seconde étape, comme au chapitre 3, l'algorithme de synthèse de texture contrainte par champ de tenseurs de structure sera inspiré de l'algorithme de Wei et Levoy [Wei00] dont il sera une adaptation.

Nous mettrons en évidence, au travers d'un ensemble d'expérimentations, l'intérêt de s'appuyer sur la couche de structure pour la synthèse de texture, en termes de capacité à reproduire la structure et notamment les variations d'orientations de l'échantillon d'entrée. Pour mesurer la qualité des textures synthétisées, plus précisément la ressemblance des intensités et des orientations des textures d'entrée et de sortie, des indices de qualité objectifs sont proposés. L'effet de la taille des voisinages de synthèse est également étudié. Enfin, l'algorithme de synthèse structure/texture proposé est comparé à cinq autres méthodes de synthèse issues de la littérature.

4.2 Synthèse de texture contrainte par champ de tenseurs

4.2.1 Algorithme de synthèse

Comme illustré à la Figure 29, l'algorithme de synthèse de texture contrainte par champ de tenseurs prend comme entrées l'exemplaire I_{in}, son champ de tenseurs de structure S_{in}, et le champ de tenseurs de structure S_{out} préalablement synthétisé à partir de S_{in}. La texture de sortie I_{out} est initialisée par un bruit aléatoire qui est modifié durant le processus de synthèse de sorte à produire une texture semblable à I_{in}. Le processus de synthèse, qui peut être itéré plusieurs fois, est similaire à celui de l'algorithme de W&L original : pour chaque pixel de I_{out}, il s'agit d'identifier le pixel de I_{in} dont le voisinage est le plus ressemblant. Cependant, le voisinage considéré doit prendre en compte les informations supplémentaires fournies par le champ de tenseurs de structure préalablement synthétisé. Chaque voisinage se décompose donc en deux composantes : un voisinage à valeurs d'intensités VI dans les images de texture (I_{in} et I_{out}), ainsi qu'un voisinage de tenseurs de structure VS dans les

images de structure (S_{in} et S_{out}). La Figure 30 montre la structure d'un voisinage de synthèse causal de taille 5×5 d'un site z, i.e. une combinaison d'un voisinage de tenseurs de structure $VS(z)$ dans un champ de tenseurs S et un voisinage à valeurs d'intensités $VI(z)$ dans une image I.

Conformément à la convention utilisée pour désigner le voisinage de tenseurs de structure, on appelle « voisinage à valeurs d'intensités » au site z d'une image I, l'ensemble $VI(z) = \{I(t), t \in N_K(z)\}$ où $N_K(z)$ est un ensemble de K sites connexes à z. Soit $VI_{in}(z)$ et $VI_{out}(z')$ deux voisinages à valeurs d'intensités dans les images d'entrée et de sortie. On définit la somme des erreurs quadratiques (*Sum Square Distance - SSD*) entre $VI_{in}(z)$ et $VI_{out}(z')$ comme étant égale au carré de la distance Euclidienne entre les vecteurs des composantes du voisinage :

$$SSD\big(VI_{in}(z), VI_{out}(z')\big) = \sum_{k=1}^{K} (I_{in}(t_k) - I_{out}(t'_k))^2, \tag{15}$$

où K est le nombre de pixels composant chaque voisinage.

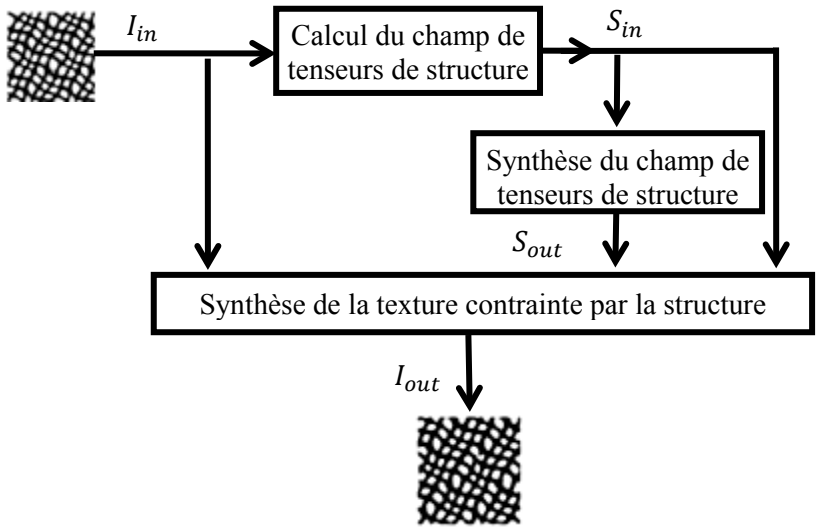


Figure 29 – *Illustration de l'approche de synthèse structure/texture proposée.*

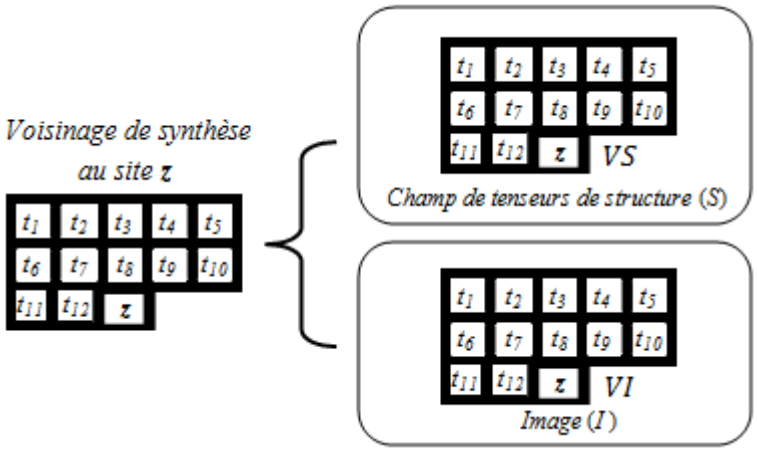


Figure 30 – *Structure d'un voisinage de synthèse en un site z.*

La métrique utilisée pour mesurer la dissimilarité entre les voisinages combine la somme des valeurs de dissimilarité des tenseurs de structure (*Sum of Structure Tensor Dissimilarity – SSTD*) définie dans l'équation (13) pour la composante « voisinage de tenseurs de structure » et la somme des erreurs quadratiques (*Sum Square Distance - SSD*) pour la composante « voisinage à valeurs d'intensités » définie dans l'équation (15). Un poids est affecté à chaque composante comme suit :

$$\gamma(z,z') = p \cdot SSD\big(VI_{in}(z), VI_{out}(z')\big) + (1-p) \cdot SSTD\big(VS_{in}(z), VS_{out}(z')\big), \qquad (16)$$

où $VI_{in}(z)$ et $VI_{out}(z')$ sont les composantes des voisinages de niveaux de gris (i.e. voisinages à valeurs d'intensités) des sites z et z' dans I_{in} et I_{out} respectivement, $VS_{in}(z)$ et $VS_{out}(z')$ sont les composantes des voisinages de tenseurs de structure des sites z et z' dans S_{in} et S_{out} respectivement, et p est un facteur de pondération ($0 \leq p \leq 1$ quand *SSD* et *SSTD* sont normalisés).

Notons que, pour $p = 1$, l'information de structure n'est pas prise en compte et une synthèse classique W&L est opérée. Au contraire, pour $p = 0$, le choix du pixel de sortie est basé uniquement sur l'information de structure et l'information d'intensité est ignorée.

Du fait de la prise en compte de l'information supplémentaire fournie par la couche de structure, la reproduction des motifs de l'exemplaire lors de la synthèse de la texture est réalisable en utilisant un voisinage plus petit que celui retenu pour la synthèse de la structure. Ce point sera illustré ultérieurement.

La Figure 31 résume l'algorithme de synthèse de texture contrainte par champ de tenseurs de structure.

Entrée: I_{in} , S_{in} , S_{out}

I_{out} ← INITIALISATIONBRUITINTENSITE (I_{in})

Tant que iter ≤ nbr_itérations

pour chaque position $z' = (x, y)$ de I_{out}

Extraire $VI_{out}(z')$ et $VS_{out}(z')$

$z_{min} \leftarrow argmin_z\{\gamma(z, z')\}$

$I_{out}(z') \leftarrow I_{in}(z_{min})$

fin pour

Incrémenter iter

Fin tant que

Sortie: I_{out}

Figure 31 – *Pseudocode de l'algorithme de synthèse de texture contrainte par champ de tenseurs de structure.*

4.2.2 Synthèse multi-échelle

Des pyramides multi-résolution de tenseurs de structure ont été utilisées dans l'étape de synthèse de la couche de structure (cf. paragraphe 3.3.4). Similairement, dans l'étape de synthèse de la couche de texture, des pyramides de tenseurs de structure ainsi que des pyramides de textures sont mises en œuvre. Ainsi, des pyramides Gaussiennes sont construites par lissage et sous-échantillonnage par un facteur 2:1. En outre, le voisinage multi-résolution à valeurs d'intensités $VI_L^{mr}(z)$ au site $z = (x, y)$ du niveau L de la pyramide est défini par la réunion du voisinage $VI_L(z)$ au même niveau ainsi que le voisinage $VI_{L+1}(z_{\downarrow})$ du site correspondant z↓=(x/2, y/2) au niveau L+1 :

$$VI_L^{mr}(z) = VI_L(z) \cup VI_{L+1}(z_{\downarrow}) , \tag{17}$$

où $VI_L(z) = \{I_L(t), t \in N_L(z)\}$ et $VI_{L+1}(z_{\downarrow}) = \{I_{L+1}(t_{\downarrow}), t_{\downarrow} \in N_{L+1}(z_{\downarrow})\}$.

La Figure 32 résume l'algorithme de synthèse de texture multi-échelle contrainte par le champ de tenseurs de structure. Les voisinages multi-résolution d'entrée $VI_{in}^{mr}(z)$ et $VS_{in}^{mr}(z)$ correspondent aux pyramides Gaussiennes de texture et de tenseur de structure d'entrée PI_{in} et PS_{in} construites respectivement à partir de I_{in} et S_{in}. De façon similaire, les voisinages multi-résolution de sortie $VI_{out}^{mr}(z')$ et $VS_{out}^{mr}(z')$ correspondent aux pyramides

Gaussiennes de texture et de tenseur de structure de sortie PI_{out} et PS_{out} construites respectivement à partir de I_{out} et S_{out}. La mesure de dissimilarité entre voisinages multi-résolution est définie par :

$$\gamma^{mr}(z,z') = p \cdot SSD\left(VI_{in}^{mr}(z), VI_{out}^{mr}(z')\right) + (1-p) \cdot SSTD\left(VS_{in}^{mr}(z), VS_{out}^{mr}(z')\right) \quad (18)$$

<u>Entrée</u>: I_{in} , S_{in} , S_{out}

PI_{in} ← DECOMPOSITIONPYRAMIDEGAUSSIENNE (I_{in})
PS_{in} ← DECOMPOSITIONPYRAMIDEGAUSSIENNE (S_{in})
PS_{out} ← DECOMPOSITIONPYRAMIDEGAUSSIENNE (S_{out})
I_{out} ← INITIALISATIONBRUITINTENSITE (I_{in})
PI_{out} ← DECOMPOSITIONPYRAMIDEGAUSSIENNE (I_{out})

Tant que iter ≤ nbr_itérations
  pour chaque niveau L de $NL - 1$ à 0
    pour chaque position $z' = (x, y)$ de PI_{out}^{L}
      Extraire $VI_{L}^{mr}(z')_{out}$ et $VS_{L}^{mr}(z')_{out}$
      $z_{min} \leftarrow argmin_{z}\{\gamma^{mr}(z,z')\}$
      $PI_{out}^{L}(z') \leftarrow PI_{in}^{L}(z_{min})$
    fin pour
  fin pour
  Incrémenter iter
Fin tant que

<u>Sortie</u>: PI_{out}^{0}

Figure 32 – *Pseudocode de l'algorithme de synthèse de texture multi-échelle contrainte par champ de tenseurs de structure.*

Par conséquent, le voisinage de synthèse multi-résolution au niveau L, dans les pyramides d'entrée et de sortie, inclut les voisinages à valeurs d'intensités et les voisinages de tenseurs de structure au même niveau L, ainsi que les voisinages à valeurs d'intensités et les voisinages de tenseurs de structure des positions correspondantes au niveau L+1 précédemment synthétisé.

Dans le processus de synthèse de la couche de texture, il faut noter que tous les étages de la pyramide de tenseurs de structure de sortie (PS_{out}) et de la pyramide de texture de sortie (PI_{out}) sont traités périodiquement comme expliqué au paragraphe 3.3.7. Des exemples de textures synthétiques périodiques sont donnés en Annexe D. Comme dans le cas des champs de tenseurs de structure, des distorsions aux bords peuvent apparaître si la taille des motifs n'est pas compatible avec la périodicité de l'image.

Comme déjà mentionné, la taille des voisinages utilisés pour la synthèse de la couche de texture est généralement plus petite que celle des voisinages utilisés pour la synthèse du champ de tenseurs de structure. Cependant, les meilleures performances de l'algorithme proposé s'accompagnent toujours d'une charge de calcul supplémentaire nécessaire pour synthétiser la couche de structure. Ce constat a motivé l'accélération de l'algorithme de synthèse de la couche de structure qui est détaillée au chapitre précédent. En revanche, une synthèse complète pour les étages des pyramides Gaussiennes est utilisée dans l'étape de synthèse de la couche de texture, sans aucune interpolation, afin d'éviter tout effet lié à la sous-optimalité d'une telle accélération, en termes de respect de la dynamique et de la structure de l'échantillon d'entrée.

4.3 Résultats

Cette partie illustre les résultats de l'algorithme de synthèse à deux étapes proposé. L'impact du paramètre de pondération structure/texture sur la qualité de la texture synthétisée est étudié. Nous proposons l'utilisation de quatre mesures objectives pour évaluer les résultats de synthèse. Il s'agit de statistiques d'ordre 1 et 2 des intensités et des orientations locales. L'influence de la taille des voisinages de synthèse est étudiée et analysée. Finalement, l'algorithme proposé est évalué et comparé à l'algorithme non-paramétrique classique de W&L [Wei00], à l'approche paramétrique de Portilla et Simoncelli (P&S) [Por00], à la méthode à base de *patch* de Efros et Leung (E-L) [Agu13] ainsi qu'aux algorithmes de Tartavel et al. (Tar) [Tar14] et Galerne et al. (Gal) [Gal11] en utilisant différentes textures d'entrée.

4.3.1 Efficacité de la synthèse de la couche de structure et influence de la pondération structure/texture

Afin de montrer la pertinence de la synthèse préalable de la couche de structure, la Figure 33 présente des résultats de synthèse de l'algorithme proposé en utilisant différentes

valeurs pour le facteur de pondération p dans l'équation (16) : 0, 0.35 et 1. La première colonne illustre, de haut en bas, l'échantillon d'entrée avec l'orientation (gauche) et la cohérence (droite) de son champ de tenseurs de structure, l'orientation et la cohérence du champ de tenseurs de structure synthétisé. Les colonnes suivantes montrent la texture synthétisée (ligne 1), l'orientation (ligne 2) et la cohérence (ligne 3) du champ de tenseurs de structure de cette texture, pour p valant 0, 0.35 et 1 (respectivement pour les colonnes 2, 3 et 4). La texture d'entrée est de taille 64×64 et les textures de sortie sont de taille 128×128. Des pyramides Gaussiennes à deux étages, des voisinages causaux de taille 15×15 pour la synthèse du tenseur de structure et 13×13 pour la synthèse de la texture, un parcours lexicographique et la norme de Frobenius sont utilisés. Les algorithmes de synthèse de la structure et de synthèse de la texture sont respectivement itérés trois et deux fois. Des résultats complémentaires obtenus pour des valeurs différentes de p sont fournis en Annexe E.

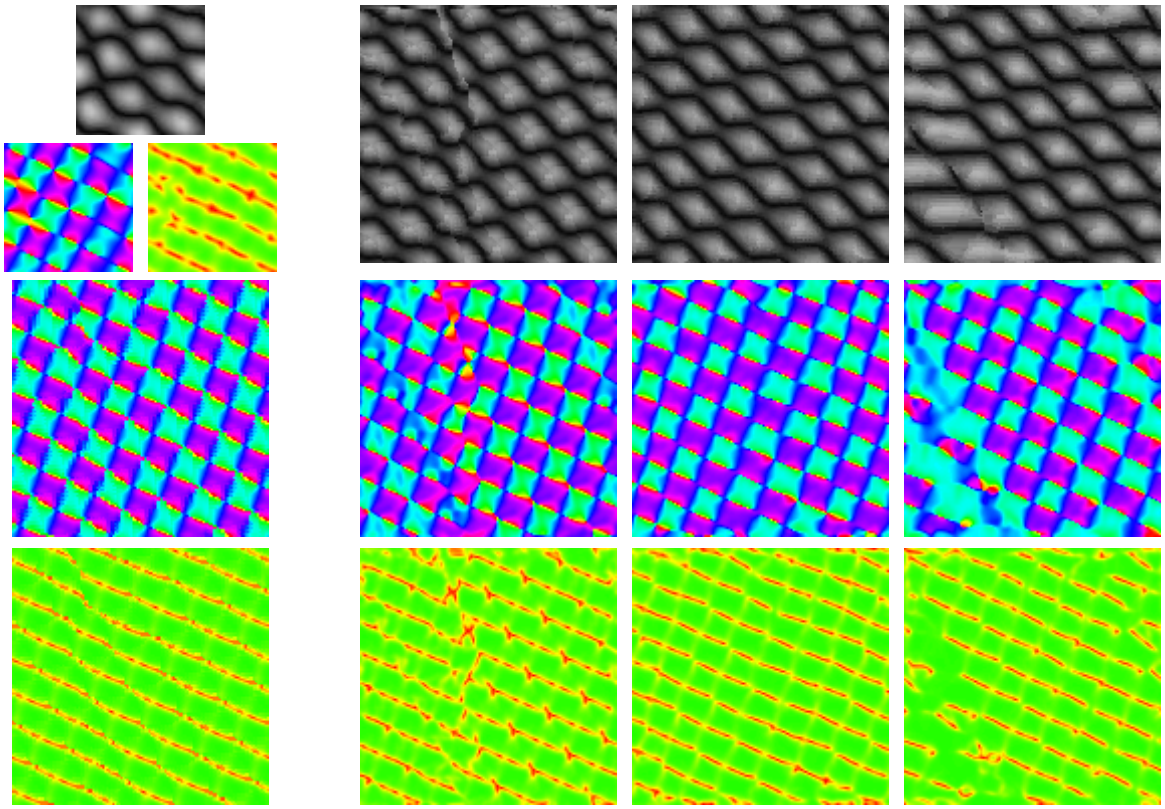


Figure 33 – *Résultats de synthèse pour différentes valeurs du paramètre de pondération* p *dans l'équation (16). 1ère colonne (de haut en bas) : la texture d'entrée avec l'orientation et la cohérence du champ de tenseurs de structure d'entrée, l'orientation et la cohérence du champ de tenseurs de structure synthétisé. 2ème, 3ème et 4ème colonne (de haut en bas) : la texture synthétisée par l'algorihtme de synthèse proposé et son orientation et cohérence avec respectivement* $p = 0$, $p = 0.35$ *et* $p = 1$. *Des pyramides Gaussiennes à 2 étages, des voisinages causaux de taille 15×15 pour la synthèse du tenseur de structure et 13×13 pour la synthèse de la texture, un parcours lexicographique et la norme de Frobenius sont utilisés.*

On peut noter que la synthèse W&L pure (i.e. $p = 1$, l'information de structure n'étant pas prise en compte) ne parvient pas à reproduire correctement la variation d'orientations de l'échantillon d'entrée (cf. quatrième colonne). Dans le cas où l'information d'intensité est ignorée ($p = 0$), le pixel de sortie est basé uniquement sur l'information de structure, et la texture synthétisée semble plus régulière que l'exemplaire montrant en outre des artefacts indésirables (cf. deuxième colonne). Au contraire, avec $p = 0.35$, l'approche proposée réussit à reproduire les variations d'orientations de l'exemplaire et aboutit à une texture de sortie de meilleure qualité (cf. troisième colonne).

Cet exemple montre que la texture de sortie peut être mieux synthétisée en prenant en considération la couche de structure en supplément de l'information d'intensité.

4.3.2 Evaluation objective de la qualité de synthèse

En complément d'une évaluation qualitative, par nature subjective, une analyse quantitative des résultats est essentielle. Aussi, un ensemble de mesures objectives qui compare les caractéristiques des textures d'entrée et de sortie est proposé. Les textures synthétisées sont donc évaluées en utilisant des statistiques du 1er et du 2ème ordre obtenues à partir du champ d'intensité et du champ des orientations locales : les histogrammes [Urs12] mais aussi les coefficients d'autocorrélation des intensités et des orientations locales permettent d'évaluer la ressemblance entre les structures (i.e. la morphologie des motifs) des textures d'entrée et de sortie.

La définition de l'autocorrélation d'orientations locales n'étant pas triviale, nous nous référons à la formulation complexe suivante. Soit Z_u la formulation complexe de l'orientation au site u exprimée par :

$$Z_u = C_u e^{j2O_u} \tag{19}$$

où C_u et O_u désignent respectivement le facteur de cohérence et le facteur d'orientation au site u.

Dans ce mémoire, l'autocorrélation des orientations locales avec un déplacement w est calculée comme suit :

$$\emptyset_z(w) = \|\langle Z_u \overline{Z_{u+w}} \rangle\|, \tag{20}$$

où $\langle . \rangle$ désigne la moyenne et $\|c\|$ représente le module du nombre complexe c (la formulation utilisée est détaillée en Annexe F).

4.3.3 Influence de la taille des voisinages de synthèse

Comme déjà mentionné, la taille du voisinage de synthèse du champ de tenseurs de structure a un impact sur la qualité du résultat (cf. paragraphe 3.4.3). La taille du voisinage de synthèse de la texture affecte également la qualité de la texture synthétisée.

Nous présentons à la Figure 34 l'influence conjointe des tailles de voisinages utilisés pour la couche de structure *VS* et pour la couche de texture *V_I_S* (i.e. combinant le voisinage à valeurs d'intensité et le voisinage de tenseurs de structure). Les résultats de synthèse du champ de tenseurs de structure sont les mêmes qu'à la Figure 26.*a*. La première ligne illustre, de gauche à droite, l'échantillon d'entrée, son image d'orientation, les images d'orientation des champs de tenseurs de structure synthétisés en utilisant des voisinages de structure de tailles respectivement égales à 5×5, 9×9 et 13×13. Les tailles des voisinages utilisés pour les couches de texture et de structure n'étant pas nécessairement identiques, les textures synthétisées en utilisant les contraintes de structure de la première ligne sont montrées à la deuxième ligne en utilisant un voisinage de texture de taille 5×5, à la troisième ligne avec un voisinage de taille 7×7, et à la quatrième ligne avec un voisinage de texture 9×9. La première colonne montre les résultats correspondants obtenus avec une synthèse W&L seule. En haut à gauche de la figure, est fournie la représentation des tailles 5×5, 9×9 et 13×13 à l'échelle de la texture d'entrée. L'approche mono-échelle est utilisée et les résultats sont obtenus après trois itérations. La figure comportant les images d'orientation des textures synthétisées est donnée en Annexe G.

L'autocorrélation des orientations locales (cf. paragraphe 4.3.2) est utilisée pour évaluer la qualité des textures synthétisées. Le Tableau 3 présente la distance Euclidienne entre les coefficients d'autocorrélation des orientations locales considérés pour un déplacement $\mathrm{w} \in [\![-15; 15]\!] \times [\![-15; 15]\!]$, et calculés sur le champ de tenseurs de structure de la texture d'entrée et le champ de tenseurs de la texture synthétisée.

On peut remarquer dans la Figure 34 que le voisinage de tenseurs de structure de taille 5×5 n'est pas suffisamment important pour capturer les motifs de la structure d'entrée, et perturbe donc le champ de tenseurs de structure synthétique, ce qui aboutit à des textures désordonnées, indépendamment de la taille du voisinage de la texture utilisée. Ceci est prouvé par les valeurs élevées des différences d'autocorrélations d'orientation à la troisième colonne (*VS5*) du Tableau 3. Le même constat est établi pour les résultats obtenus en utilisant un

voisinage de structure de taille 9×9 avec une performance accrue, mise en évidence par la diminution significative de la distance Euclidienne (*VS9*). En revanche, un voisinage de structure de taille 13×13 est suffisant pour capturer les motifs de la structure et aboutir à des textures de sortie satisfaisantes avec de meilleurs résultats d'autocorrélation (*VS13*).

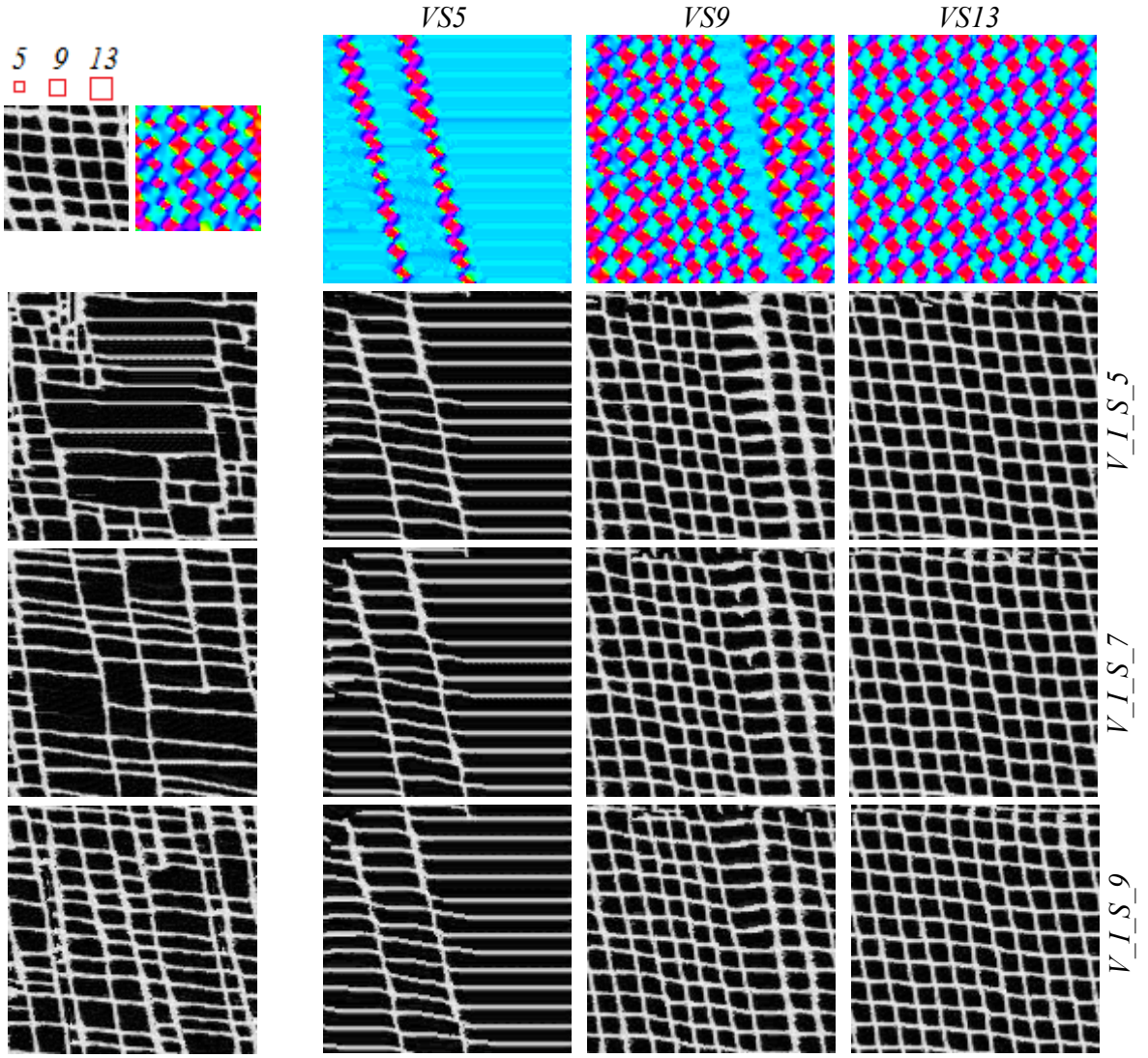


Figure 34 – *Influence de la taille du voisinage sur la synthèse des couches de structure et de texture. La 1ère ligne montre (de gauche à droite) l'échantillon d'entrée, son image d'orientation, les images d'orientation des champs de tenseurs synthétisés avec des voisinages de structure de taille 5×5, 9×9 et 13×13. Les textures synthétisées en utilisant les contraintes de la 1ère ligne sont montrées à la 2ème ligne en utilisant un voisinage de texture de taille 5×5, à la 3ème ligne avec un voisinage de taille 7×7 et à la 4ème ligne avec un voisinage de texture 9×9, tandis que la 1ère colonne montre les résultats obtenus par l'algorithme de W&L. En haut à gauche : représentation des tailles 5×5, 9×9 et 13×13 des voisinages à l'échelle de l'échantillon d'entrée.*

La synthèse basée uniquement sur les intensités, obtenue en faisant abstraction de la couche de structure, ne parvient pas à reproduire les motifs de l'exemplaire, quelle que soit la taille du voisinage (comme il est montré à la première colonne de la Figure 34). Ceci est

vérifié par les différences de la distance Euclidienne à la deuxième colonne (W&L) du Tableau 3.

Tableau 3 – *Distance Euclidienne entre les coefficients d'autocorrélation des orientations locales des résultats de la Figure 34.*

Distance Euclidienne	*W&L*	*VS5*	*VS9*	*VS13*
V_I_S_5	6.76	7.83	2.17	2.02
V_I_S_7	3.47	9.55	2.08	2.03
V_I_S_9	2.81	9.42	2.08	2.01

Dès lors que le champ de tenseurs de structure est correctement synthétisé et utilisé comme contrainte pour la synthèse de la couche de texture, la qualité des textures synthétiques ne s'améliore que de manière négligeable avec l'augmentation de la taille du voisinage de la texture. En d'autres termes, grâce à l'information supplémentaire fournie par la couche de structure, la reproduction des motifs de l'exemplaire est réalisable en utilisant un voisinage de texture de plus petite taille.

4.3.4 Comparaison de la méthode proposée avec les approches existantes

L'objectif de cette partie est d'analyser la capacité de notre méthode à synthétiser différents types de textures. L'algorithme proposé (S/T) est donc évalué et comparé avec l'algorithme non-paramétrique de Wei et Levoy (W&L), l'approche paramétrique de Portilla et Simoncelli (P&S), la méthode à base de *patches* de Efros et Leung (E-L), les algorithmes de Tartavel et al. (Tar) et Galerne et al. (Gal). Différentes textures d'entrée sont utilisées, issues de la base de photographies de Brodatz [Bro66] et de clichés de matériaux observés par microscopie électronique en transmission.

Les images utilisées pour la validation expérimentale sont montrées à la Figure 35 tandis qu'une partie des résultats de synthèse sont présentés aux figures 36-38 et 41 (textures régulières structurées) et aux figures 39 et 40 (textures de matériaux plus irrégulières). Les résultats qui restent sont disponibles en Annexe J. Pour chaque résultat, la première colonne montre, de haut en bas, l'échantillon d'entrée, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les deuxième à quatrième colonnes montrent, d'en haut à gauche, jusqu'en bas à droite, les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'algorithme de synthèse à deux étapes

structure/texture proposé (S/T). Dans les Figures 36-40, l'image d'orientation de chaque texture synthétisée est représentée juste au-dessous et les figures qui incluent les images de cohérences sont données en Annexe H. Dans la Figure 41, seules les textures synthétisées sont montrées et les figures qui incluent les images d'orientation et de cohérences sont données en Annexe I.

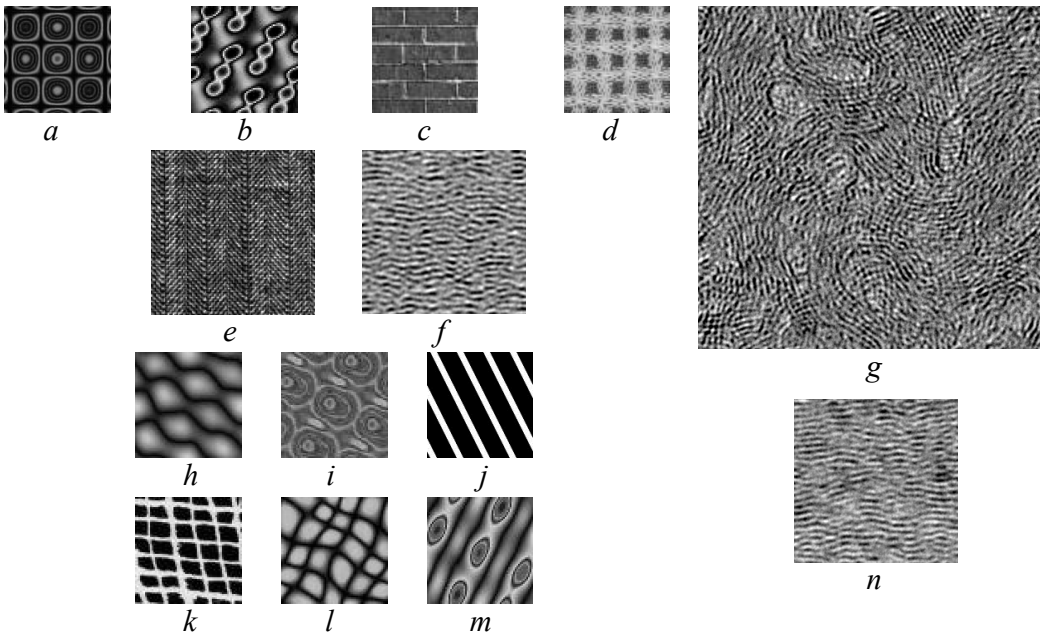


Figure 35 – *Echantillons de texture utilisés pour la validation expérimentale. Les textures a-e, et h-m proviennent* de *la base de données de Brodatz* [Bro66]. Les *textures f, g et n sont des échantillons de matériaux observés par microscopie électronique en transmission.*

L'écart-type du noyau Gaussien du tenseur est fixé à $\sigma = 1.5$ pour les images a, b, c, e, g, j, k et l, à $\sigma = 2.0$ pour les images d, h, i et m et à $\sigma = 1.0$ pour les images f et n.

Concernant les textures montrées dans les Figure 36.*a* et Figure 37.*d*, les résultats obtenus avec l'approche proposée et avec E-L sont meilleurs que ceux obtenus avec W&L, P&S, Tar et Gal. Les résultats de la Figure 36.*b* sont meilleurs avec W&L qu'avec P&S, Tar et Gal. Toutefois, certains artefacts existent toujours dans les images de sortie tandis que S/T et E-L aboutissent à des images sans artefacts et de meilleure qualité.

La Figure 37.*c* illustre un mauvais résultat de synthèse obtenu par les six approches, montrant des difficultés à conserver les dynamiques et la structure de l'échantillon d'entrée, ce qui aboutit à des textures synthétiques présentant des motifs indésirables.

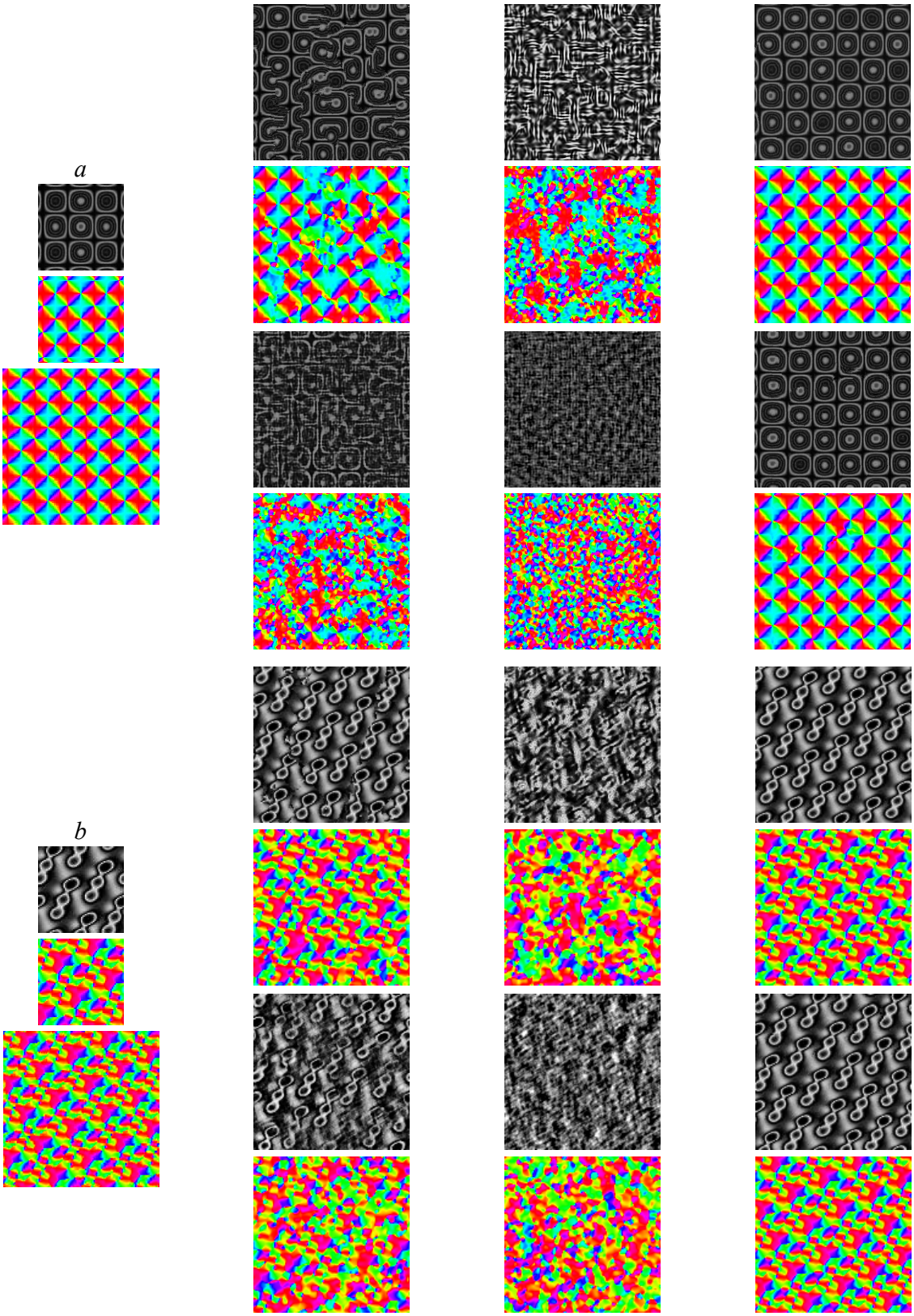


Figure 36 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation de chaque texture qui est représentée juste au-dessous.*

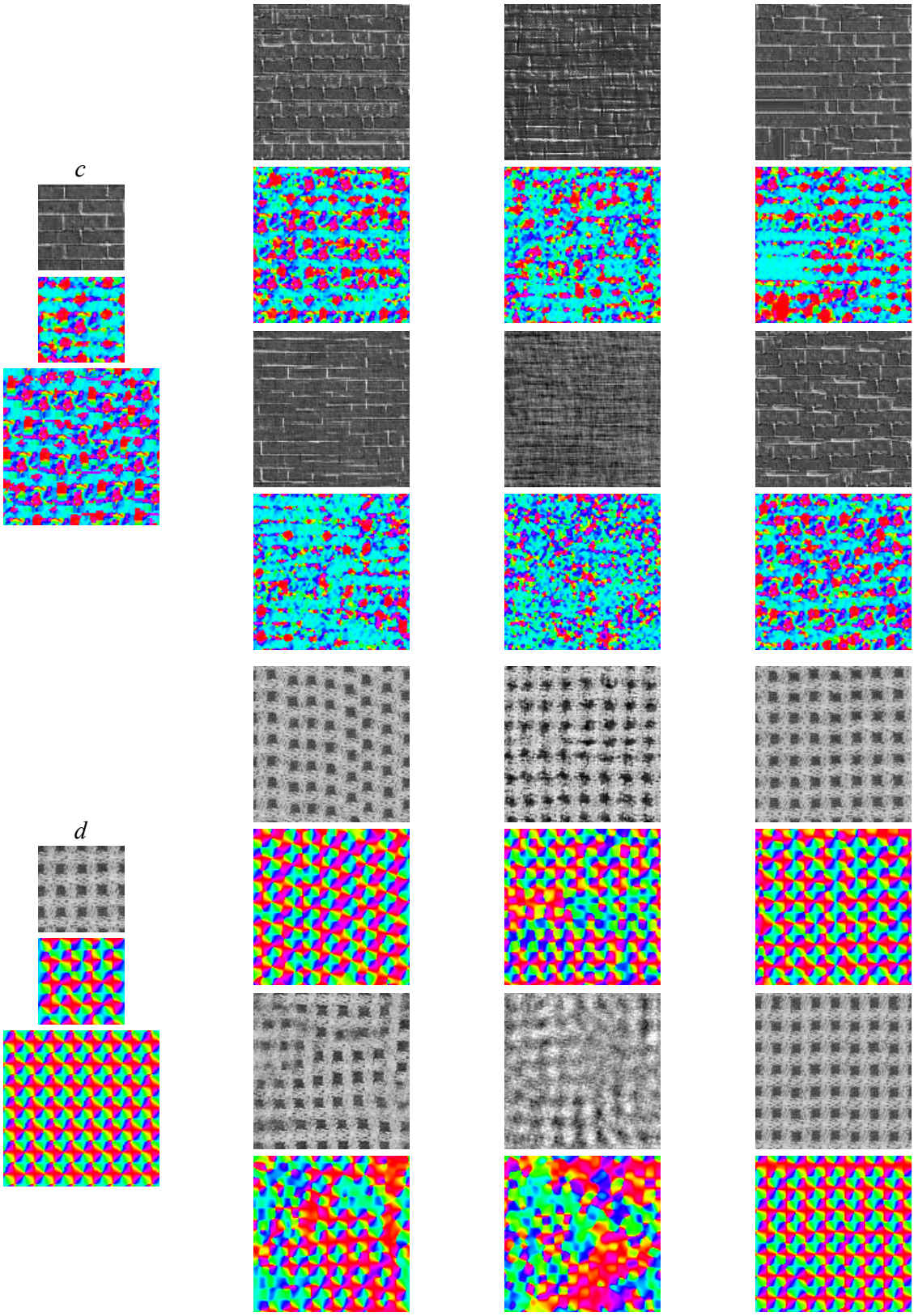


Figure 37 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation de chaque texture qui est représentée juste au-dessous.*

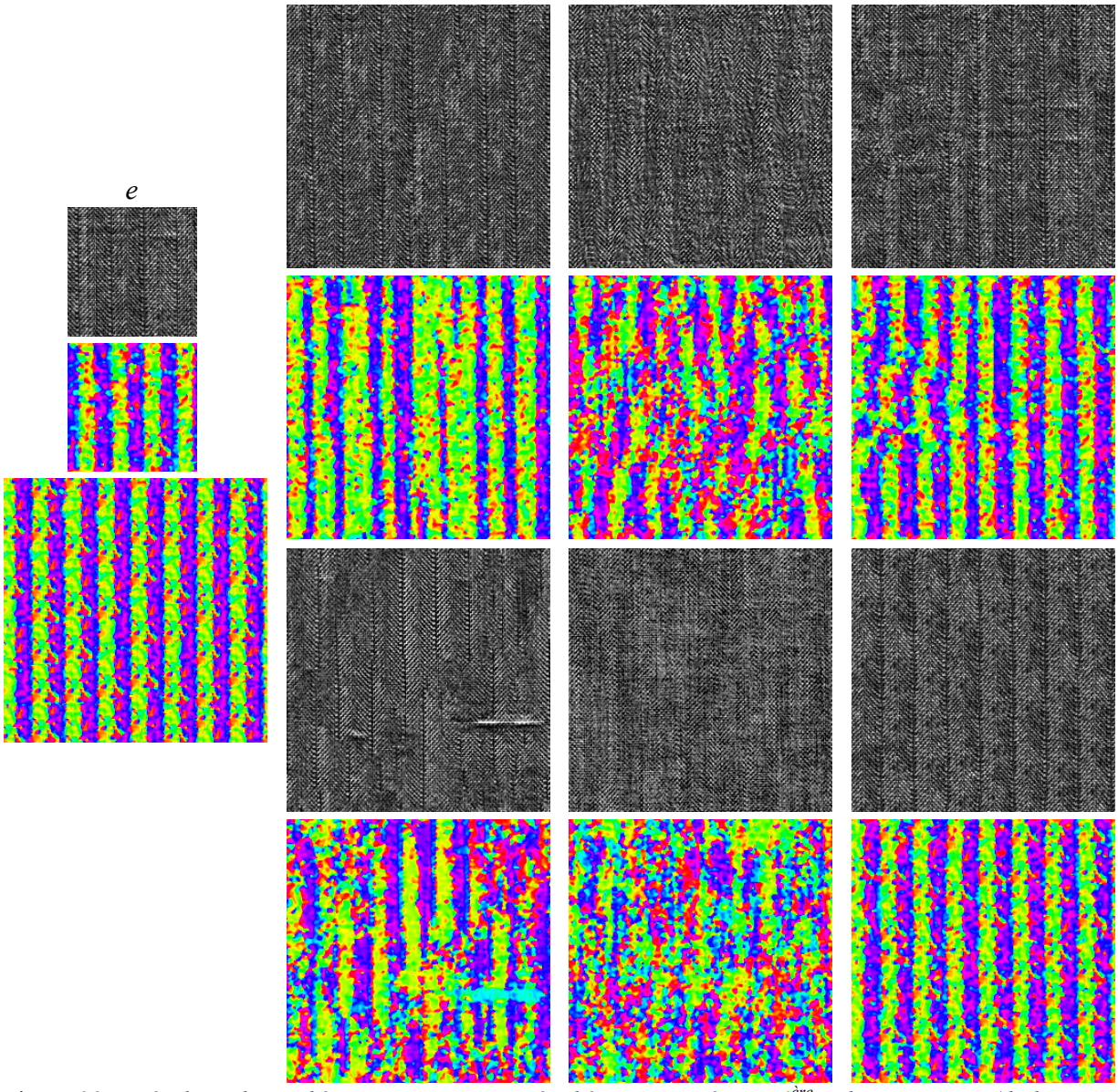


Figure 38 – *Résultats de synthèse sur une texture régulière structurée. La 1ère colonne montre (de haut en bas) l'exemplaire, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation de chaque texture qui est représentée juste au-dessous.*

L'algorithme proposé surpasse les autres algorithmes à la Figure 38.*e* surtout en termes de préservation de la structure. On peut clairement observer que la couche de structure est bien synthétisée et que l'image d'orientation synthétique est très semblable à l'image d'orientation de la texture. Grâce à l'information supplémentaire fournie par la couche de structure, l'approche proposée est capable de reproduire avec succès les variations d'orientations de l'exemplaire même lorsque W&L, P&S, E-L, Tar et Gal ne parviennent pas à préserver la structure.

Les échantillons des Figures 39 et 40 sont des images de dépôts pyrolytiques dans des composites carbone-carbone [Sav93], observées par microscopie électronique en transmission à haute résolution (*High Resolution Transmission Electron Microscopy - HRTEM*). Des échantillons de plus grande taille sont disponibles en Annexe K. Ces textures anisotropes sont difficiles à synthétiser. La plupart des approches existantes ont tendance à produire des textures plus régulières que l'exemplaire, en échouant à reproduire les différentes variations d'orientation [Urs12].

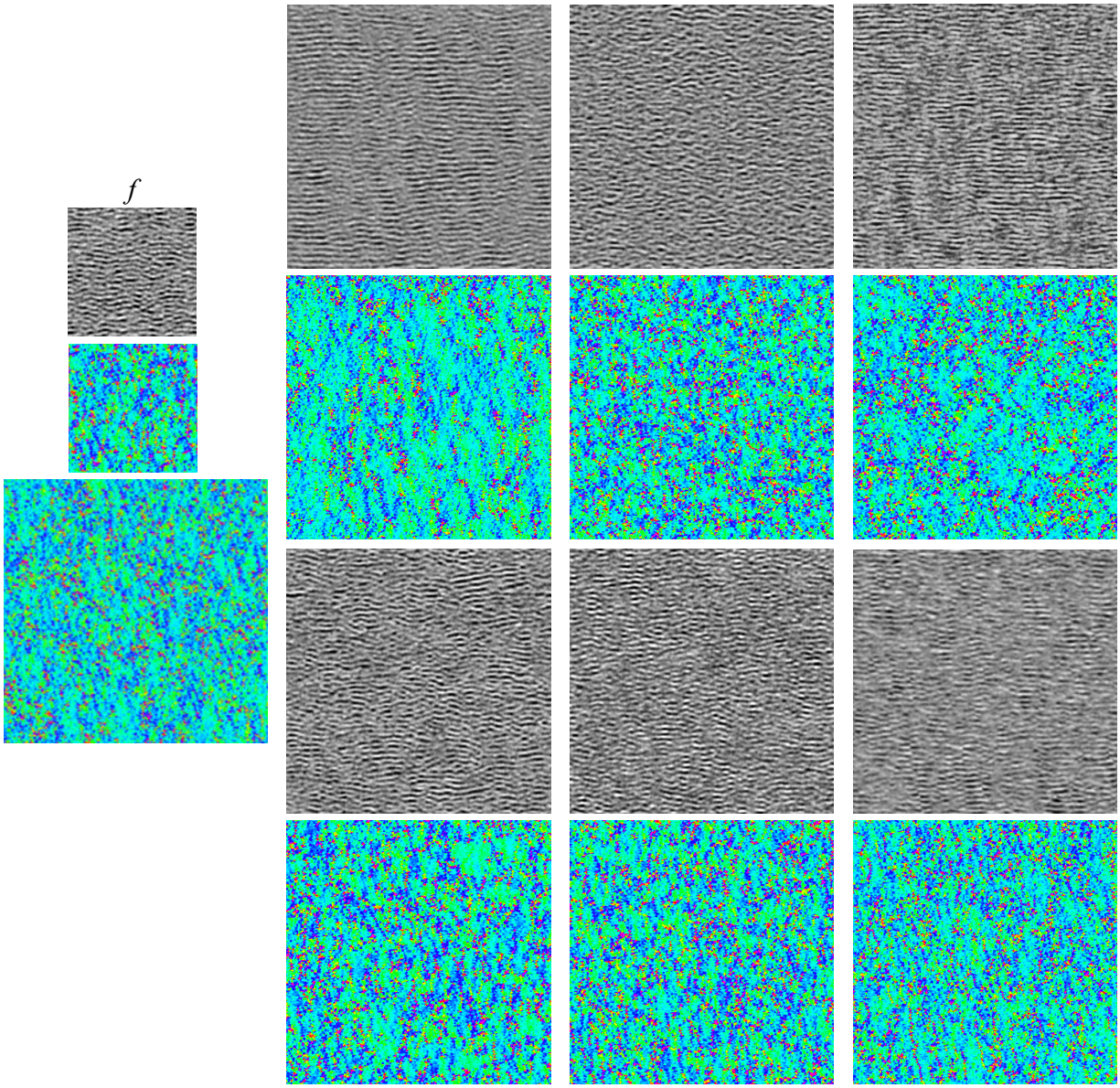


Figure 39 – *Résultats de synthèse sur une texture irrégulière obtenue par microscopie électronique en transmission à haute résolution. La 1ère colonne montre (de haut en bas) l'exemplaire, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation de chaque texture qui est représentée juste au-dessous.*

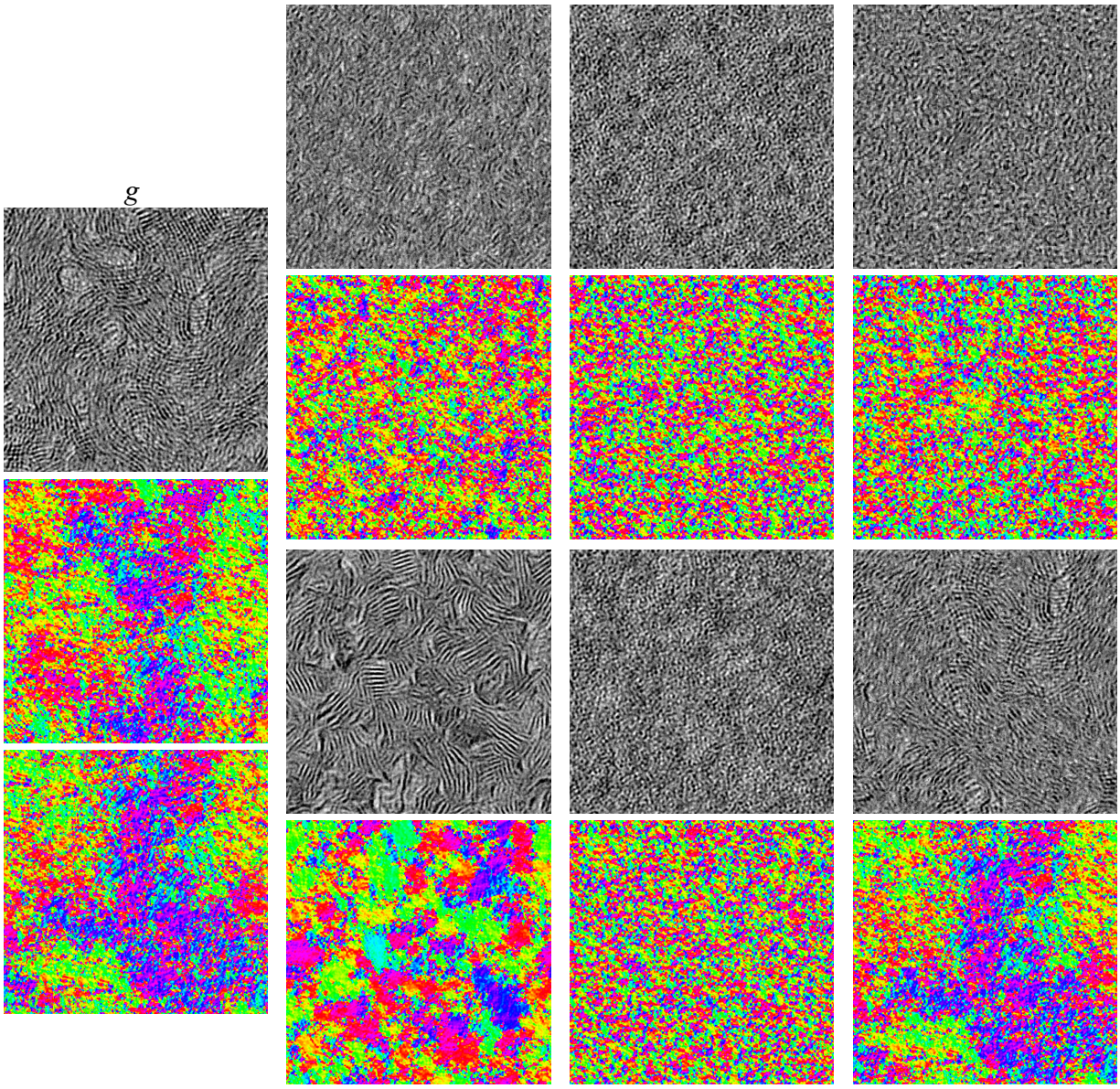


Figure 40 – *Résultats de synthèse sur une texture irrégulière obtenue par microscopie électronique en transmission à haute résolution. La 1ère colonne montre (de haut en bas) l'exemplaire, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation de chaque texture qui est représentée juste au-dessous.*

On peut remarquer que l'approche proposée, P&S, Tar et Gal aboutissent à des textures synthétiques de bonne qualité dans la Figure 39.*f* tandis que les images de sortie obtenues avec W&L et E-L sont plus régulières que la texture d'entrée. Dans la Figure 40.*g*, aucune méthode n'arrive à reproduire la variabilité d'orientation à grande échelle observée sur l'exemplaire, à l'exception de l'approche S/T qui aboutit à un résultat plus réaliste en respectant mieux les orientations des structures de l'exemplaire.

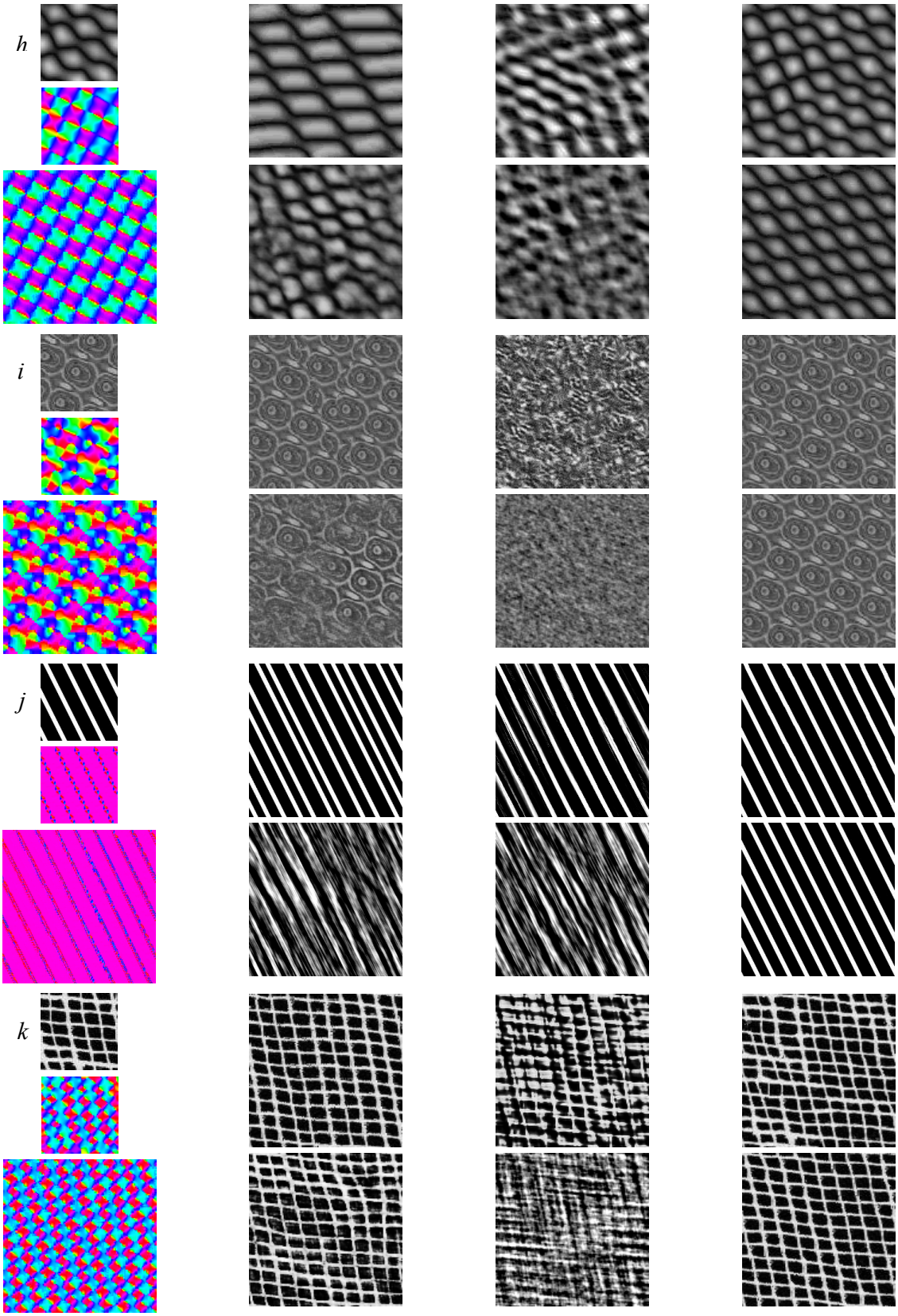


Figure 41 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, l'image d'orientation du champ de tenseurs de structure d'entrée et l'orientation du champ de tenseurs de structure synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T).*

Bien que E-L donne de meilleurs résultats que W&L, P&S, Tar et Gal pour les textures structurées des Figures 41.*h*, *i* et *j*, des artefacts indésirables sont toutefois remarqués dans les textures synthétiques. En revanche, l'algorithme de synthèse structure/texture proposé donne des résultats plus satisfaisants. Dans la Figure 41.*k* les méthodes S/T et W&L se comportent de façon similaire et aboutissent à des textures synthétisées de qualité satisfaisante en termes de préservation de la structure et des dynamiques, mais qui semblent un peu plus régulière que la texture d'entrée.

Bien que l'évaluation qualitative subjective présentée ci-dessus fasse apparaître que les résultats obtenus avec l'algorithme proposé sont satisfaisants (à l'exception de la Figure 37.*c*), une analyse quantitative est essentielle. Une étude objective est ainsi proposée. Il s'agit de comparer les caractéristiques des textures d'entrée et de sortie en utilisant les statistiques introduites au paragraphe 4.3.2 (les histogrammes des intensités, les histogrammes d'orientations, les coefficients d'autocorrélation des intensités et les coefficients d'autocorrélation des orientations locales).
On propose de calculer la divergence de Kullback-Leibler [Kul51] entre les histogrammes des intensités et entre les histogrammes d'orientations. Les coefficients d'autocorrélation des intensités et des orientations locales sont considérés pour des déplacements horizontaux et verticaux limités à 15 pixels et comparés en utilisant la distance Euclidienne. Les graphes de la Figure 42 résument l'évaluation objective sur l'ensemble des 14 résultats de synthèse montrés aux Figures 36 à 41 et en Annexe J. Ils confirment généralement les remarques subjectives faites précédemment, notre méthode conduisant habituellement aux meilleurs résultats ou étant très proche des meilleures performances obtenues.

Parmi les quatre graphes proposés, la distance entre les coefficients d'autocorrélation des orientations locales semble être la plus appropriée pour traduire le respect des formes et de l'organisation des motifs de la texture exemplaire. Par exemple, les défauts de structure présents avec P&S et Gal dans les Figures 36.*a* et *b* ou encore avec P&S, Gal et Tar dans la Figure 37.*d* ainsi qu'avec toutes les méthodes sauf l'approche proposée dans la Figure 40.*g* apparaissent clairement sur le graphe des coefficients d'autocorrélation des orientations. Au contraire, les histogrammes des intensités et les histogrammes d'orientations ne fournissent pas suffisamment d'information pour bien analyser la préservation de la structure. C'est le cas pour les résultats des Figures 36.*a*, 37.*d* et 38.*e* où les dynamiques semblent être similaires avec W&L et l'approche proposée tandis que les motifs des textures sont très différents et

déformés avec W&L. Enfin, le succès de la méthode proposée dans la reproduction des motifs des échantillons de matériau de pyrocarbone des Figures 39.*f* et 40.*g* est vérifié.

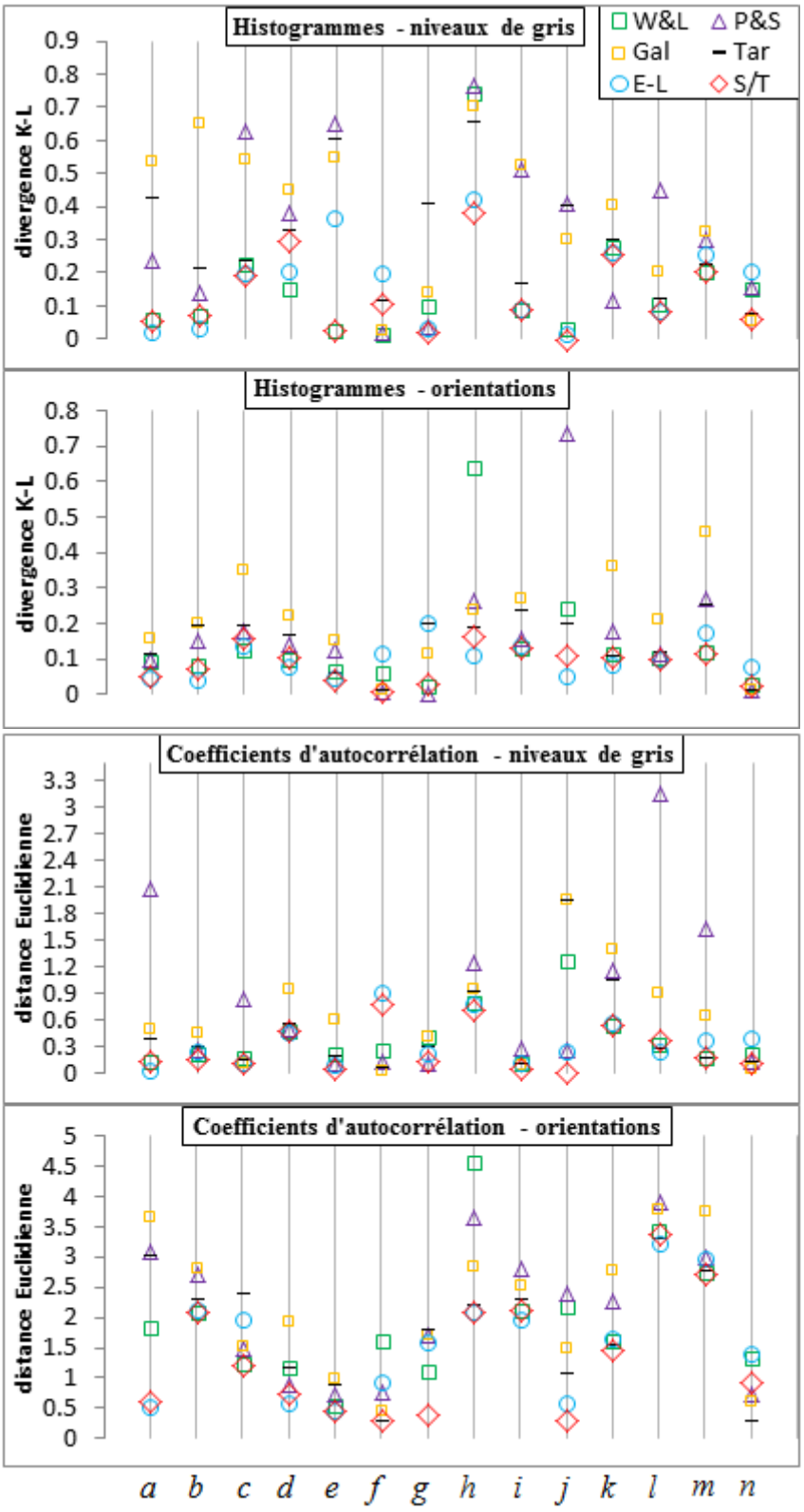


Figure 42 – *Indicateurs objectifs pour les résultats de synthèse montrés aux Figures 36-41 et en Annexe J.*

4.4 Conclusion

Nous avons présenté une approche de synthèse de texture contrainte par un champ de tenseurs de structure préalablement synthétisé à partir du champ de tenseurs de la texture exemplaire.

Le gain en termes de reproduction de la structure et des variations d'orientation est tout d'abord établi en présentant l'influence du poids de pondération structure/texture sur la qualité des textures synthétisées.

Une étude sur l'effet de la taille des voisinages utilisés dans les algorithmes de synthèse a montré qu'en raison de l'information supplémentaire fournie par la structure, la reproduction des motifs de l'exemplaire est réalisable en utilisant un voisinage de texture de plus petite taille que celle utilisée pour la synthèse du champ de tenseurs de structure. Celui-ci doit être à l'échelle du plus grand motif observé afin de préserver la structure de l'échantillon d'entrée.

Une procédure expérimentale comparant l'approche de synthèse proposée à cinq algorithmes concurrents (Wei et Levoy, Portilla et Simoncelli, Efros et Leung, Tartavel et al., et Galerne et al.) a été présentée. Différentes textures d'entrée issues de la base de données de Brodatz [Bro66] et d'images de matériaux ont été utilisés. Les textures synthétisées sont comparées aux textures d'entrée en utilisant les statistiques de 1er et 2ème ordre des intensités et des orientations locales (notamment les histogrammes des intensités, les histogrammes d'orientations, les coefficients d'autocorrélation des intensités et les coefficients d'autocorrélation des orientations locales). Les résultats de synthèse montrent que les textures obtenues par l'approche proposée sont généralement meilleurs ou très proches des meilleurs résultats. En outre, l'approche proposée surpasse les cinq autres algorithmes de synthèse et synthétise avec succès les textures dans de nombreuses situations où ces algorithmes ne parviennent pas à reproduire les motifs de l'image d'entrée.

Chapitre 5

Les extensions de l'algorithme de synthèse proposé

Sommaire

5.1 Introduction 86
5.2 Synthèse de texture couleur 86
5.2.1 Extension de l'approche structure/texture 87
5.2.2 Résultats 88
5.3 Synthèse de texture 3D 93
5.3.1 Synthèse de texture 3D à partir d'un exemplaire 3D 93
5.3.2 Résultats 95
5.3.3 Inférence 2D/3D du tenseur de structure 98
5.4 Synthèse de texture contrainte par champ de structure arbitraire 102
5.4.1 Construction du dictionnaire de recherche 103
5.4.2 Construction de la contrainte et processus de synthèse 104
5.4.3 Résultats 108
5.4.3.1 Pertinence du champ de tenseurs contrainte 108
5.4.3.2 Effet du pas angulaire du dictionnaire de recherche 110
5.4.3.3 Exemples de résultats 111
5.5 Conclusion 115

5.1 Introduction

Ce chapitre présente plusieurs extensions de l'algorithme de synthèse à deux étapes structure/texture. Dans un premier temps, une extension de l'algorithme pour la synthèse de texture couleur est développée en proposant deux méthodes différentes pour le calcul du tenseur de structure d'une image couleur. L'extension à la synthèse de textures 3D est traitée dans un second temps puis la problématique de la synthèse de textures volumiques par inférence 2D/3D du champ de tenseurs de structure est abordée. Enfin, une dernière extension de l'approche de synthèse consiste à contraindre la synthèse de la couche de texture par un champ de structure arbitraire dans l'objectif de créer des textures de formes spécifiées et non forcément stationnaires.

5.2 Synthèse de texture couleur

Beaucoup d'applications graphiques nécessitent la synthèse de textures couleur pour représenter les textures du monde réel. Les textures synthétisées peuvent imiter les détails de la surface d'objets réels, comme les variations de couleur, ou les déformations géométriques. L'utilisation de textures scannées ou dessinées à la main est inadéquate pour recouvrir des objets dans les applications graphiques (*texture mapping*) [Win94]. En effet, leur taille originelle n'ayant aucune raison d'être adaptée à celle des objets, leur utilisation nécessite un tuilage dont les frontières peuvent être visibles nuisant ainsi au réalisme de la scène.

La plupart des algorithmes de synthèse de la littérature (cf. paragraphe 2.3.4) sont extensibles aux images couleur en considérant les trois canaux (rouge, vert et bleu) de la texture couleur dans le modèle *RGB* [Wei00] [Kwa03] [Wei03]. Toutefois, comme dans le cas de textures en niveaux de gris, pour la synthèse de textures anisotropes structurées, les approches existantes ont souvent tendance à produire des textures plus régulières que l'exemplaire [Wei00] [Kop07].

En nous fondant sur l'algorithme de synthèse structure/texture proposé au chapitre précédent, nous présentons ici une extension à la synthèse de textures couleur. Deux approches sont utilisées pour le calcul du tenseur de structure sur des images couleur. Les résultats de synthèse montreront que la méthode proposée aboutit à des résultats de qualité meilleure ou comparable à ceux obtenus par l'algorithme classique de Wei et Levoy, avec une réduction significative de la charge de calcul.

5.2.1 Extension de l'approche structure/texture

La première étape de l'algorithme de synthèse proposé, consiste à synthétiser la couche de structure, représentée par le champ de tenseur de structure. Par conséquent, un point important dans l'extension couleur concerne le calcul du tenseur de structure sur une texture couleur. Di Zenzo [Zen86] propose une formulation du tenseur pour le gradient d'une image multi composantes basée sur l'extraction d'un vecteur unique (direction et amplitude de variation maximale). Weijer et Gevers [Wei04] proposent de faire la somme des trois composantes du tenseur de structure calculées pour chaque canal dans le modèle *RGB*. La même méthode est utilisée dans [Gar08] avec l'ajout d'un lissage Gaussien.

Dans ce mémoire, deux approches différentes sont utilisées pour le calcul du champ de tenseurs d'une image couleur. La première approche consiste à extraire la composante de luminance (Y) de la texture couleur d'entrée, comme défini dans la recommandation ITU-R BT.601 [ITU11]:

$$Y = 0.299R + 0.587G + 0.114B \ , \tag{21}$$

où R, G et B représentent les composantes rouge, verte et bleue de l'image d'entrée, respectivement.
Le tenseur de structure est ensuite calculé à partir de la composante de luminance (Y) de la même façon qu'on le calcule à partir des images en niveaux de gris.

La deuxième méthode de calcul d'un champ de tenseurs de structure couleur $\bar{S}^C$ repose sur l'addition des tenseurs pour les différents canaux comme suit [Wei04]:

$$\bar{S}^C = \begin{bmatrix} \bar{S}^C_{xx} & \bar{S}^C_{xy} \\ \bar{S}^C_{xy} & \bar{S}^C_{yy} \end{bmatrix}, \tag{22}$$

où les composantes du champ de tenseurs couleur $\bar{S}^C_{xx}$, $\bar{S}^C_{xy}$ et $\bar{S}^C_{yy}$ s'expriment :

$$\bar{S}^C_{ab} = \sum_{c \in \{R,G,B\}} S^c_{ab} \quad ; \quad ab \in \{xx, xy, yy\} \tag{23}$$

avec S^R_{ab}, S^G_{ab} et S^B_{ab} qui désignent les composantes du champ de tenseurs de structure obtenues sur les trois canaux R, G et B de la texture couleur, respectivement.

L'algorithme de synthèse du champ de tenseurs de structure est identique à celui présenté au chapitre 3, dans le cas d'un champ de tenseurs issu d'une texture en niveaux de

gris. L'étape consistant à utiliser le champ de tenseurs de structure synthétisé pour contraindre la synthèse de la texture est elle aussi similaire au cas des niveaux de gris à ceci près que, dans le cas de données vectorielles (i.e. couleur), la métrique utilisée pour mesurer la ressemblance des voisinages aux sites *z* et *z'* (cf. équation (16)) s'exprime par :

$$\gamma^C(z,z') = p \cdot CSSD\big(VI_{in}(z), VI_{out}(z')\big) + (1-p) \cdot SSTD\left(VS_{in}^C(z), VS_{out}^C(z')\right), \qquad (24)$$

où *SSTD* est la somme des valeurs de dissimilarité des tenseurs de structure (*Sum of Structure Tensor Dissimilarity*) définie dans l'équation (13), *p* est le facteur de pondération et *CSSD* est la somme des erreurs quadratiques calculée sur des données couleur (*Color Sum of Squares Distance*) comme suit :

$$\begin{aligned} CSSD\big(VI_{in}(z), VI_{out}(z')\big) = \sum_{k=1}^{K} & (I_{in}^R(t_k) - I_{out}^R(t'_k))^2 \\ & + (I_{in}^G(t_k) - I_{out}^G(t'_k))^2 + (I_{in}^B(t_k) - I_{out}^B(t'_k))^2 , \end{aligned} \qquad (25)$$

où $I_{in}^c(t_k)$ (resp. $I_{out}^c(t'_k)$); $c \in \{R, G, B\}$ désigne la $k^{\text{ème}}$ intensité du voisinage ${VI_{in}}^c(t_k)$ (resp. ${VI_{out}}^c(t'_k)$), dans les canaux *R*, *G* et *B* respectivement et *K* est le nombre de pixels dans chaque voisinage.

5.2.2 Résultats

Cette partie vise à évaluer l'algorithme proposé en utilisant différentes textures couleur d'entrée issues de la base de données couleur de Brodatz [Bro]. La Figure 43 présente des résultats de synthèse sur trois textures différentes. Pour chaque résultat, la première colonne montre, à la première ligne, la texture d'entrée, à la deuxième ligne, l'image d'orientation de son champ de tenseurs de structure calculé en additionnant les canaux du tenseur (à gauche) et en utilisant la composante de luminance (à droite) et, à la troisième ligne, la texture synthétisée en utilisant l'algorithme de W&L. Les trois autres colonnes montrent, en haut, l'image d'orientation du champ de tenseurs synthétisé et, en bas, la texture résultante. Les deuxième à quatrième colonnes sont relatives respectivement aux approches utilisant l'additivité des canaux du tenseur pour le calcul du champ de tenseur de structure, le calcul du champ de tenseurs sur la composante de luminance (*Y*) et l'algorithme de synthèse accéléré par interpolation bilinéaire avec *Ls* = 1 (cf. paragraphe 3.3.5) et avec le calcul du tenseur de structure sur la composante *Y*.

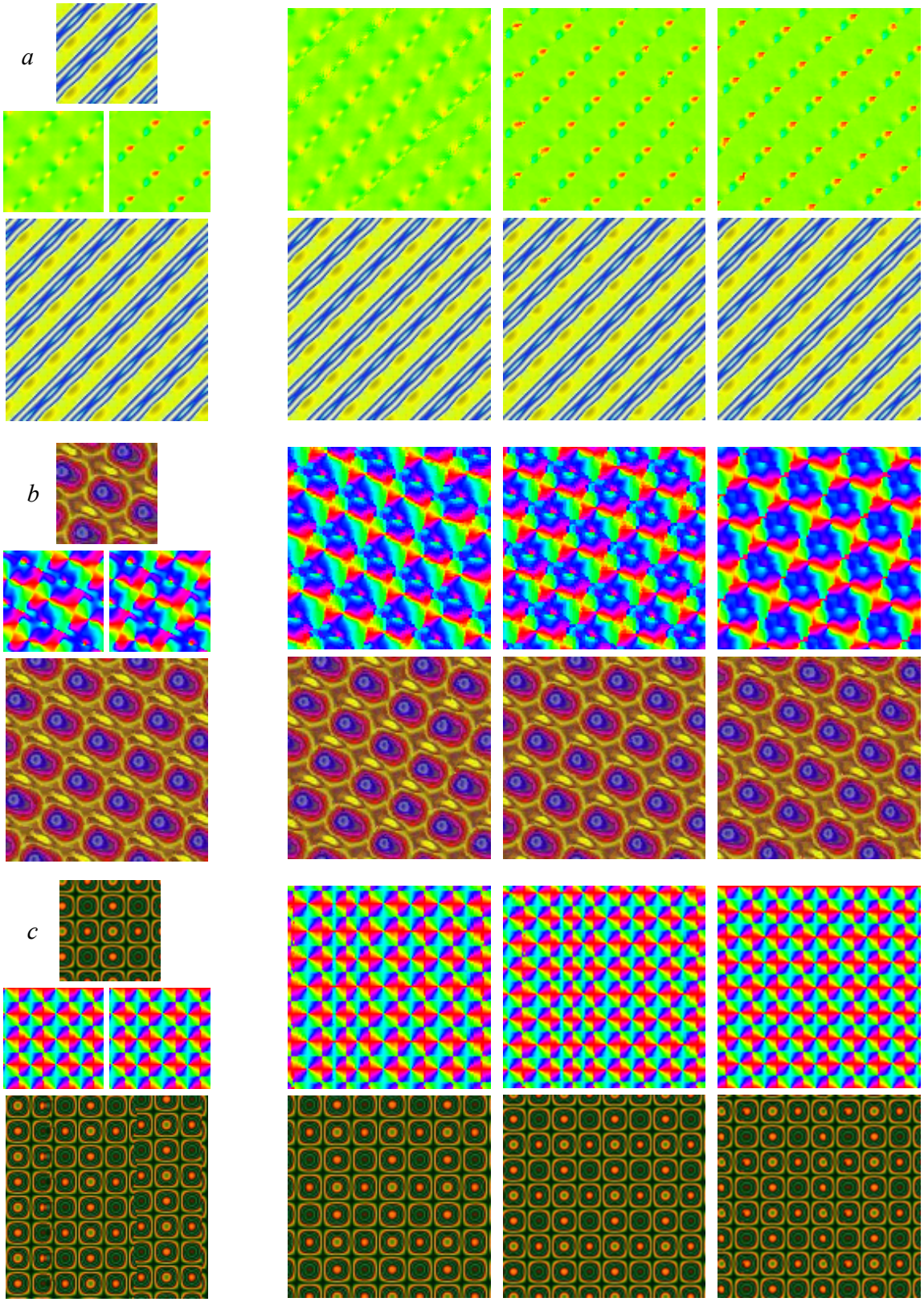


Figure 43 – *Résultats de synthèse. Pour chaque résultat, la 1ère colonne, en haut : l'exemplaire, l'orientation du tenseurs d'entrée calculé en utilisant l'additivité des canaux de tenseurs (à gauche) et sur la composante de luminance Y (à droite), et en bas : la texture synthétisée par W&L. 2ème, 3ème et 4ème colonne, en haut : l'orientation du tenseur synthétisé et en bas : la texture obtenue, en utilisant l'additivité des canaux du tenseur, en calculant le tenseur sur la composante Y et en utilisant l'accélération par interpolation avec le calcul du tenseur sur Y, respectivement.*

Notons qu'aucune accélération logicielle – e.g. l'accélération par quantification vectorielle (*Tree-Structured Vector Quantization* - *TSVQ*) – n'a pas été utilisée pour éviter tout effet lié à la sous-optimalité d'une telle accélération [Wei00].

Trois itérations sont utilisées pour obtenir le champ de tenseurs synthétisé dans les résultats *a* et *b* et quatre itérations dans le résultat *d*. Pour tous les résultats, les textures obtenues avec l'approche proposée sont montrées après deux itérations. Pour la synthèse du tenseur, des pyramides Gaussiennes à deux échelles sont utilisées dans les résultats *a*, *b* et *c* avec un voisinage de taille 11×11, 13×13 et 21×21, respectivement. Une synthèse de texture mono-échelle et un voisinage causal avec à un parcours lexicographique sont utilisés pour tous les résultats. La taille du voisinage de synthèse de la texture est 9×9 dans les résultats *a* et *b*, et 21×21 dans le résultat *c*.

On peut observer que les textures obtenues avec la synthèse contrainte par la structure, en utilisant les deux approches pour le calcul du tenseur de structure couleur, sont similaires à ceux qui sont obtenus avec W&L, dans les résultats de la Figure 43.*a* et *b*. Au contraire, les deux approches surpassent W&L dans le résultat de la Figure 43.*c* surtout en termes de conservation de la structure. Dans le résultat de la Figure 43.*c* la texture obtenue avec W&L présente quelques artefacts indésirables et l'alternance et la périodicité des motifs de l'exemplaire ne sont pas respectées. Au contraire, notre approche aboutit à des textures de meilleure qualité, sans artefact, et semblent très similaires à l'échantillon d'entrée donnant l'impression qu'ils ont été produits par le même processus. En d'autres termes, grâce à l'information supplémentaire fournie par la couche de structure, l'approche proposée est capable de reproduire avec succès les variations d'orientations dans l'exemplaire même lorsque W&L ne parvient pas à maintenir la structure.

Dans les résultats des Figures 43.*b* et *c*, les images d'orientation des champs de tenseurs d'entrée sont pratiquement similaires pour les deux méthodes de calcul du tenseur de structure couleur. Ce n'est pas le cas pour la texture de la Figure 43.*a* où chaque approche donne une image d'orientation visuellement différente. Cependant, pour l'ensemble des trois exemplaires, il est difficile de distinguer les textures synthétiques obtenues par les deux approches. Il apparait également que, dans tous les résultats, les images obtenues en utilisant l'interpolation bilinéaire sont similaires à celles générées par l'approche non-accélérée. Le tenseur de structure interpolé reste de bonne qualité et les images d'orientation synthétiques

ressemblent aux images d'orientation de la texture en termes de préservation de la structure et de la dynamique.

Le Tableau 4 présente les temps de simulation relatifs aux résultats de la Figure 43. La deuxième ligne montre le temps de calcul de l'algorithme de W&L. Les troisième et quatrième lignes montrent le temps de calcul de notre algorithme en utilisant la méthode de synthèse classique de tous les niveaux et en utilisant l'algorithme accéléré par interpolation bilinéaire, respectivement. Tous les temps de calcul présentés sont en secondes et mesurés en utilisant un processeur '*Intel Core i7-2670QM, 2.20 GHz*', sans parallélisation. Même si la synthèse de la couche de texture est réalisable en utilisant un voisinage de taille inférieure à celle de l'étape de synthèse de structure, la surperformance de l'algorithme proposé est obtenue au détriment du temps de calcul. Ceci est montré d'après les temps de calcul de l'algorithme non-accéléré à la troisième ligne du Tableau 4.

Tableau 4 – *Temps de simulation (en secondes) pour les textures de la Figure 43.*

Texture:	*a*	*b*	*c*
W&L	28	35	756
Méthode proposée	41	42	1114
Algorithme accéléré	20	21	424

En revanche, l'algorithme accéléré par l'utilisation de l'interpolation bilinéaire dans l'étape de synthèse de la couche de structure, surpasse W&L en termes de temps de calcul, dans tous les résultats du Tableau 4. Par exemple, W&L et l'algorithme classique proposé prennent respectivement 756 et 1114 secondes pour générer la texture de sortie dans le résultat de la Figure 43.*c*, tandis que l'algorithme accéléré nécessite 424 secondes. Il est important de noter que même lorsque l'algorithme de synthèse de W&L est capable de synthétiser avec succès la texture d'entrée, la version accélérée de la synthèse à deux étapes proposée est avantageuse en réduction du temps de synthèse sans aucune perte de qualité de la texture de sortie, comme c'est le cas dans les résultats de la Figure 43.*a* et *b*.

La Figure 44 montre deux résultats de synthèse sur des textures irrégulières présentant de fortes variations directionnelles. Chaque résultat montre de gauche à droite, la texture d'entrée, la texture synthétisée en utilisant l'algorithme de W&L, le résultat obtenu avec l'algorithme non-accéléré et avec l'algorithme accéléré, en calculant les tenseurs de structure sur la composante de luminance. On peut observer que dans le résultat *a*, les textures obtenues

avec l'approche de synthèse contrainte par le champ de tenseurs sans et avec accélération, sont similaires à celle qui est synthétisée avec W&L. D'autre part, W&L n'arrive pas à reproduire les variations d'orientations de l'exemplaire, dans le résultat *b*, tandis que les résultats obtenus par l'algorithme de synthèse proposé sont plus réalistes et respectent mieux les orientations des structures de l'échantillon d'entrée.

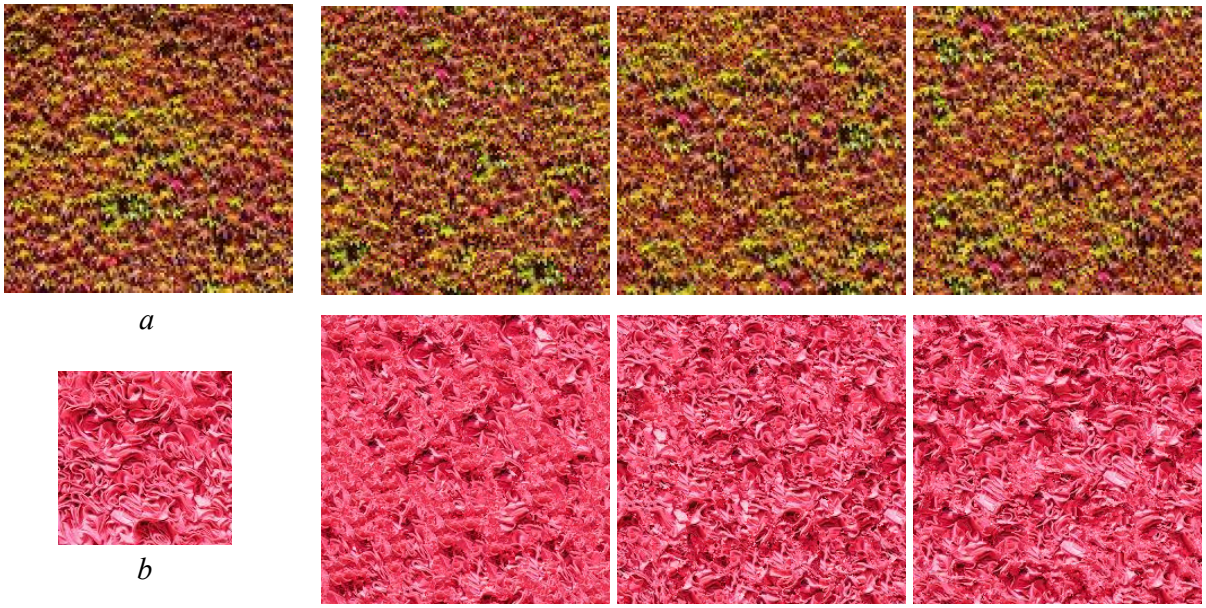


Figure 44 – *Résultats de synthèse sur des textures irrégulières. Pour chaque résultat (de gauche à droite) : l'exemplaire, la texture obtenue en utilisant W&L, les textures synthétisées par la méthode proposée sans accélération et avec accélération par interpolation, en calculant le tenseur de structure sur la composante de luminance.*

Des résultats de synthèse supplémentaires sur des textures régulières structurées et irrégulières sont données en Annexe L.

5.3 Synthèse de texture 3D

Dans cette partie, nous présenterons l'extension de l'algorithme de synthèse proposé à la synthèse de textures volumiques (3D), dans un premier temps. Nous discuterons dans un second temps de la problématique de la synthèse d'un champ de tenseurs de structure 3D à partir d'un ou de plusieurs champs de tenseurs 2D par inférence 2D/3D, selon les méthodologies décrites dans [Urs13] et [Wei03].

5.3.1 Synthèse de texture 3D à partir d'un exemplaire 3D

L'algorithme de synthèse en deux étapes structure/texture s'étend de façon naturelle à la synthèse de textures 3D à partir d'exemplaires 3D. Cette extension nécessite le calcul du champ de tenseurs de structure 3D de l'échantillon volumique. Le champ de tenseurs donne donc une information sur l'orientation et l'anisotropie locale des structures 3D de cet échantillon. D'une manière similaire au calcul du champ de tenseurs de structure d'une image 2D (cf. paragraphe 3.2), le champ de tenseurs de structure tridimensionnel S d'un volume I est une matrice 3×3 symétrique semi-définie positive, construite à partir des champs de gradients $\nabla I = [I_x\ I_y\ I_z]$:

$$S = W * \begin{pmatrix} I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & I_z^2 \end{pmatrix} = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{xy} & S_{yy} & S_{yz} \\ S_{xz} & S_{yz} & S_{zz} \end{pmatrix}, \tag{26}$$

où W est un noyau 3D de lissage Gaussien.

Afin de synthétiser un champ de tenseurs de structure volumique similaire à celui qui est calculé sur le volume d'entrée, le principe de synthèse est le même qu'en 2D : un bruit de tenseurs 3D est généré dans une première phase d'initialisation, en choisissant aléatoirement des tenseurs du volume d'entrée, tandis que le processus de synthèse proprement dit consiste ensuite à modifier la sortie 3D bruitée afin de la faire ressembler à l'entrée. La synthèse se fait tenseur par tenseur et, pour chaque tenseur de sortie, le tenseur d'entrée dont le voisinage correspond le mieux à celui de sortie est cherché et copié à la position cible dans le champ de sortie. Une fois calculé, le champ de tenseurs 3D synthétisé est utilisé pour contraindre la synthèse de la texture 3D par recherche de la meilleure correspondance, en prenant en considération l'information de structure portée par la contrainte.

Les métriques de comparaison de tenseurs de structure, les systèmes de voisinages (Figure 45) et l'approche de synthèse multi-échelle par décomposition en pyramides Gaussiennes (Figure 46) utilisés sont les mêmes que ceux de la synthèse 2D, en prenant en considération la profondeur (i.e. la dimension z) des données tridimensionnelles traitées. Il faut noter que pour la métrique *Shape-Orientation* (*SO*), il est nécessaire de calculer les orientations du tenseur de structure tridimensionnel, à partir de ses vecteurs propres, et de former des mesures de cohérence à partir de ses valeurs propres.

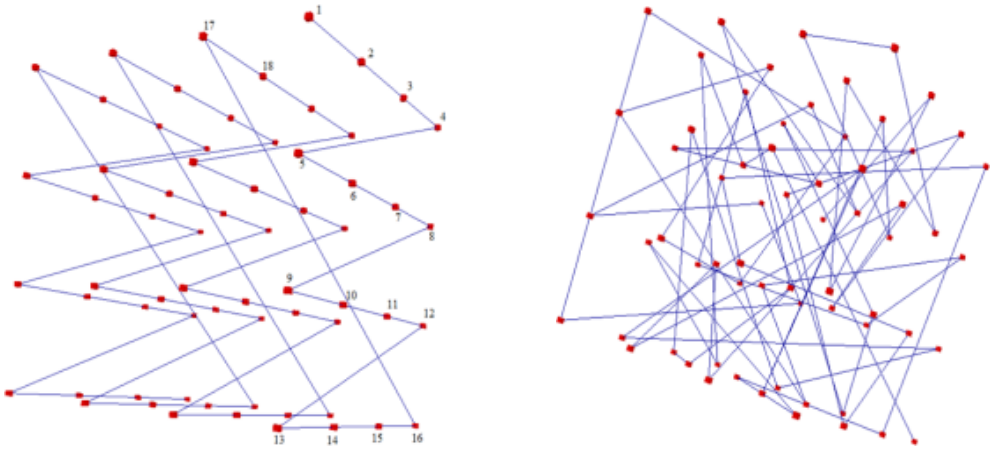


Figure 45 – *Illustration du parcours de synthèse lexicographique (à gauche) et aléatoire (à droite) dans le cas de synthèse de champs de tenseurs de structure et de texture tridimensionnels* [Urs12] [Urs13].

En outre, le calcul du gradient se limite à l'intérieur de l'exemplaire 3D laissant une zone inutilisée dans la procédure de recherche de la meilleure ressemblance comme expliqué au paragraphe 3.3.6, et les champs de tenseurs et les textures de sortie sont traités périodiquement comme dans le paragraphe 3.3.7, en prenant en compte la profondeur des données 3D.

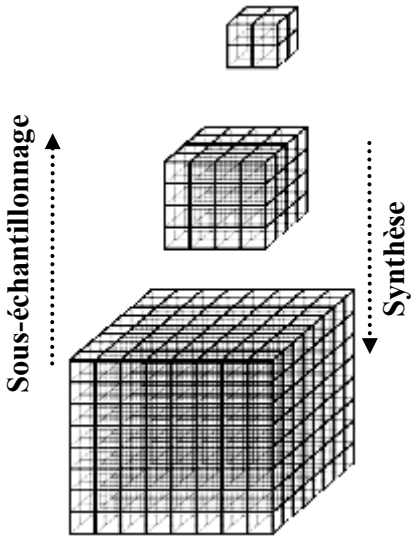


Figure 46 – *Illustration de la manière dont le principe de synthèse multi-résolution paraît en 3D.*

5.3.2 Résultats

Cette partie montre quelques résultats de synthèse tridimensionnelle de texture contrainte par champ de tenseurs de structure. Dans les Figures 47 et 48, la première colonne montre l'échantillon volumique d'entrée. La deuxième colonne montre, de haut en bas, les orientations des champs de tenseurs du volume d'entrée $S^f = \begin{bmatrix} S_{xx} & S_{xy} \\ S_{xy} & S_{yy} \end{bmatrix}$ correspondant à la projection de S sur la vue de face, $S^d = \begin{bmatrix} S_{yy} & S_{yz} \\ S_{yz} & S_{zz} \end{bmatrix}$ correspondant à la projection de S sur la vue de droite et $S^h = \begin{bmatrix} S_{xx} & S_{xz} \\ S_{xz} & S_{zz} \end{bmatrix}$ correspondant à la projection de S sur la vue de haut (cf. Figure 49). La troisième colonne montre (de haut en bas) les mêmes champs d'orientation pour les tenseurs de structure synthétisés. La quatrième colonne montre la texture obtenue par l'algorithme de synthèse structure/texture proposé.

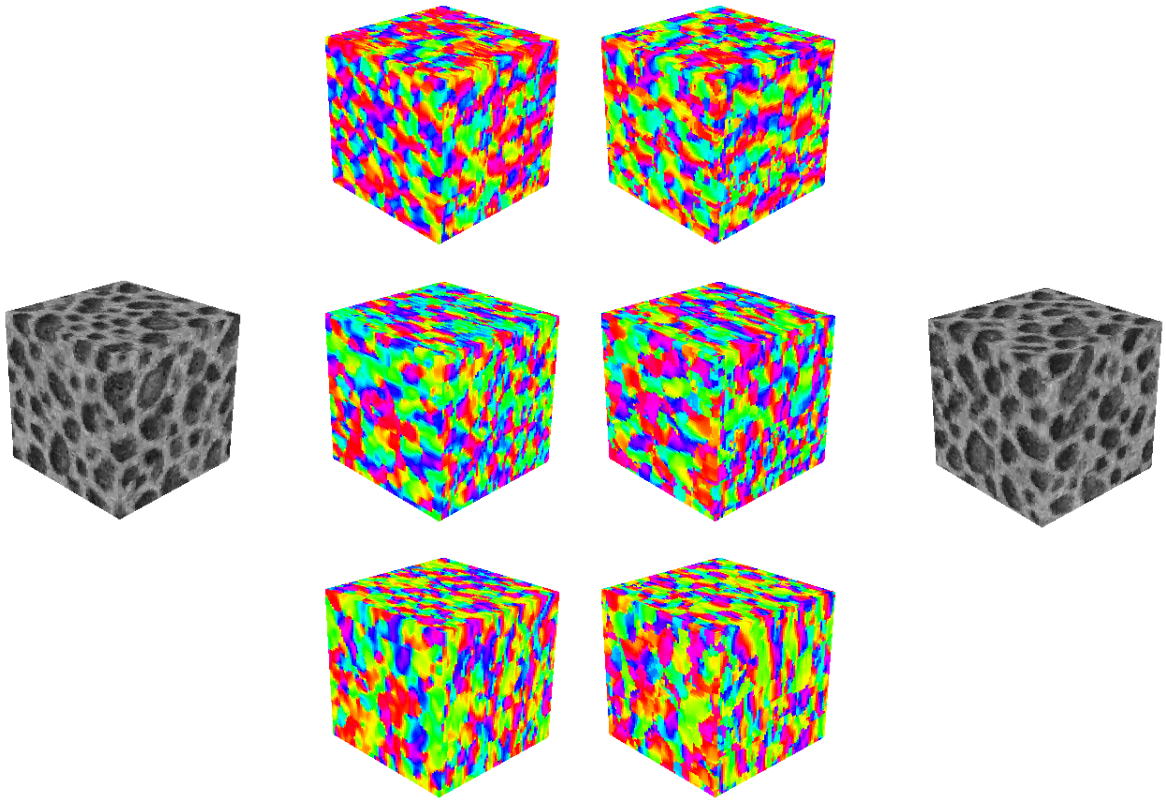


Figure 47 – *Résultat de synthèse sur une texture tridimensionnelle. 1ère colonne : l'échantillon d'entrée, 2ème colonne (de haut en bas) : l'orientation du champ de tenseurs de structure d'entrée montrée par projection sur la vue de face, sur la vue de droite et sur la vue de haut. 3ème colonne : orientation du champ de tenseurs de structure synthétisé, projeté sur les vues de face, de droite et de haut. 4ème colonne : la texture volumique obtenue en utilisant l'approche de synthèse contrainte par champ de tenseurs de structure. L'approche mono-échelle est utilisée avec des voisinages de taille 15×15×15 pour la synthèse de la couche de structure et 11×11×11 pour la synthèse de la texture.*

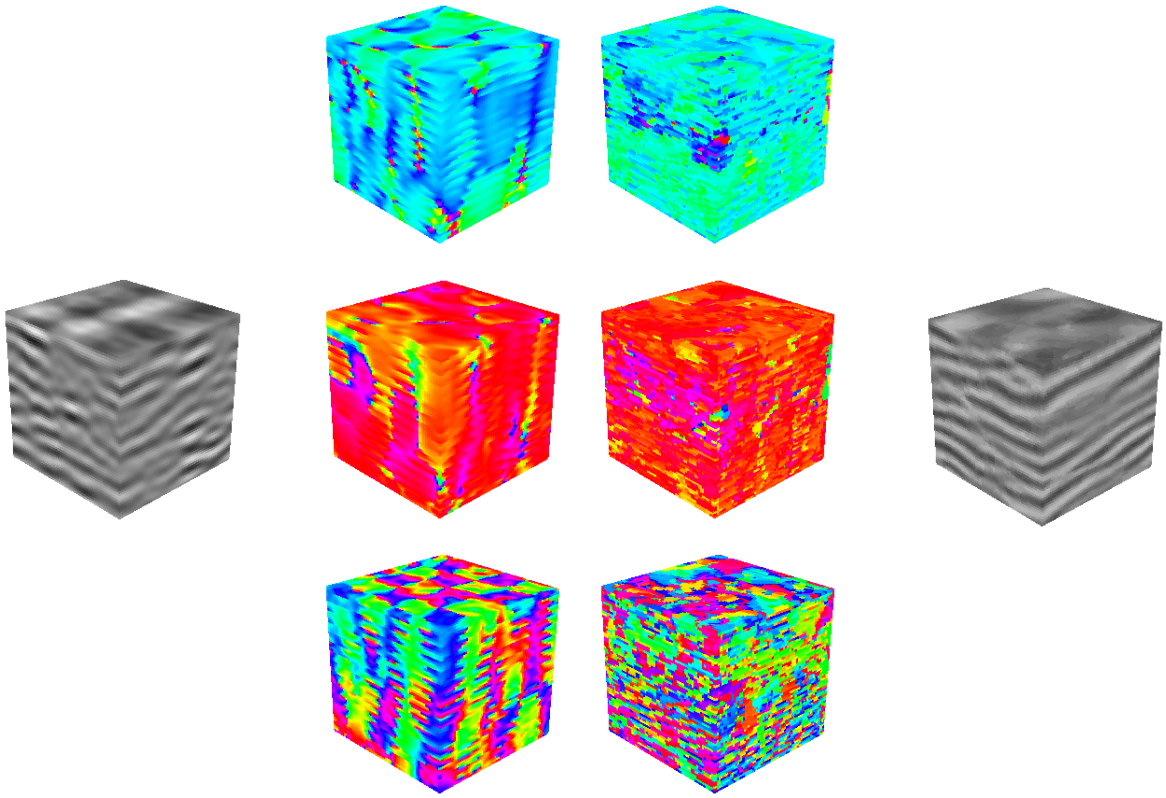


Figure 48 – *Résultat de synthèse sur une texture tridimensionnelle. 1ère colonne : l'échantillon d'entrée, 2ème colonne (de haut en bas) : l'orientation du champ de tenseurs de structure d'entrée montrée par projection sur la vue de face, sur la vue de droite et sur la vue de haut. 3ème colonne : orientation du champ de tenseurs de structure synthétisé, projeté sur les vues de face, de droite et de haut. 4ème colonne : la texture volumique obtenue en utilisant l'approche de synthèse contrainte par champ de tenseurs de structure. Des voisinages de synthèse de taille 11×11×11 avec des pyramides à deux étages et en utilisant l'approche mono-échelle sont utilisés pour la synthèse de la structure et de la texture, respectivement.*

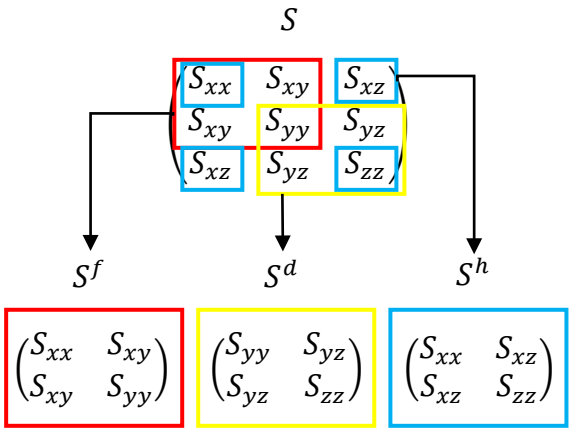


Figure 49 – *Illustration de la projection d'un tenseur de structure 3D en trois tenseurs 2D (S^f), (S^d) et (S^h), appartenant aux plans face, droite et haut, respectivement.*

Dans la Figure 47, l'approche mono-échelle est utilisée avec des voisinages de taille 15×15×15 pour la synthèse de la couche de structure et 11×11×11 pour la synthèse de la texture, tandis que dans la Figure 48, des voisinages de synthèse de taille 11×11×11 avec des pyramides à deux étages et en utilisant l'approche mono-échelle sont utilisés pour la synthèse de la structure et de la texture, respectivement. Trois itérations pour la synthèse du champ de tenseurs, deux itérations pour la synthèse de la texture et la métrique Euclidienne sont utilisées dans les deux figures. Les images d'entrée et de sortie sont de taille 64×64×64.

On peut remarquer dans la Figure 47 que l'approche proposée arrive à reproduire les variations d'orientations de l'échantillon d'entrée et fournit un champ de tenseurs de structure synthétique similaire au champ de tenseurs d'entrée, ce qui aboutit à une texture de sortie de bonne qualité qui semble être générée par le même processus que l'exemplaire.

Au contraire, l'algorithme de synthèse proposé montre des difficultés à conserver la structure de l'exemplaire dans la Figure 48, donnant un champ de tenseurs déformé. Il aboutit finalement à une texture synthétique plus régulière que l'exemplaire et portant des motifs indésirables.

La Figure 50 montre un résultat de synthèse sur un échantillon de texture extrait à partir d'une image sismique tridimensionnelle. La texture volumique d'entrée est montrée à gauche et la texture synthétisée est à droite. Des voisinages de synthèse de taille 11×11×11 avec des pyramides à deux étages et en utilisant l'approche mono-échelle, sont utilisés pour la synthèse de la structure et de la texture, respectivement. Trois et deux itérations sont respectivement utilisées pour la synthèse de la couche de structure et de la couche de texture.

On peut observer que la synthèse contrainte par le champ de tenseurs aboutit à une texture tridimensionnelle très ressemblante à l'échantillon d'entrée. Les motifs texturaux longilignes quasi-horizontaux, se retrouvent dans le bloc de sortie qui semble réaliste et de bonne qualité en termes de préservation de la structure et des dynamiques. Toutefois, certaines structures à très grande échelle ne sont pas restituées (voir les structures très contrastées qui traversent le bloc à mi-hauteur).

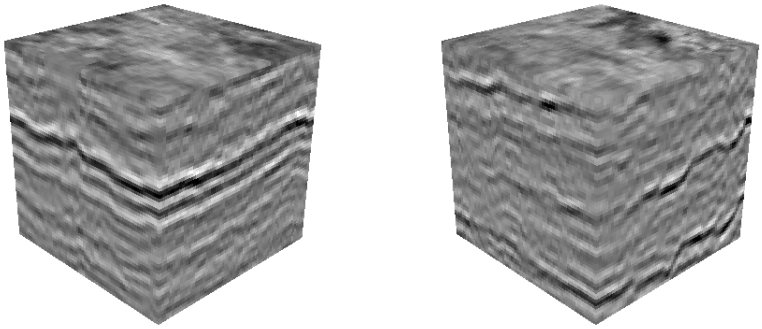


Figure 50 – *Résultat de synthèse sur une texture sismique. A gauche : l'échantillon d'entrée. A droite : la texture synthétisée par l'approche de synthèse à deux étapes structure/texture proposée. Des voisinages de synthèse de taille 11×11×11 avec des pyramides à deux étages et en utilisant l'approche mono-échelle, sont utilisés pour la synthèse de la structure et de la texture, respectivement.*

5.3.3 Inférence 2D/3D du tenseur de structure

Dès lors qu'un échantillon 3D n'est pas disponible [Wei01], la synthèse de texture volumique à partir d'un ou de plusieurs échantillons 2D présente un intérêt, notamment dans le contexte de l'analyse des matériaux [Dac10], comme déjà mentionné (cf. paragraphe 2.3.2). On parle d'approches de synthèse par inférence 2D/3D [Urs14] dont l'objectif est ici de générer une texture tridimensionnelle qui montre une structure plausible et cohérente avec chacun des échantillons d'entrée, qui sont généralement des sections 2D observées avec différentes vues à différentes résolutions.

Pour aboutir à une synthèse 2D/3D à deux étapes structure/texture, une inférence 2D/3D du champ de tenseurs de structure est nécessaire. Il s'agit de modéliser le bloc de tenseurs de sortie pour qu'il corresponde aux caractéristiques des champs de tenseurs d'entrée 2D, suivant les méthodologies présentées dans [Urs13] et [Wei03]. L'algorithme commence donc à partir d'un bruit de tenseurs de structure volumique et procède tenseur par tenseur. Comme dans le cas de synthèse 2D, pour chaque tenseur de sortie, le tenseur d'entrée ayant la meilleure ressemblance est cherché et copié à la position cible en sortie. Afin d'inférer l'information 3D lorsque seule l'information 2D est disponible, on examine les voisinages 2D du tenseur de sortie courant, à partir des trois vues orthogonales (face, droite et haut). Le voisinage de sortie tridimensionnel (VS_{out}^{3D}) est divisé donc en trois voisinages 2D orthogonaux dans la vue de face (VS_{out}^{f}), dans la vue droite (VS_{out}^{d}) et dans la vue haut (VS_{out}^{h}) (Figure 51). Une synthèse multi-2D/2D est ainsi appliquée; pour chaque voisinage, le tenseur ayant la meilleure ressemblance est cherché dans le champ de tenseurs d'entrée de la

vue correspondante, comme un problème de recherche 2D-2D. Plus précisément, chacun des tenseurs 3D (S_{out}) du voisinage (VS_{out}^{f}) est projeté sur le plan de face pour aboutir à un tenseur 2D (S_{out}^{f}) et la recherche de la meilleure ressemblance se fait dans l'échantillon d'entrée 2D correspondant. De même pour les tenseurs 3D des voisinages (VS_{out}^{d}) et (VS_{out}^{h}) à partir desquelles sont extraits les tenseurs 2D (S_{out}^{d}) et (S_{out}^{h}) par projection sur les plans de droite et de haut, respectivement (Figure 49).

Par conséquent, afin de convenablement inférer le champ de tenseurs de sortie en le traitant d'une façon multi-2D, une valeur (i.e. matrice 3×3) résultante de la combinaison appropriée des trois tenseurs 2D trouvés séparément pour chaque vue doit être attribuée au tenseur de structure 3D synthétisé.

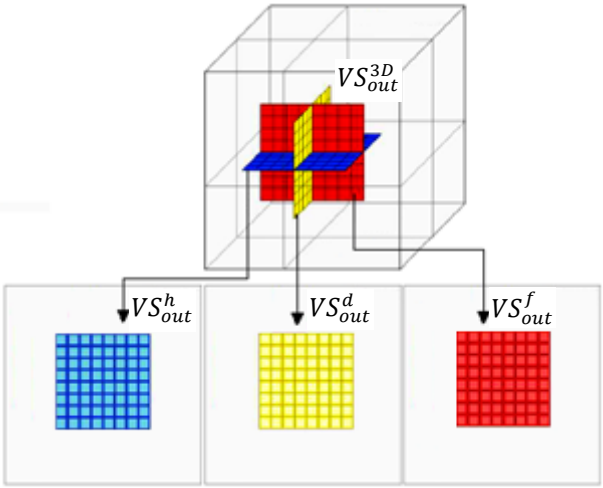


Figure 51 – *Illustration du découpage d'un voisinage de synthèse de tenseurs 3D (VS_{out}^{3D}) en trois voisinages 2D orthogonaux; vue de face (VS_{out}^{f}), vue de droite (VS_{out}^{d}) et vue de haut (VS_{out}^{h}).*

La problématique est donc de combiner les trois tenseurs 2D choisis séparément pour chaque vue et exprimés par :

$$S_{out}^{f} = \begin{pmatrix} S_{xx}^{f} & S_{xy}^{f} \\ S_{xy}^{f} & S_{yy}^{f} \end{pmatrix} \quad , \quad S_{out}^{d} = \begin{pmatrix} S_{yy}^{d} & S_{yz}^{d} \\ S_{yz}^{d} & S_{zz}^{d} \end{pmatrix} \quad , \quad S_{out}^{h} = \begin{pmatrix} S_{xx}^{h} & S_{xz}^{h} \\ S_{xz}^{h} & S_{zz}^{h} \end{pmatrix} \qquad (27)$$

où S_{out}^{f}, S_{out}^{d} et S_{out}^{h} sont les tenseurs trouvés dans les champs de tenseurs d'entrée des vues de face, droite et haut, respectivement.

Une méthode basique consiste à calculer la moyenne des composantes de la diagonale qui se retrouvent deux fois dans la matrice 3×3, comme suit :

$$S_{out}^{3D} = \begin{pmatrix} \frac{1}{2}\left(S_{xx}^{f} + S_{xx}^{h}\right) & S_{xy}^{f} & S_{xz}^{h} \\ S_{xy}^{f} & \frac{1}{2}\left(S_{yy}^{f} + S_{yy}^{d}\right) & S_{yz}^{d} \\ S_{xz}^{h} & S_{yz}^{d} & \frac{1}{2}\left(S_{zz}^{d} + S_{zz}^{h}\right) \end{pmatrix} \tag{28}$$

Toutefois, nous voyons deux problèmes liés à cette solution. D'une part, la combinaison de composantes très différentes de tenseurs peut conduire à un tenseur 3D résultant peu pertinent. En effet, faire une telle combinaison linéaire consisterait à mélanger des tenseurs pouvant être, dans certains cas, très éloignés, car les trois solutions sont recherchées de façon indépendante. Par ailleurs, la matrice 3×3 ainsi obtenue n'est pas nécessairement semi-définie positive (*SDP*). En d'autres termes, elle n'est d'une manière générale pas assimilable à un tenseur de structure.

Bien qu'une méthode de combinaison de tenseurs satisfaisante n'ait pas été développée dans ce mémoire, différentes solutions envisageables sont discutées. Concernant la non pertinence de la matrice obtenue, une solution possible consiste à pondérer différemment les composantes de la diagonale qui se retrouvent deux fois dans la matrice S_{out}^{3D} pour qu'elles aient les valeurs les plus proches possibles. Une recherche conjointe dans les vues face, droite et haut, consistant à chercher les positions *z_f*, *z_d* et *z_h* dans les champs de tenseurs d'entrée qui minimisent la somme des valeurs de dissimilarité des tenseurs donnée dans (29), pourrait peut-être résoudre le problème, au détriment de la charge calculatoire prohibitive associée à une telle recherche.

$$\begin{aligned} SSTD\left(VS_{in}^{f}(z)\middle|VS_{in}^{d}(z)\middle|VS_{in}^{h}(z), VS_{out}^{3D}(z')\right) = \sum_{k=1}^{K}\Big\{ & M\left(S_{in}^{f}(t_k), S_{out}^{f}(t'_k)\right) \\ & + M\left(S_{in}^{d}(t_k), S_{out}^{d}(t'_k)\right) + M\left(S_{in}^{h}(t_k), S_{out}^{h}(t'_k)\right) \Big\}, \end{aligned} \tag{29}$$

où $t_k \in N_K(z)$, $t'_k \in N_K(z')$, K est le nombre de tenseurs de structure dans chaque voisinage, $S_{in}^{v}(t_k)$ (resp. $S_{out}^{v}(t'_k)$) $- v \in \{f, d, h\}$ désigne le $k^{\text{ème}}$ tenseur de structure du voisinage $VS_{in}^{v}(z)$ (resp. $VS_{out}^{v}(z')$) et M est l'une des quatre métriques : M_{SED}, $M_{SO,\alpha}$, M_{FN} et M_{LE} (cf. paragraphe 3.3.2).

Une approche alternative non optimale consiste à faire les trois recherches de façon séquentielle, de sorte que la solution obtenue pour la vue de face contraigne la recherche en vue de droite, et les solutions face et droite contraignent la recherche en vue du haut.

Concernant l'exigence d'une matrice semi-définie positive (*SDP*), l'application d'une pondération qui garantisse le caractère *SDP* serait peut-être la meilleure solution. Néanmoins, une telle pondération n'est pas facile à établir. Il serait donc intéressant de chercher la matrice *SDP* la plus proche de S_{out}^{3D}, par projection dans l'espace des matrices 3×3 semi-définies positives.

5.4 Synthèse de texture contrainte par champ de structure arbitraire

Cette partie présente une extension de l'algorithme de synthèse structure/texture permettant de générer des textures de formes arbitraires, et donc pas forcément stationnaires, à partir d'un exemplaire d'entrée. L'algorithme procède en trois étapes : (i) la texture initiale est utilisée pour construire un dictionnaire de champs d'intensités et de tenseurs; (ii) un champ de tenseurs de structure *contrainte* est construit à partir d'une image ou d'une fonction de base choisie arbitrairement, et dite *texture arbitraire*; (iii) le champ de tenseurs *contrainte* obtenu est utilisé pour contraindre la synthèse de la texture en cherchant la meilleure ressemblance au sein du dictionnaire (Figure 52). Trois méthodes différentes pour la construction du champ de tenseurs de structure *contrainte* sont proposées. Afin d'évaluer si l'algorithme reproduit fidèlement l'aspect visuel de l'exemplaire, plusieurs tests sont effectués avec différentes textures arbitraires. Notons que cette approche de synthèse 2D est un algorithme préliminaire ouvrant la voie vers la synthèse de textures tridimensionnelles de formes arbitraires, ce qui sera utile pour la conception de matériaux virtuels (matériaux carbonés par exemple).

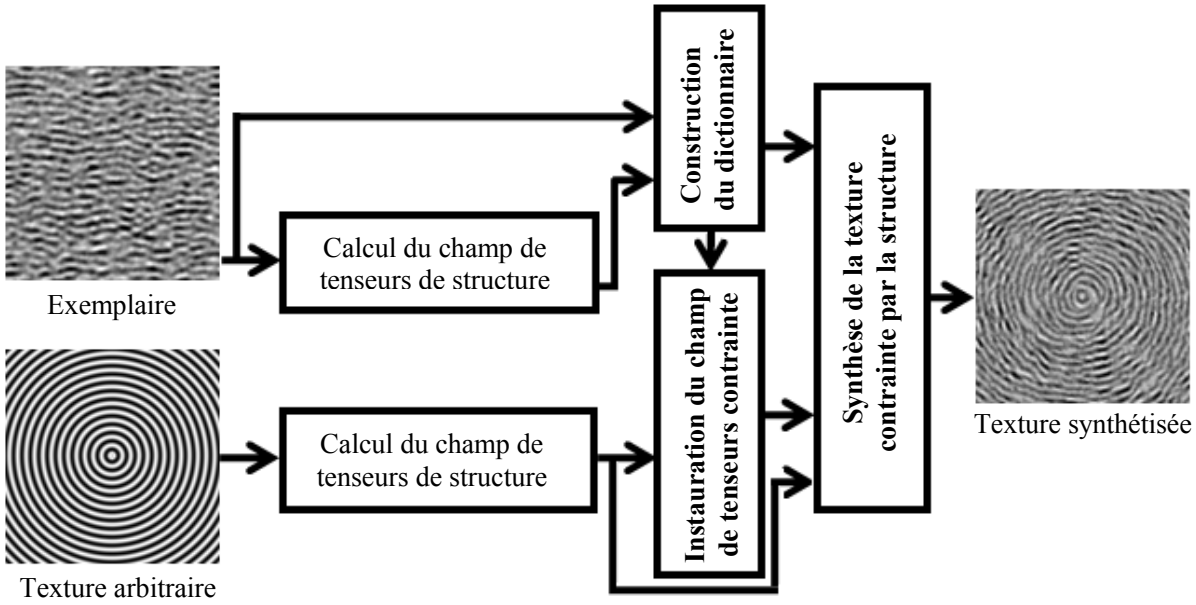


Figure 52 – *Illustration de l'algorithme de synthèse structure/texture permettant de générer des textures de formes arbitraires.*

5.4.1 Construction du dictionnaire de recherche

La première étape de l'algorithme consiste à construire un dictionnaire à valeurs d'intensité et à valeurs tensorielles à partir de la texture d'entrée I_{in} et de son champ de tenseurs S_{in}.

Soit $D(I_{in})$ le dictionnaire créé à partir de I_{in} : $D(I_{in}) = \{V_k\ , k = 1 \text{ à } K\}$ un ensemble de K voisinages bidimensionnels V_k (par exemple carrés ou causaux en forme de « L ») où K est le nombre de sites de I_{in} correspondant à un voisinage complet. Chaque voisinage V_k est tel que :

$$V_k = (VI_k, VS_k)\ , \tag{30}$$

où VI_k est le $k^{\text{ème}}$ voisinage d'intensité dans I_{in} et VS_k le voisinage de tenseurs de structure correspondant, dans le champ de tenseurs S_{in}.

Afin d'enrichir ce dictionnaire avec des configurations d'orientation non présentes dans l'exemplaire, $D(I_{in})$ est remplacé par un dictionnaire $D_M(I_{in})$ incluant $D(I_{in})$. $D_M(I_{in})$ est constitué de voisinages extraits d'images $\{Rot_m(I_{in}), m = 0, \dots, M-1\}$ où $Rot_m(I_{in})$ est la rotation de I_{in} d'un angle $\varphi_m = 2\pi m/M$ ainsi que des champs de tenseurs de structure correspondants. Notons que la rotation appliquée s'appuie sur une interpolation par sinus cardinal.

La première ligne de la Figure 53 illustre un exemple de création de différentes versions de textures tournées à partir de l'exemplaire en utilisant un pas angulaire $\varphi_m = 10°$. Les deuxième et troisième lignes présentent respectivement, les images d'orientation et les images de cohérence des champs de tenseurs correspondants (i.e. calculés sur les textures de la première ligne). Ces images sont à la base de la création du dictionnaire utilisé dans le processus de synthèse.

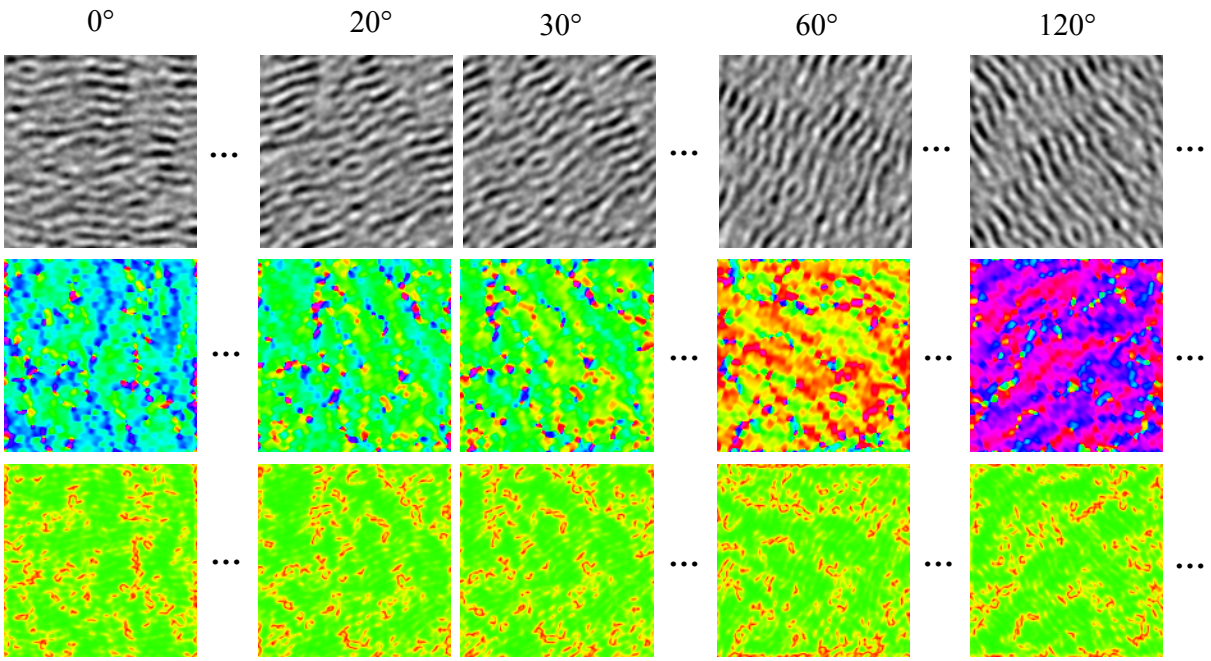


Figure 53 – *Exemple de textures et de champs de tenseurs tournés avec un pas angulaire* $\varphi_m = 10°$. *1ère ligne (de gauche à droite) : l'exemplaire* (0°)*, les versions tournées de* 20°, 30°, 60° et 120°. *2ème et 3ème ligne : images d'orientation et de cohérence des champs de tenseurs correspondants aux textures de la 1ère ligne.*

5.4.2 Construction de la contrainte et processus de synthèse

Le dictionnaire de recherche étant construit, l'étape suivante de notre algorithme de synthèse consiste à construire un champ de tenseurs de structure contrainte S_{con} à partir d'une texture arbitraire imposée I_{arb}. Ce champ de tenseurs sera utilisé ensuite pour contraindre la synthèse de la texture en cherchant la meilleure ressemblance au sein du dictionnaire. Trois méthodes différentes ont été envisagées pour le calcul de S_{con} :

- Utilisation du champ de tenseurs de la texture arbitraire comme champ de tenseurs contrainte (*A_FS*). Il s'agit donc de calculer le champ de tenseurs de structure S_{arb} et d'appliquer $S_{con} = S_{arb}$. Cette méthode a été évoquée dans [Akl15].
- Instauration du champ de tenseurs contrainte par modulation (*M_FS*). Cette méthode, détaillée dans [Akl15], opère en trois parties :
  - Calcul du champ de tenseurs S_{in} de la texture d'entrée I_{in}.
  - Calcul du champ de tenseurs de structure S_{arb} de la texture arbitraire I_{arb} et du champ d'orientation O_{arb} associé.
  - Modulation des orientations de S_{in} par O_{arb}.

- Instauration du champ de tenseurs contrainte par synthèse (*CT_LS*). Cette méthode s'avère la plus efficace (cf. Annexe M) et est décrite ci-dessous.

Du fait que seule l'orientation est modulée, l'approche *M_FS* fournit des résultats satisfaisants tant que l'incrément angulaire de la texture arbitraire n'est pas trop important. En revanche, il s'avère difficile d'aboutir à une synthèse fidèle quand l'orientation s'écarte largement de celle de l'exemplaire (cf. Annexe M). La méthode *CT_LS* offre une alternative en synthétisant le champ de tenseurs contrainte S_{con} à partir du dictionnaire $D_M(I_{in})$ en y considérant uniquement les voisinages de tenseurs de structure. Il s'agit de limiter la recherche de la ressemblance dans le dictionnaire $D_M(I_{in})$ au champ de tenseurs tourné selon le même angle que celui imposé par une carte d'orientation *consigne* O_{csg} issue du champ de tenseurs S_{arb} de la texture arbitraire imposée I_{arb}.

On définit $D_M(S_{in})_{\mathrm{Agl}(z')}$ comme étant le champ de tenseurs du dictionnaire $D_M(I_{in})$ tourné selon l'angle $O_{csg}(z')$ imposé par la carte d'orientation consigne O_{csg}. $D_M(VS_{in})_{\mathrm{Agl}(z')}(z)$ désigne le voisinage de tenseurs de structure à la position z de $D_M(S_{in})_{\mathrm{Agl}(z')}$.

On opère donc en deux étapes :

- Calcul du champ de tenseurs de structure S_{arb} de la texture arbitraire I_{arb} et du champ d'orientation consigne O_{csg} associé.
- Synthèse du champ de tenseurs de structure contrainte S_{con} à partir du dictionnaire $D_M(I_{in})$ en utilisant le champ d'orientation consigne O_{csg} pour pointer à la partie de $D_M(I_{in})$ uniquement concernée dans la recherche de la meilleur ressemblance :
  S_{con} est initialisé par un bruit de tenseurs de structure et modifié durant la synthèse par recherche de la meilleure ressemblance pour chaque position z'=(x, y). Cette recherche se fait uniquement dans $D_M(S_{in})_{\mathrm{Agl}(z')}$, et le tenseur correspondant $S_{k(min)}$ est copié à la position cible $S_{con}(z')$.

Le champ de tenseurs *contrainte* S_{con} étant créé, le processus de synthèse de la texture contrainte par S_{con} est similaire à notre algorithme de synthèse de base (cf. paragraphe 4.2), à l'exception que la meilleure ressemblance est uniquement cherchée dans la partie du dictionnaire $D_M(I_{in})$ tournée selon l'angle $O_{csg}(z')$ et comprenant des voisinages de tenseurs de structure et à valeurs d'intensités.

Dans l'étape de synthèse de S_{con} ainsi que pour la synthèse de la texture contrainte par S_{con}, l'approche multi-échelle par décomposition en pyramides Gaussiennes est également judicieuse, surtout dans le cas de textures contenant des structures à grande échelle. Le dictionnaire $D_M(I_{in})$ est ainsi généré à l'aide de voisinages multi-échelles extraits dans les pyramides d'intensité et de tenseurs construites à partir des M textures tournées et de leur champ de tenseurs correspondants. De même, des pyramides de textures et de tenseurs sont construites à partir de I_{out} et S_{con}, respectivement. L'algorithme (*CT_LS*) de synthèse du champ de tenseurs *contrainte* et de la texture est résumé à la Figure 54.

Notons que dans les différentes approches d'instauration du champ de tenseur *contrainte*, et dans les approches de synthèse de la texture, la texture arbitraire imposée peut être remplacée par une fonction analytique. Le champ d'orientation O_{csg} utilisé comme consigne pourrait lui-même être défini de façon analytique. L'utilisation d'une texture arbitraire serait dans ce cas inutile.

Entrée: I_{in} & I_{arb}

$D_M(I_{in}) \leftarrow$ CREATIONDICTIONNAIRE (I_{in})

$S_{arb} \leftarrow$ CHAMPTENSEURSDESTRUCTURE (I_{arb})

$O_{csg} \leftarrow$ CARTEORIENTATION (S_{arb})

$S_{in} \leftarrow$ CHAMPTENSEURSDESTRUCTURE (I_{in})

$S_{con} \leftarrow$ INITIALISATIONBRUITTENSEURSDESTRUCTURE (S_{in})

Tant que iter ≤ nbr_itérations_struct

pour chaque position *z'* = (*x*, *y*) de S_{con}

Extraire $VS_{con}(z')$

$\mathrm{Agl}(z') \leftarrow$ EXTRACTIONANGLE ($O_{csg}(z')$)

$k(min) \leftarrow argmin_z\{SSTD\ (D_M(VS_{in})_{\mathrm{Agl}(z')}(z)\ ,\ VS_{con}(z'))\}$

$S_{con}(z') \leftarrow S_{k(min)}$

fin pour

Incrémenter iter

Fin tant que

$I_{out} \leftarrow$ INITIALISATIONBRUITINTENSITE (I_{in})

Tant que iter ≤ nbr_itérations

pour chaque position *z'* = (*x*, *y*) de I_{out}

Extraire $VI_{out}(z')$ et $VS_{con}(z')$

$\mathrm{Agl}(z') \leftarrow$ EXTRACTIONANGLE ($O_{csg}(z')$)

$k(min) \leftarrow argmin_z\{\gamma(z_{\mathrm{Agl}(z')}, z')\}$

$I_{out}(z') \leftarrow I_{k(min)}$

fin pour

Incrémenter iter

Fin tant que

Sortie: I_{out}

Figure 54 – *Pseudocode de l'algorithme de synthèse du champ de tenseurs contrainte et de la texture (CT_LS).*

5.4.3 Résultats

Cette partie montre les résultats de l'algorithme de synthèse *CT_LS* proposé. Dans un premier temps, l'effet du champ de tenseurs *contrainte* instauré ainsi que l'influence du pas angulaire utilisé pour la construction du dictionnaire de recherche seront étudiés. Des résultats obtenus en utilisant différentes textures arbitrairement imposées seront présentés dans un second temps.

5.4.3.1 Pertinence du champ de tenseurs *contrainte*

Afin de mesurer l'utilité de la couche de structure lors de la synthèse de la texture de forme arbitraire, la Figure 55 montre des résultats de synthèse avec différentes valeurs du poids de pondération structure/texture. La première ligne montre, de gauche à droite, un exemplaire de matériau et son champ d'orientation, une texture arbitraire concentrique, son champ d'orientation et l'image d'orientation du champ de tenseurs de structure *contrainte* synthétisé par l'approche *CT_LS* en utilisant un voisinage causal de taille 11×11 avec l'approche mono-échelle. Les deuxième à quatrième lignes montrent, de gauche à droite et de haut en bas, les textures synthétisées en utilisant un poids de pondération structure/texture (cf. équation (16)) p = 1, 0.95, 0.85, 0.8, 0.7, 0.5, 0.3, 0.1 et 0. Les textures d'entrée et de sortie sont de taille 256×256 et 512×512, respectivement. La métrique Euclidienne est utilisée et les images montrées sont obtenues après deux itérations. On peut remarquer dans la Figure 55 que la synthèse W&L pure (i.e. $p = 1$) n'arrive pas à bien reproduire les motifs de l'échantillon d'entrée et aboutit à une texture ayant des motifs plus réguliers que l'exemplaire, surtout au coin supérieur gauche de l'image. Bien que son aspect général reste correct, la texture de sortie contient des défauts qui apparaissent comme des paquets de lignes quasi parfaites et parallèles, et des plages de niveau de gris quasi constant, sans structure.

On observe les mêmes caractéristiques avec p = 0.95 et p = 0.85, avec une performance légèrement améliorée. Les défauts semblent avoir disparu avec p = 0.8, ce qui est dû à l'utilisation du champ de structure. L'aspect général de la texture de sortie ne change pas vraiment entre p = 0.7 et p = 0.3. Avec p = 0.1, des dislocations et des changements brutaux de contraste commencent à apparaître. Le nombre et la rugosité de ces défauts augmentent avec p = 0, où le pixel de sortie est basé uniquement sur l'information de structure. Cet exemple montre que la texture de sortie est mieux synthétisée en considérant la couche de structure représentée par le champ de tenseur *contrainte*, par rapport au cas où

l'information de niveau de gris est uniquement considérée. Ceci à condition que la couche de structure synthétisée soit de bonne qualité.

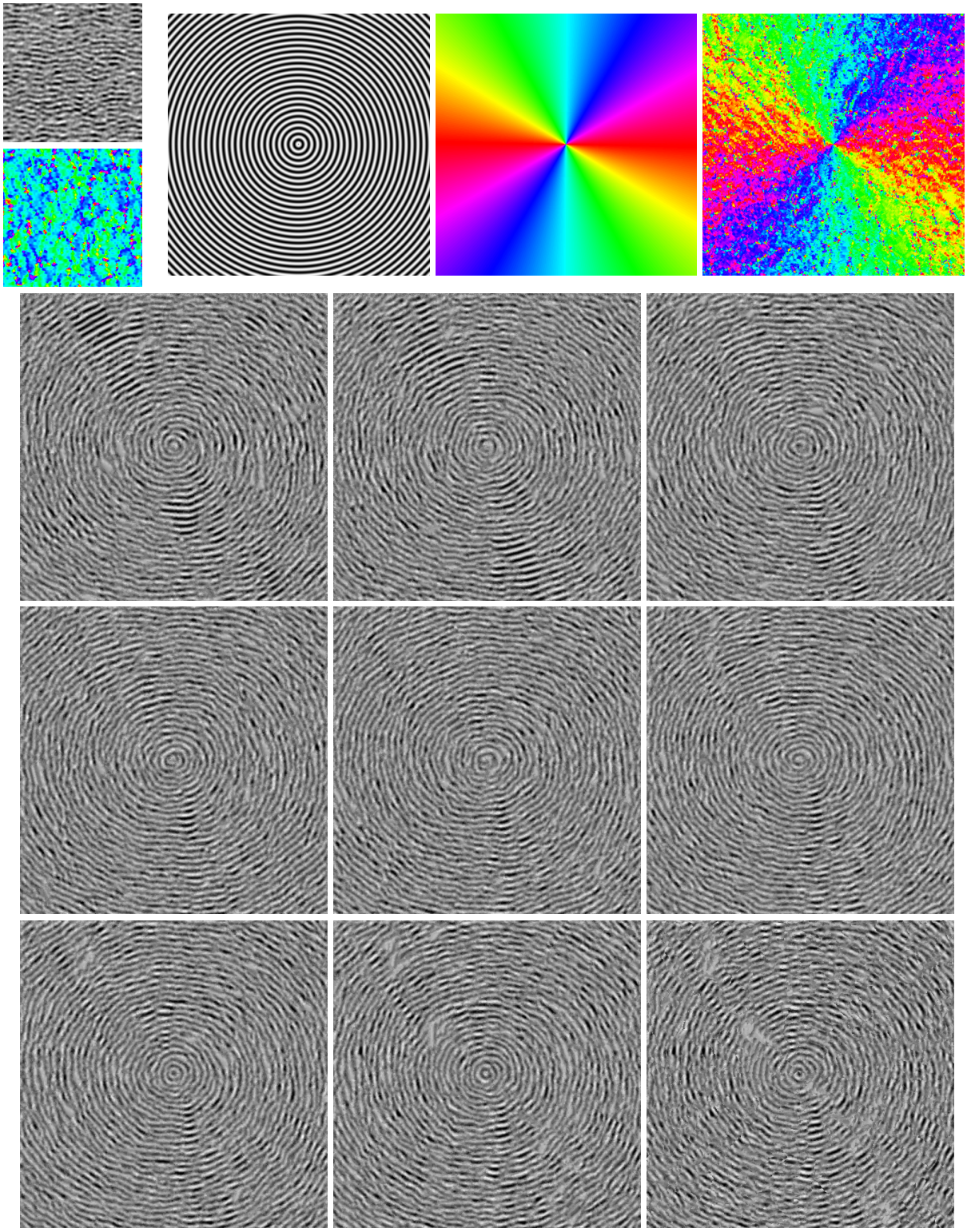


Figure 55 – *Résultats de synthèse avec différentes valeurs du poids de pondération structure/texture. 1ère ligne (de gauche à droite) : l'exemplaire et son champ d'orientation, la texture arbitraire, son orientation et l'orientation du champ de tenseurs synthétisé par l'approche CT_LS. 2ème à 4ème lignes (de gauche à droite et de haut en bas) : les textures synthétisées avec p = 1, 0.95, 0.85, 0.8, 0.7, 0.5, 0.3, 0.1 et 0.*

5.4.3.2 Effet du pas angulaire du dictionnaire de recherche

La Figure 56 montre des résultats obtenus à l'aide de la méthode *CT_LS* à partir de la même texture d'entrée, et en utilisant trois valeurs différentes pour le pas angulaire. La première ligne montre, de gauche à droite, un exemplaire de matériau et son champ d'orientation, la texture arbitraire concentrique et son champ d'orientation. Les deuxième et troisième lignes montrent de gauche à droite, les orientations du champ de tenseurs *contrainte* obtenu et les textures résultantes en utilisant *CT_LS* avec un pas angulaire de 45°, 20° et 5°, respectivement. Un voisinage de taille 11×11 est utilisé avec l'approche mono-échelle, pour la synthèse du champ de tenseurs et pour la synthèse de la texture. La métrique Euclidienne est utilisée et les images montrées sont obtenues après deux itérations.

On peut remarquer, dans la première colonne de la Figure 56, que le champ de tenseurs synthétique obtenu avec un pas angulaire de 45° est distordu et assez différent de celui de la texture arbitraire. La texture de sortie suit les orientations du champ de tenseurs *contrainte* et présente une structure octogonale contenant des motifs déformés. La raison à cela est que le dictionnaire de recherche construit avec un pas de 45°, n'inclue pas les versions de l'exemplaire et leurs champs de tenseurs ayant les orientations suffisantes à la reproduction de la nature concentrique de la texture arbitraire. En utilisant un pas angulaire de 20°, les défauts disparaissent du champ de tenseurs synthétisé. Bien que la texture résultante soit de meilleure qualité que celle obtenue avec le pas de 45°, des motifs indésirables et qui n'existent pas dans l'échantillon d'entrée sont observés.

Au contraire, la synthèse avec un pas angulaire de 5° conduit à un champ de tenseurs similaire à celui de la texture arbitraire, et par la suite, la texture de sortie habillée par les niveaux de gris de l'exemplaire, est fidèle à l'orientation de la texture arbitraire et aux caractéristiques de l'échantillon d'entrée. Ces différentes observations montrent qu'il est indispensable de bien adapter le pas angulaire utilisé dans le processus de construction du dictionnaire de recherche aux données d'entrée de l'algorithme de synthèse.

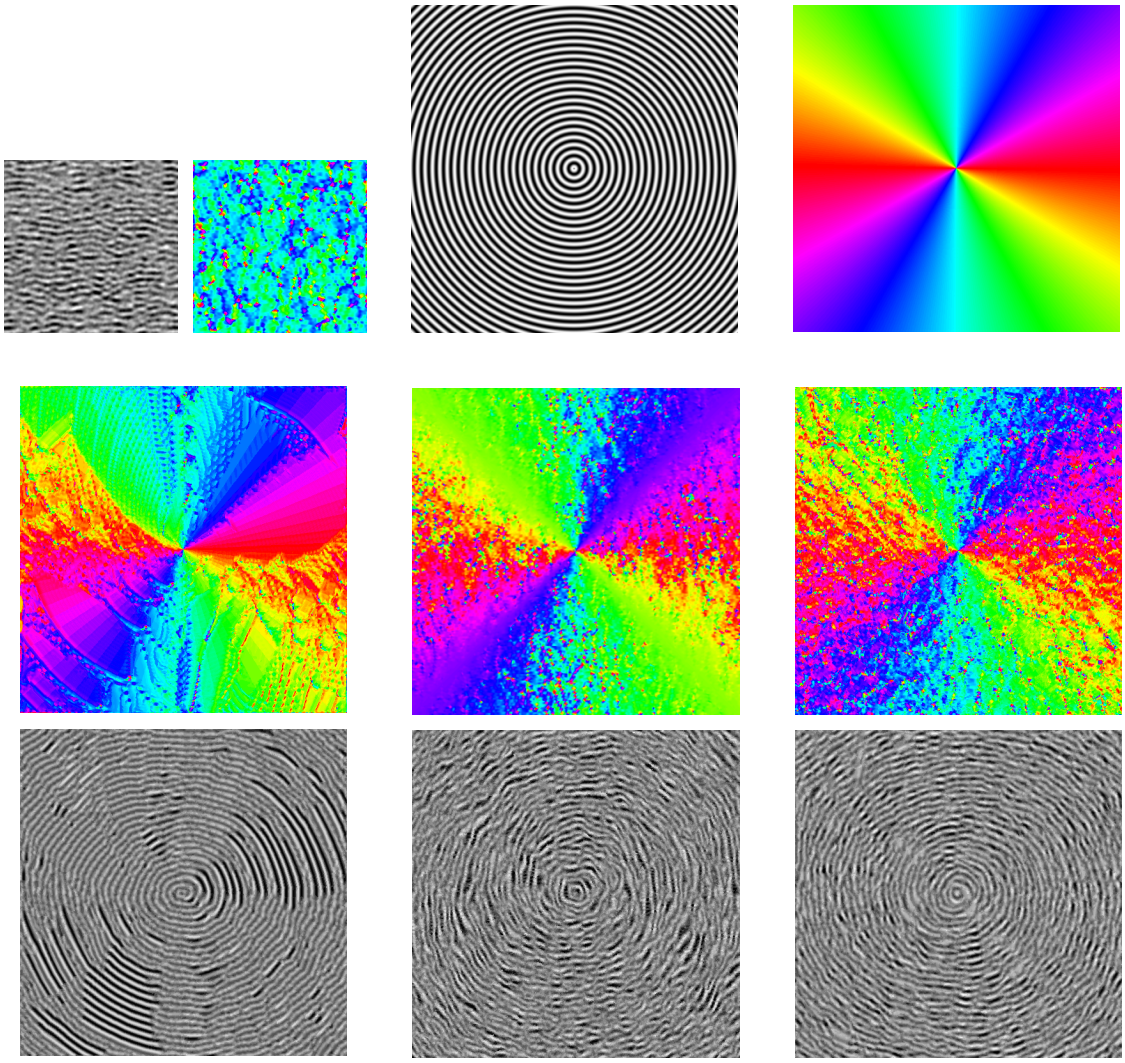


Figure 56 – *Résultats de synthèse avec différentes valeurs du pas angulaire. 1ère ligne (de gauche à droite) : l'exemplaire et son champ d'orientation, la texture arbitraire et son d'orientation. 2ème et 3ème lignes (gauche à droite) : l'orientation du champ de tenseurs contrainte obtenu et les textures résultantes en utilisant CT_LS avec un pas angulaire de 45°, 20° et 5°*, respectivement. *Les images sont obtenues après 2 itérations.*

5.4.3.3 Exemples de résultats

Le but de cette partie est d'analyser la capacité de l'approche de synthèse *CT_LS* à synthétiser des textures de formes arbitraires différentes. La Figure 57 montre des résultats obtenus en utilisant un échantillon de matériaux comme exemplaire et une texture arbitraire en forme de vagues. La première ligne montre l'échantillon d'entrée (à gauche) et son image d'orientation (à droite). La deuxième ligne montre la texture arbitrairement imposée (à gauche) et l'image d'orientation de son champ de tenseurs de structure (à droite). La troisième ligne montre les orientations du champ de tenseurs *contrainte* synthétisé (à gauche) et la texture résultante (à droite).

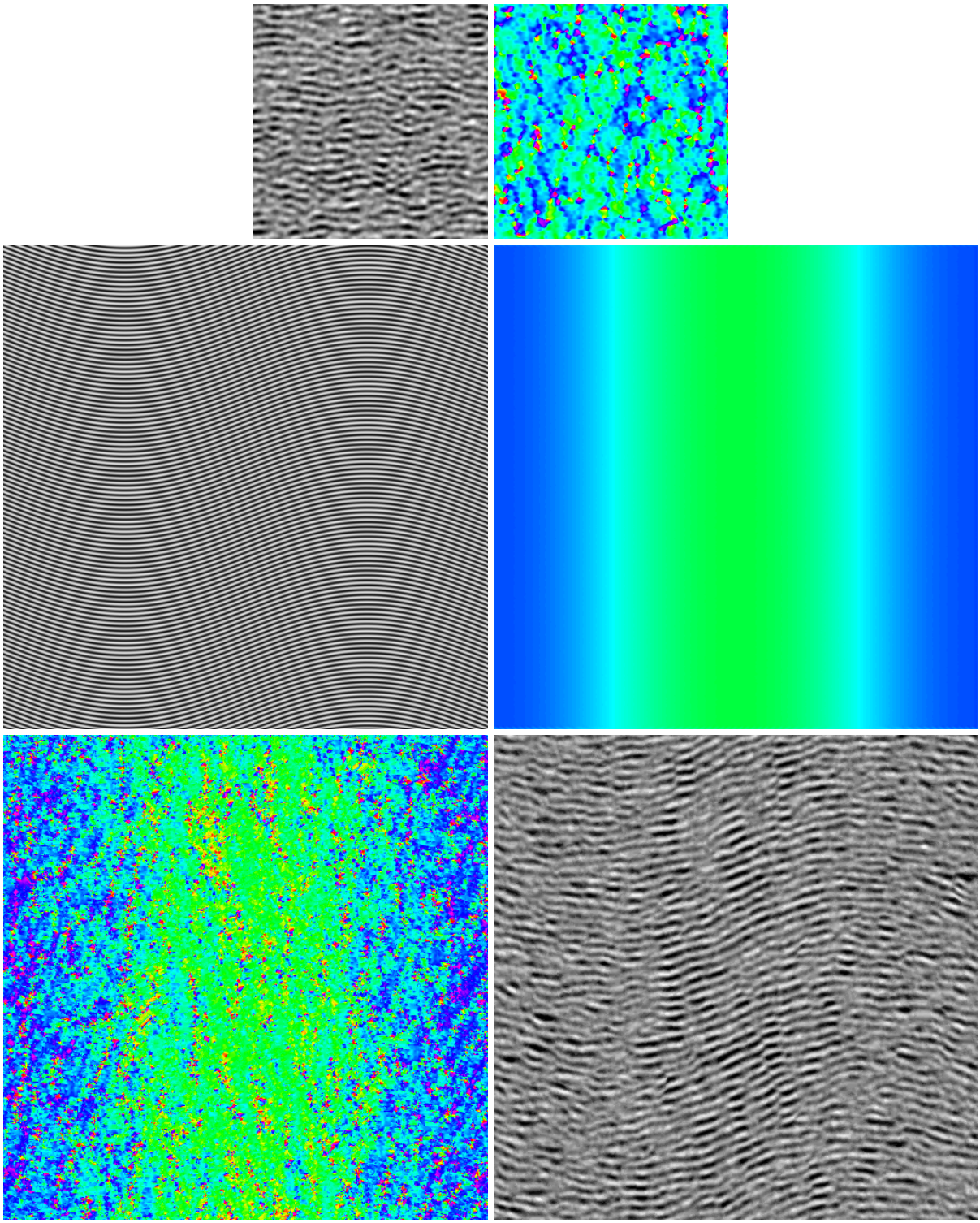


Figure 57 – *Résultats de synthèse en utilisant un échantillon de matériau comme exemplaire. 1ère ligne : échantillon d'entrée (à gauche) et son image d'orientation (à droite). 2ème ligne : texture arbitraire en forme de vagues (à gauche) et son image d'orientation (à droite). 3ème ligne : l'image d'orientation du champ de tenseurs contrainte synthétisé (à gauche) et la texture synthétisée (à droite). Un pas angulaire de 1° est utilisé et les images sont obtenues après 3 itérations.*

Un voisinage causal de taille 9×9 en utilisant des pyramides Gaussiennes à deux étages et avec l'approche mono-échelle, sont utilisés pour la synthèse du champ de tenseurs et pour la synthèse de la couche de texture, respectivement. Le dictionnaire de recherche est

construit avec un pas angulaire de 1°, et les images montrées sont obtenues après trois itérations.

On peut remarquer dans la Figure 57 que l'approche proposée réussit à synthétiser, à partir du champ de tenseurs d'entrée, un champ de tenseurs qui représente la structure de la texture arbitraire. L'algorithme de synthèse contrainte par le champ de structure obtenu est donc capable de synthétiser la texture de sortie ayant la structure spécifiée par la texture arbitraire, en l'habillant par les motifs de l'échantillon d'entrée. Toutefois, la texture synthétique présente en son milieu quelques défauts structuraux; les franges longitudinales dans la région du centre sont un peu plus longues que celles de l'exemplaire.

La Figure 58 présente un résultat de synthèse en utilisant des images de matériaux composites tissés comme exemplaire et texture arbitraire. La première ligne montre l'exemplaire (à gauche) et son image d'orientation (à droite). La deuxième ligne montre la texture arbitraire (à gauche) et son image d'orientation (à droite) utilisée comme carte de consigne dans le processus de synthèse de la texture. La troisième ligne montre l'orientation du champ de tenseurs contrainte synthétisé (à gauche) et la texture résultante (à droite). La couche de structure est synthétisée en utilisant des pyramides à deux étages tandis que l'approche mono-échelle est utilisée pour la synthèse de la texture. Un voisinage causal de taille 11×11, un pas angulaire de 1° et deux itérations sont utilisés dans les deux étapes de l'algorithme.

On peut observer dans la Figure 58 que l'approche de synthèse proposée aboutit à un champ de structure maintenant les tenseurs de l'exemplaire et représentant la structure de la texture arbitraire. La synthèse de la couche de texture contrainte par le champ de tenseurs obtenu arrive à générer une texture ayant la forme de la texture arbitraire, en reproduisant les motifs de l'exemplaire. Cette texture synthétisée respecte les dynamiques et les caractéristiques essentielles de l'exemplaire.

Ces résultats prometteurs obtenus ouvrent la voie vers la synthèse de textures 3D de formes arbitraires, ce qui sera intéressant dans l'analyse et la conception de matériaux carbonés.

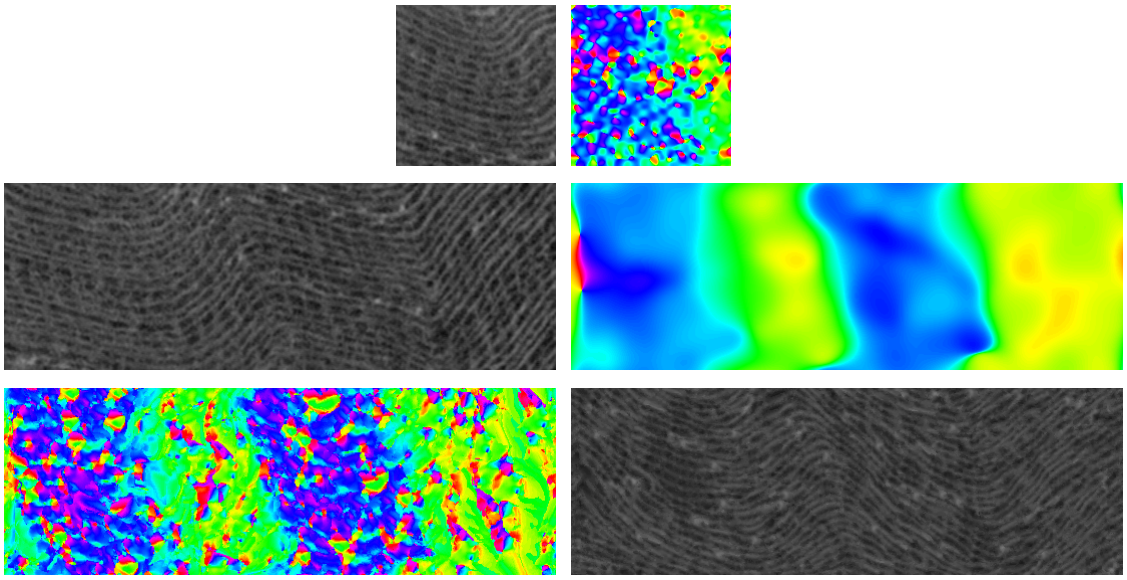


Figure 58 – *Résultats de synthèse en utilisant deux différents échantillons de matériau carboné comme exemplaire et texture arbitraire. 1ère ligne : l'exemplaire (à gauche) et son image d'orientation (à droite). 2ème ligne : la texture arbitraire (à gauche) et son image d'orientation (à droite). 3ème ligne : l'orientation du champ de tenseurs synthétisé (à gauche) et la texture résultante (à droite). Le champ de tenseurs est synthétisé en utilisant des pyramides à deux étages tandis que l'approche mono-échelle est utilisée pour la synthèse de la texture.*

La Figure 59 présente un résultat de synthèse montrant la capacité de l'algorithme proposé à synthétiser des lettres, en respectant les caractéristiques de l'échantillon d'entrée. La première ligne montre la texture d'entrée (à gauche) et son image d'orientation (à droite). La deuxième ligne montre la texture arbitraire (à gauche) contenant des lettres et son image d'orientation (à droite) utilisée comme carte de *consigne* dans le processus de synthèse de la texture. La troisième ligne montre l'image d'orientation du champ de tenseurs *contrainte* synthétisé (à gauche) et la texture résultante (à droite). L'approche mono-échelle est utilisée avec un voisinage causal de taille 11×11 et 7×7 pour la synthèse de la couche de structure et de la couche de texture, respectivement. La métrique Euclidienne est utilisée avec un pas angulaire de 1°. L'image d'orientation du champ de tenseurs synthétique et la texture de sortie sont obtenues après deux itérations.

On peut observer dans la Figure 59 que la méthode proposée réussit à générer un champ de tenseurs qui représente les structures de la texture arbitraire. Par conséquent, la synthèse de la couche de texture arrive à reproduire les lettres de la texture arbitraire en respectant les caractéristiques et l'aspect visuel de l'échantillon d'entrée anisotrope.

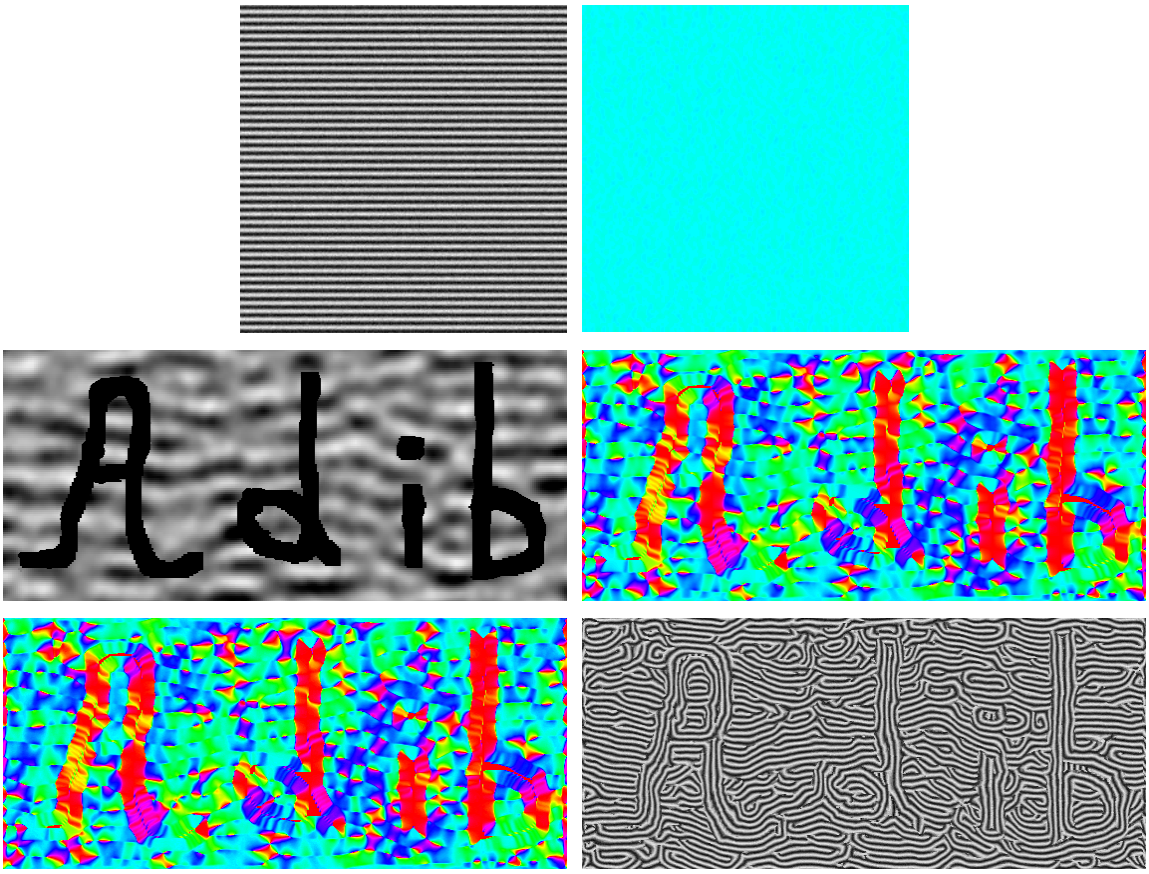


Figure 59 – *Résultats de synthèse en utilisant une texture arbitraire contenant des lettres. 1ère ligne : l'exemplaire (à gauche) et son image d'orientation (à droite). 2ème ligne : la texture arbitraire (à gauche) et son image d'orientation (à droite). 3ème ligne : l'orientation du champ de tenseurs synthétisé (à gauche) et la texture résultante (à droite). L'approche mono-échelle est utilisée avec un voisinage causal de taille 11×11 et 7×7 pour la synthèse de la couche de structure et de la couche de texture, respectivement.*

5.5 Conclusion

Ce chapitre a présenté trois extensions différentes de l'algorithme de synthèse structure/texture proposé. Dans la première partie du chapitre, une extension de l'algorithme de synthèse à deux étapes pour synthétiser des textures couleur est développée. Il s'agit de calculer le champ de tenseurs de structure d'une texture couleur sur la composante de luminance ou bien d'additionner les tenseurs des différents plans *R*, *G* et *B*. La métrique utilisée pour chercher la meilleure ressemblance dans la texture et le champ de tenseurs d'entrée, combine donc la somme des valeurs de dissimilarité des tenseurs et la somme des erreurs quadratiques calculées sur les champs *R*, *G* et *B*. Les résultats de synthèse montrent que l'approche proposée arrive à bien synthétiser le champ de tenseurs *contrainte* et par conséquent à générer une texture de sortie satisfaisante, en termes de structure et de couleur. La méthode proposée réussit à reproduire les orientations de l'exemplaire même dans le cas où la synthèse W&L pure, ne parvient pas à respecter la structure de l'entrée.

En outre, l'approche de synthèse accélérée à l'aide de l'interpolation bilinéaire, aboutit à une réduction significative de la charge de calcul par rapport à l'algorithme de synthèse classique de W&L.

L'extension de l'algorithme de base 2D, à la synthèse de textures 3D est traitée dans la deuxième partie du chapitre. La problématique d'inférence 2D/3D du tenseur de structure, dans le cas de synthèse d'une texture volumique à partir d'un ou de plusieurs exemplaires 2D, est également abordée. Il s'agit de combiner d'une manière appropriée, trois tenseurs de structure 2D pour aboutir à un tenseur 3D qui représente l'orientation et l'anisotropie du bloc 3D, tout en garantissant la propriété de semi-définie positivité du tenseur de structure 3D.

La dernière partie du chapitre présente une variante de notre algorithme de base qui permet de synthétiser des textures de formes arbitraires voire non stationnaires. L'algorithme commence par construire un dictionnaire de recherche à partir de l'échantillon d'entrée et de son champ de tenseurs de structure. Une approche est par la suite proposée pour synthétiser un champ de tenseurs qui suit les structures du champ de tenseurs d'une texture arbitrairement imposée. Le champ de tenseurs synthétisé est finalement utilisé pour contraindre la synthèse de la texture. Les résultats obtenus montrent la capacité de l'algorithme proposé à synthétiser des textures de formes spécifiées, tout en préservant les caractéristiques et les aspects visuels de l'exemplaire.

Chapitre 6
Conclusion

L'objectif des travaux exposés dans ce mémoire est la synthèse de texture en utilisant la couche de structure. L'outil choisi pour la représentation de la couche de structure est le tenseur de structure qui s'appuie sur une estimation du gradient et permet une extraction fiable des informations telles que l'orientation locale, l'énergie et le degré d'anisotropie en tout point de la texture.

Guidés par l'exigence de la reproduction fidèle des structures de la texture sous-jacente, nous avons proposé une approche de synthèse en deux étapes structure/texture : la couche de structure représentée par le champ de tenseurs de structure local est synthétisée dans la première étape, et utilisée pour contraindre la synthèse de la texture, dans la deuxième étape. L'approche de Wei et Levoy a été utilisée dans les deux étapes de synthèse, en l'adaptant aux spécificités des valeurs tensorielles des images par l'utilisation de différentes métriques de mesures tensorielles et par le choix de décompositions multi-échelles pertinentes. L'adoption d'une approche d'accélération fondée sur l'interpolation bilinéaire a abouti à une réduction significative de la complexité calculatoire. Une accélération logicielle par parallélisation au moyen d'une carte graphique (GPU) a également été mise en œuvre.

L'application de l'algorithme proposé sur des textures synthétiques et naturelles a permis de montrer l'influence des différents paramètres de synthèse. Une démarche de comparaison objective avec différentes approches existantes, fondée sur les statistiques d'ordre 1 et 2 des niveaux de gris et des orientations locales, a montrée d'excellents résultats en termes de reproduction des orientations de l'échantillon d'entrée et du respect de ses dynamiques. Un atout majeur de la méthode proposée est sa capacité à aboutir à des résultats de même voire de meilleure qualité que ceux obtenus par les algorithmes classiques et sa capacité à synthétiser avec succès la texture de sortie dans des situations difficiles pour lesquelles les algorithmes existants ne parviennent pas à reproduire la structure de l'exemplaire.

La dernière partie de ce mémoire était consacrée aux extensions de l'algorithme de synthèse structure/texture. La première extension traite de la synthèse de texture couleur et se

fonde sur le calcul du champ de tenseurs de structure couleur. L'approche de synthèse à deux étapes résultante est apte à synthétiser avec succès une variété de textures couleur régulières et irrégulières.

L'algorithme de synthèse de textures 3D à partir d'exemplaires 3D en deux étapes structure/texture s'étend de manière naturelle à partir de l'algorithme de synthèse 2D. Pour aboutir à une synthèse 2D/3D à deux étapes structure/texture, une inférence 2D/3D du champ de tenseurs de structure consiste à diviser le voisinage de sortie tridimensionnel en trois voisinages 2D orthogonaux et à appliquer une synthèse multi-2D/2D. Les difficultés rencontrées à ce point mettent en avant une question qui reste ouverte sur la façon de combiner les trois tenseurs 2D choisis séparément pour chaque vue. Bien qu'une méthode de combinaison de tenseurs satisfaisante n'ait pas été développée dans ce mémoire, différentes solutions possibles selon cette direction ont été discutées et seront à la base de travaux futurs.

Enfin, une variante de l'algorithme de base permettant la génération de textures de formes arbitraires est mise en place. L'aptitude à synthétiser des textures de formes spécifiées, tout en préservant les caractéristiques de la texture d'entrée, en utilisant la couche de structure de la texture imposée, font de cette approche de synthèse structure/texture une démarche pertinente.

Il nous reste à souligner les perspectives de développement de ces travaux :

Pour tous les résultats, présentés aux chapitres 3 et 4, aucune accélération logicielle e.g. l'accélération par quantification vectorielle (*Tree-Structured Vector Quantization - TSVQ*) n'a été utilisée, pour éviter toute dégradation des résultats liée à la sous-optimalité de telles solutions. L'adaptation de ces algorithmes d'accélération au voisinage de tenseurs de structure est cependant digne d'intérêt et pourrait faire l'objet de travaux complémentaires.

En outre, la qualité des résultats obtenus est encourageante, en termes de structure et de dynamique, et prouve que la méthode proposée est avantageuse pour reproduire fidèlement les variations d'orientation de l'exemplaire. Ces résultats prometteurs, obtenus surtout sur les échantillons de matériaux, ouvrent une voie vers la synthèse de textures plus complexes, anisotropes et non-stationnaires, et de textures montrant des motifs multi-échelles de grandes dimensions, comme dans le cas des échantillons de matériaux de carbone à structure laminaire.

En ce qui concerne la problématique de l'inférence 2D/3D du tenseur de structure, l'étude et le développement des solutions possibles (cf. paragraphe 5.3.3) qui rendent compte de la pertinence de la matrice obtenue par la combinaison de trois tenseurs 2D, et de l'exigence de la semi-définie positivité de cette matrice, devra être poursuivie. L'utilisation du champ de tenseurs de structure 3D obtenu comme contrainte à la synthèse de la couche de texture volumique est envisageable. L'extension couleur de l'algorithme de synthèse de textures 3D à partir d'échantillons 2D s'avèrera facile à mettre en œuvre par une simple utilisation du tenseur de structure couleur développé au paragraphe 5.2.1.

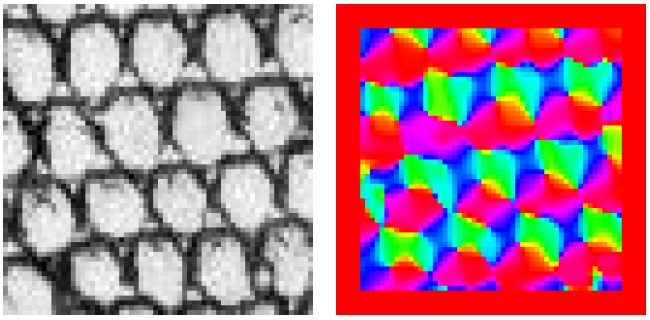
Enfin, il serait intéressant de poursuivre l'approche de synthèse de textures de formes arbitraires et d'étudier son efficacité dans la synthèse de matériaux carbonés. Appliqué aux carbones pyrolytiques, aux graphènes et aux suies, cela permettrait par exemple de synthétiser des structures sphériques concentriques non observables expérimentalement en 3D, à partir de sections 2D d'empilements laminaires (observables en microscopie) et d'une consigne découlant de la connaissance de la structure 3D du matériau. Ainsi une description plus réaliste de la structure du matériau serait obtenue, avec des applications prometteuses pour une caractérisation de ces matériaux sensibles.

Annexes

Annexe A
Traitement spécifique aux bords du champ de tenseurs de structure d'entrée

Le champ de tenseurs de structure S_{in} de l'échantillon d'entrée I_{in} est construit à partir des champs de gradient $\nabla I = [I_x\ I_y]$ estimés à l'aide des noyaux de dérivées Gaussiennes D_x et D_y d'écart-type σ_D (cf. paragraphe 3.2). Puisque D_x et D_y dépassent les frontières sur les bords extérieurs de I_{in}, le calcul du gradient se limite à l'intérieur de I_{in} laissant une zone (couronne) inutilisée dans la procédure de recherche de la meilleure ressemblance (Figure A.1).

Soit Z_C une couronne ayant une largeur T_C. En fait, T_C dépend de la taille des noyaux D_x et D_y, et de la taille du noyau de lissage Gaussien $W = G_\sigma$ d'écart-type σ. Dans le cas de synthèse multi-échelle, on définit T_C^L comme la largeur de la couronne Z_C^L de l'étage L; $L = 0, \dots, NL - 1$, où NL est la profondeur (i.e. le nombre d'étages) de la pyramide Gaussienne. Z_C^L dépend également de la taille de la fenêtre de lissage Gaussienne G_{σ_L} d'écart-type σ_L utilisée pour construire la pyramide Gaussienne.



I_{in} O_{in}

Figure A.1 – *Illustration de la zone inutilisée dans la procédure de recherche de la meilleure ressemblance. A gauche : texture d'entrée. A droite : le champ de tenseurs de structure de la texture d'entrée représenté par son image d'orientation. La couronne rouge représente la zone inutilisée dans la recherche de la meilleure ressemblance.*

Par conséquent, on définit une pyramide de tenseurs de structure utile de largeur W_u^L et de hauteur H_u^L, comme la version de la pyramide de tenseurs classique de largeur W^L et de hauteur H^L, dont nous avons extrait les couronnes Z_C^L à chaque étage L. La recherche de la meilleure ressemblance aux voisinages des champs de tenseurs de sortie, se fait donc dans la pyramide de tenseurs de structure utile.

La largeur de l'étage à plus haute résolution est exprimée par :

$$W_u^0 = W^0 - (D + G) \ , \tag{A.1}$$

où D et G présentent les tailles inutiles dérivées par le calcul du gradient et par le lissage du tenseur, respectivement, et sont donnés par :

$$D = 2 \times [3\sigma_D] \quad , \quad G = 2 \times [3\sigma] \, , \tag{A.2}$$

où [.] désigne l'arrondi en entier par excès.

W_u^1 est donc exprimé par :

$$W_u^1 = \frac{W_u^0 - Liss}{2} = \frac{W^0 - D - G - Liss}{2} \ , \tag{A.3}$$

où $Liss$ présente la taille inutile dérivée par le lissage Gaussien utilisé pour construire les étages supérieures de la pyramide :

$$Liss = 2 \times [3\sigma_L] \ , \tag{A.4}$$

Par conséquent, la largeur du dernier étage (i.e. à plus basse résolution) est donnée par:

$$W_u^{NL-1} = \frac{W_u^{NL-2} - Liss}{2} = \frac{W^0 - D - G}{2^{NL-1}} - Liss\left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{NL-1}}\right), \tag{A.5}$$

Le terme $\left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{NL-1}}\right)$ étant une série géométrique de raison et de premier terme égales à $\frac{1}{2}$, on peut donc écrire :

$$W_u^{NL-1} = \frac{W^0 - D - G}{2^{NL-1}} - Liss\left(1 - \frac{1}{2^{NL-1}}\right) \tag{A.6}$$

De la même façon, H_u^{NL-1} peut être exprimé par :

$$H_u^{NL-1} = \frac{H^0 - D - G}{2^{NL-1}} - Liss\left(1 - \frac{1}{2^{NL-1}}\right) \ , \tag{A.7}$$

Finalement, les largeurs et les hauteurs de chaque étage de la pyramide de tenseurs de structure utile, sont calculées comme suit :

$$\begin{cases} W_u^L = W_u^{NL-1} \times 2^{NL-1-L} \\ H_u^L = H_u^{NL-1} \times 2^{NL-1-L} \end{cases} \quad ; \ L = 0, \dots, NL - 1 \tag{A.8}$$

Ayant une texture de taille 256×256 par exemple, avec $\sigma_D = 1$, $\sigma = 1.5$, $\sigma_L = 1.6$ et $NL = 3$, on aura : $D = 6$, $G = 10$ et $Liss = 10$ et par conséquent $W_u^0 = H_u^0 = 208$, $W_u^1 = H_u^1 = 104$ et $W_u^2 = H_u^2 = 52$.

Annexe B
Traitement spécifique aux bords du champ de tenseurs de structure de sortie

Puisque le voisinage de synthèse dépasse les bords extérieurs du champ de tenseurs S_{out} en cours de synthèse, ce dernier est considéré comme une donnée toroïdale et sera donc géré périodiquement i.e. si le même champ est répété et copié sur les deux directions, horizontale et verticale, le champ obtenu est continu et les motifs sont répétés sans interruptions [Urs13].

En d'autres termes, durant le processus de synthèse, les voisinages atteignant les extrémités doivent inclure leur côté opposé pour obtenir une représentation continue. Il s'agit donc de créer un nouveau champ de tenseurs S_{out_tor} à partir du champ de tenseurs S_{out} initialisé par un bruit de tenseurs. S_{out_tor} a donc une plus grande taille que S_{out} et il est construit tel que les voisinages utilisés pour synthétiser les tenseurs du bord supérieur de S_{out} doivent contenir les tenseurs du bord inférieur de S_{out}, les voisinages des bords à droite doivent contenir les tenseurs du bord gauche et ainsi de suite. Le champ de tenseurs de structure toroïdal S_{out_tor} est donc construit à partir du champ de tenseurs de sortie S_{out} par copie des régions aux bords comme illustré à la Figure B.1. Cela assure que le tenseur ayant les coordonnées (x, y) est une copie du tenseur de coordonnées (x *mod* H, y *mod* W) où H et W représentent la hauteur et la largeur de S_{out}, respectivement [Wei00]. Après chaque itération de synthèse, S_{out_tor} est mis à jour à partir du nouveau S_{out} obtenu.

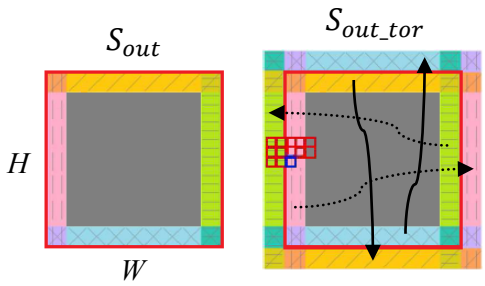


Figure B.1 – *Illustration du processus de construction du champ de tenseurs de structure périodique* S_{out_tor} *à partir du champ de tenseurs de sortie* S_{out} [Urs13].

Ainsi, dans le cas de synthèse multi-échelle, le processus de la Figure B.1 est appliqué pour chaque étage PS_{out}^{L} ($L = 0, \dots, NL - 1$, où NL est la profondeur de la pyramide) de la pyramide de sortie PS_{out} pour construire le champ de tenseurs périodique $PS_{out_tor}^{L}$ à l'étage L de la pyramide toroïdale PS_{out_tor}.

La Figure B.2 illustre un exemple de périodicité d'un champ de tenseurs de structure synthétique obtenu par le processus de synthèse proposé. La première colonne montre un exemple de calepinage de champ de tenseurs périodique (i.e. répétition et copie sur les deux directions, horizontale et verticale). Les deuxième et troisième colonnes montrent l'orientation et la cohérence du champ de tenseurs périodique (en haut) et leur assemblage selon le calepinage proposé (en bas), respectivement. On peut noter que les champs d'orientation et de cohérence obtenus sont continus et leurs motifs sont répétés sans interruptions. Ces configurations sont très efficaces en termes de coût calculatoire. En d'autres termes, cette méthode est bénéfique en synthétisant des champs de tenseurs de structure de petite taille puis construire des plus grands champs de tenseurs par l'assemblage de différents échantillons des champs déjà synthétisés.

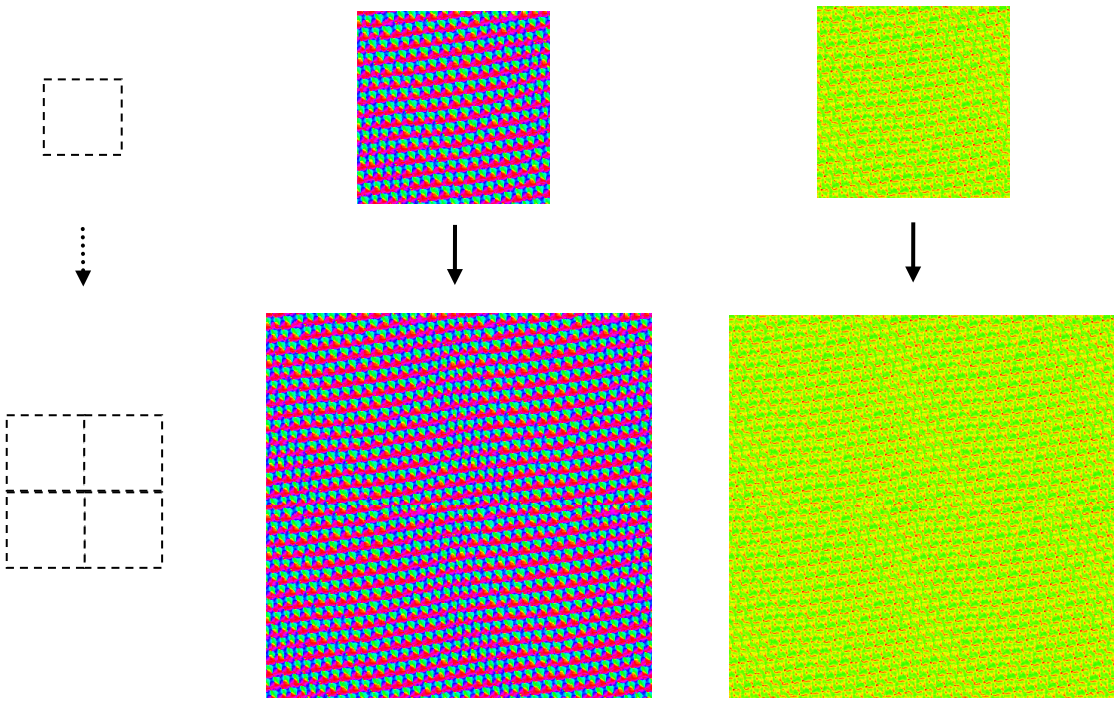


Figure B.2 – *Exemple de périodicité d'un champ de tenseurs synthétique obtenu par le processus de synthèse proposé. 1ère colonne : exemple de calepinage d'un champ de tenseurs périodique, 2ème colonne : l'orientation du champ de tenseurs périodique (en haut) et son résultat de calepinage (en bas). 3ème colonne : la cohérence du champ de tenseurs périodique (en haut) et son résultat de calepinage (en bas).*

Annexe C
Comparaison des métriques de dissimilarité de tenseurs

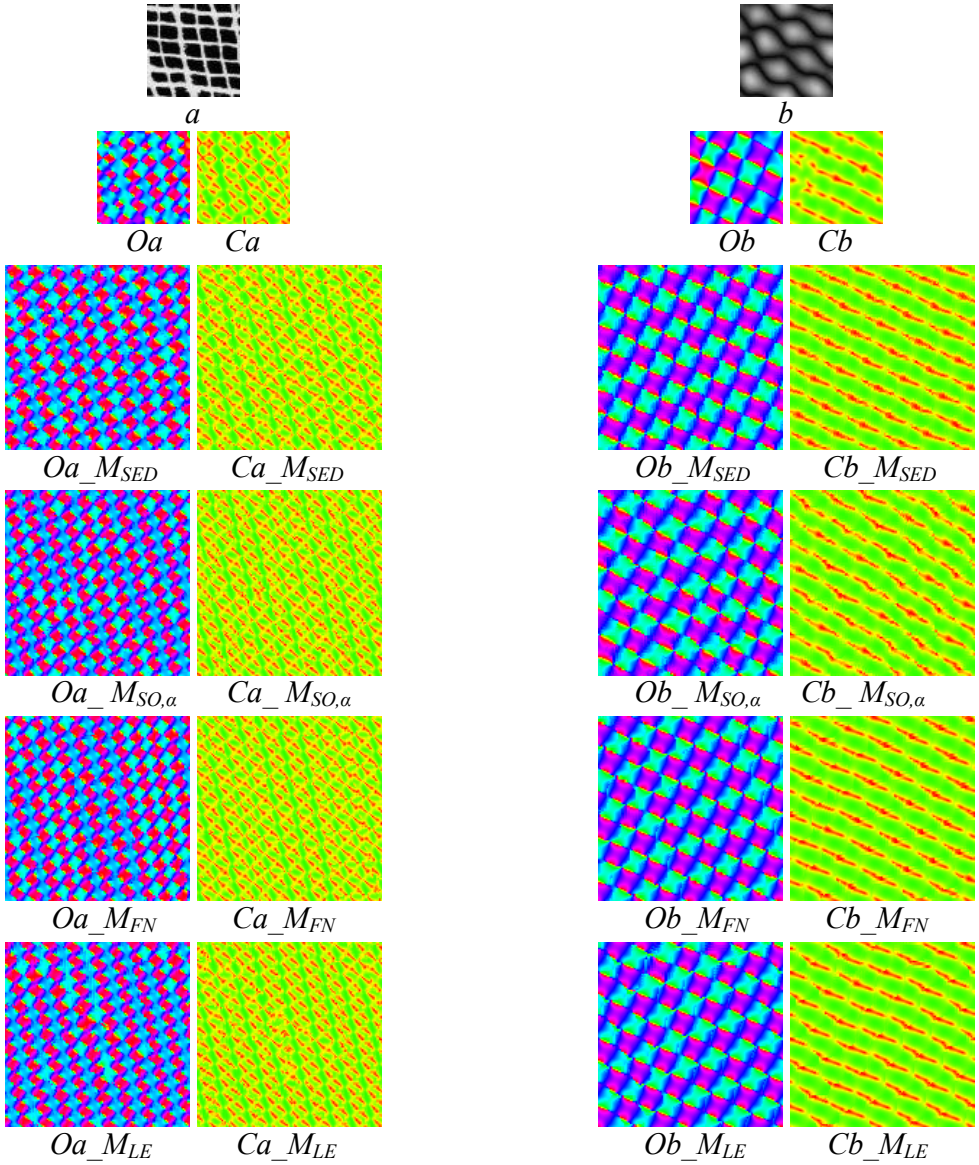


Figure C.1 – *Résultats de synthèse du champ de tenseurs de structure avec différentes métriques de l'espace tensoriel (cf. paragraphe 3.4.1.1). Chaque résultat montre, de haut en bas : l'exemplaire, le champ de tenseurs d'entrée représenté par son image d'orientation (à gauche) et de cohérence (à droite), l'orientation (à gauche) et la cohérence (à droite) du champ de tenseurs de structure obtenu en utilisant le carré de la distance Euclidienne, la métrique Shape-Orientation, la norme de Frobenius et la métrique Log-Euclidienne. Les images d'entrée et de sortie sont de taille 64×64 et 128×128, respectivement. Des pyramides Gaussiennes à 2 échelles sont utilisées avec un voisinage causal, un parcours lexicographique et 3 itérations.*

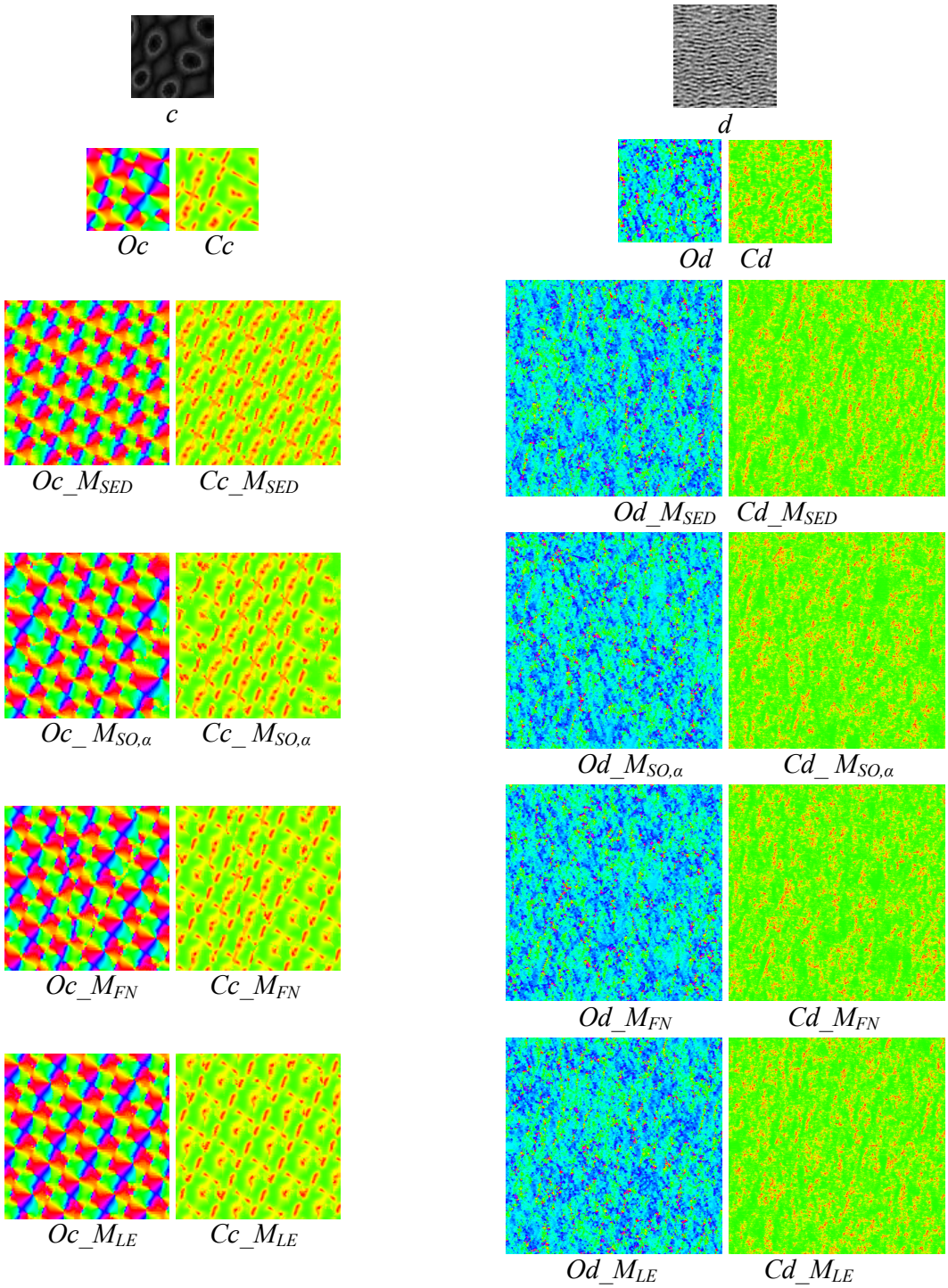


Figure C.2 – *Résultats de synthèse du champ de tenseurs de structure avec différentes métriques de l'espace tensoriel (cf. paragraphe 3.4.1.1). Chaque résultat montre, de haut en bas : l'exemplaire, le champ de tenseurs d'entrée représenté par son image d'orientation (à gauche) et de cohérence (à droite), l'orientation (à gauche) et la cohérence (à droite) du champ de tenseurs de structure obtenu en utilisant le carré de la distance Euclidienne, la métrique Shape-Orientation, la norme de Frobenius et la métrique Log-Euclidienne. Des pyramides Gaussiennes à 2 échelles sont utilisées avec un voisinage causal, un parcours lexicographique et 3 itérations. Résultat c :images d'entrée et de sortie de taille 64×64 et 128×128, respectivement. Résultat d :images d'entrée et de sortie de taille 256×256 et 512×512, respectivement.*

Annexe D
Exemples de périodicité de textures synthétiques

La Figure D.1 illustre deux exemples de périodicité de textures synthétiques obtenues par le processus de synthèse proposé. La première colonne montre un exemple de calepinage de textures périodiques (i.e. répétition et copiage sur les deux directions, horizontale et verticale). Les deuxième et troisième colonnes montrent les textures périodiques (en haut) et leurs assemblages (en bas). On peut remarquer que les résultats d'assemblage sont continus, les motifs étant répétés sans interruptions.

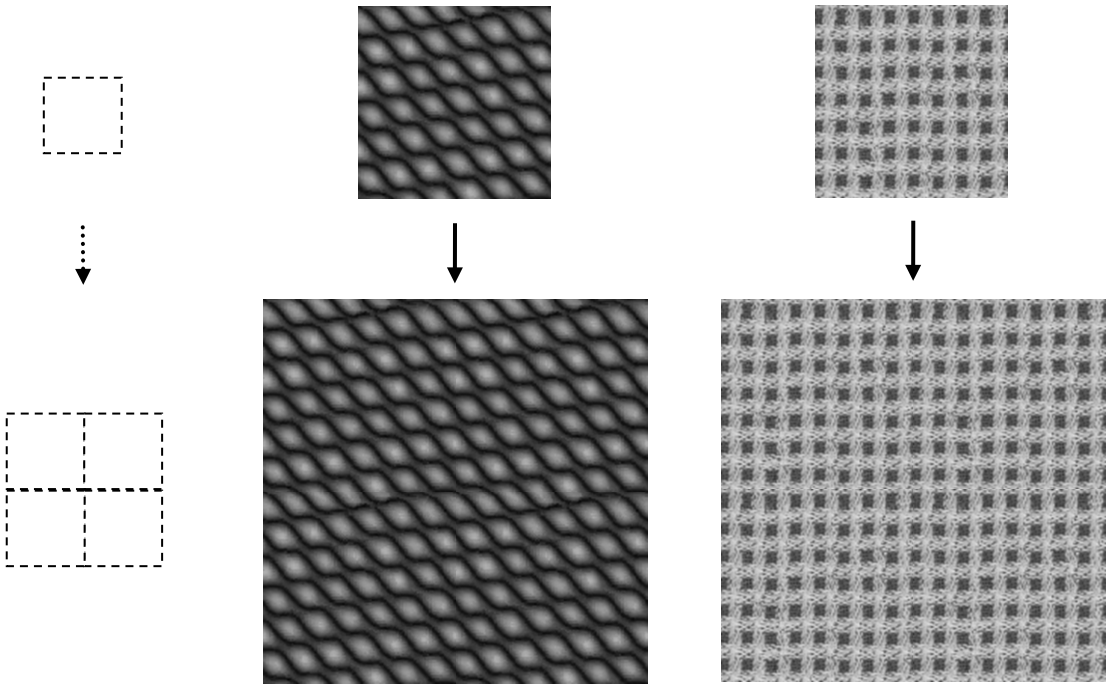


Figure D.1 – *Exemple de périodicité de textures synthétiques obtenues par le processus de synthèse proposé. 1ère colonne : exemple de calepinage de textures périodiques, 2ème et 3ème colonnes : les textures synthétiques périodiques (en haut) et leurs assemblages (en bas).*

Annexe E
Efficacité de la couche de structure et influence de la pondération structure/texture

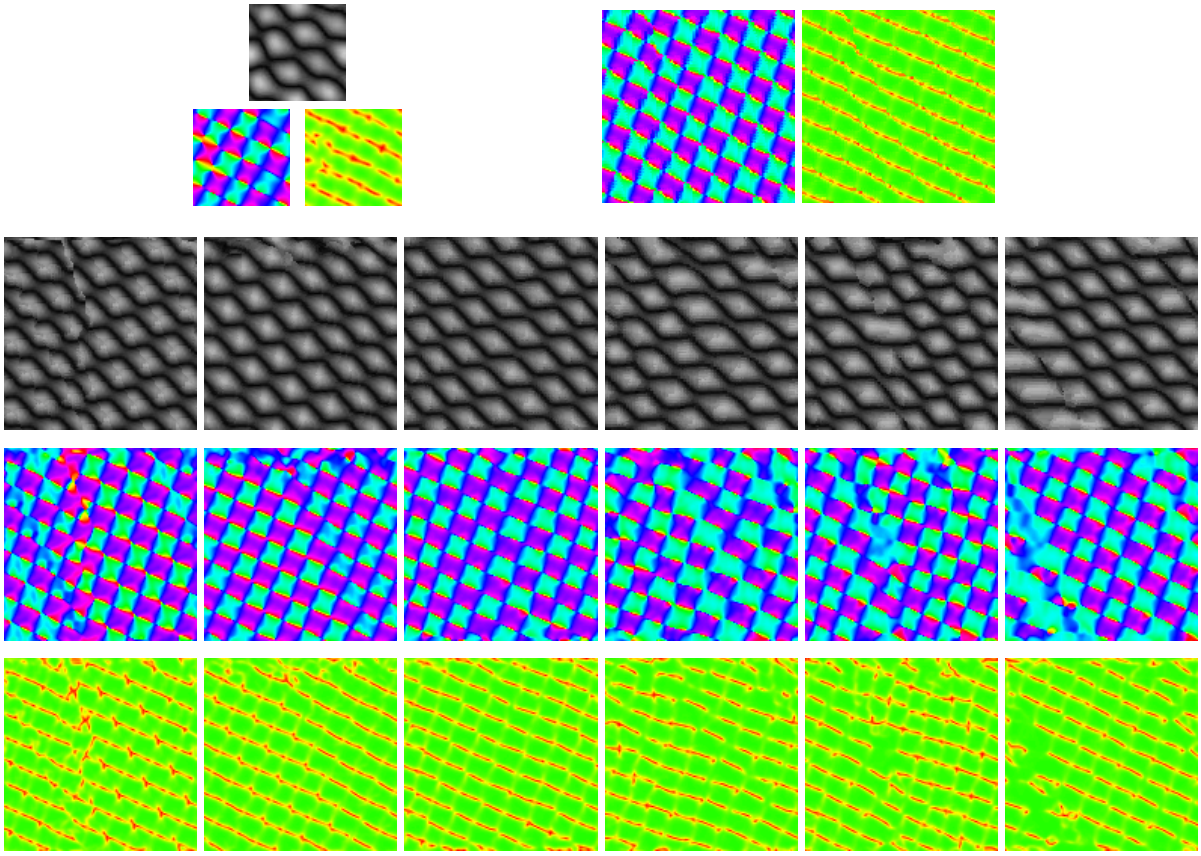


Figure E.1 – *Résultats de synthèse avec différentes valeurs du poids de pondération* p *dans l'équation (16). 1ère ligne (de gauche à droite) : la texture d'entrée avec l'orientation et la cohérence du champ de tenseurs de structure d'entrée, l'orientation et la cohérence du champ de tenseurs de structure synthétisé. Les 2ème, 3ème et 4ème lignes montrent respectivement la texture synthétisée par l'algorihtme proposé et son orientation et cohérence avec (de gauche à droite)* $p = 0$, $p = 0.2$, $p = 0.35$, $p = 0.5$, $p = 0.75$ *et* $p = 1$. *Des pyramides Gaussiennes à 2 étages, des voisinages causaux de taille 15×15 pour la synthèse du tenseur de structure et 13×13 pour la synthèse de la texture, un parcours lexicographique et la norme de Frobenius sont utilisés. 3 et 2 itérations sont respectivement effectuées pour la synthèse de la structure et de la texture.*

Annexe F
Formulation utilisée pour l'autocorrélation d'orientations locales

Notons $O_u \in [0\ \pi[$ l'orientation locale au site $u = (x, y)$ d'une image I et $C_u \in [0\ \ 1]$ la cohérence associée. L'orientation étant une donnée circulaire modulo π, il est judicieux, pour certains calculs, d'utiliser l'angle double [Bat81] [Mar00] qui sera noté $\theta_u = 2O_u \in [0\ \ 2\pi[$.

Les champs d'orientation et de cohérence peuvent être considérés conjointement comme la réalisation d'un processus aléatoire complexe $Z_u = C_u e^{j\theta_u}$ (avec $j^2 = -1$) ou d'un champ aléatoire vectoriel $V_u = (C_u \cos(\theta_u), C_u \sin(\theta_u))^t$ où t est l'opérateur transposée d'une matrice. Dans les deux cas, il est possible d'étendre les concepts de statistiques spatiales, et en particulier celui de l'autocorrélation.

En utilisant une formulation complexe, l'autocorrélation pour un déplacement $\boldsymbol{w}$ s'exprime :

$$\emptyset_z(w) = \langle Z_u \overline{Z_{u+w}} \rangle = Re_z(w) + j\, Im_z(w) \tag{F.1}$$

où $\langle\ \rangle$ désigne la moyenne, $Re_z(w)$ et $Im_z(w)$ sont les parties réelles et imaginaires de $\emptyset_z(w)$ et s'écrivent respectivement :

$$\begin{cases} Re_z(w) = \langle C_u C_{u+w}\, cos(\theta_u - \theta_{u+w}) \rangle \\ Im_z(w) = \langle C_u C_{u+w}\, sin(\theta_u - \theta_{u+w}) \rangle \end{cases} \tag{F.2}$$

La formulation vectorielle aboutit à une autocorrélation sous la forme d'une matrice carrée :

$$\emptyset_V(w) = \langle V_u\, V_{u+w}^t \rangle = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{F.3}$$

où :

$$\begin{cases} M_{11} = \langle C_u C_{u+w}\, cos(\theta_u) cos(\theta_{u+w}) \rangle \\ M_{12} = \langle C_u C_{u+w}\, cos(\theta_u) sin(\theta_{u+w}) \rangle \\ M_{21} = \langle C_u C_{u+w}\, sin(\theta_u) cos(\theta_{u+w}) \rangle \\ M_{22} = \langle C_u C_{u+w}\, sin(\theta_u) sin(\theta_{u+w}) \rangle \end{cases} \tag{F.4}$$

Il est évident que cette matrice est liée à la formulation complexe de l'autocorrélation puisque :

$$trace\big(\emptyset_V(w)\big) = Re_z(w) \tag{F.5}$$

Annexe G
Influence de la taille des voisinages de synthèse de structure et de texture

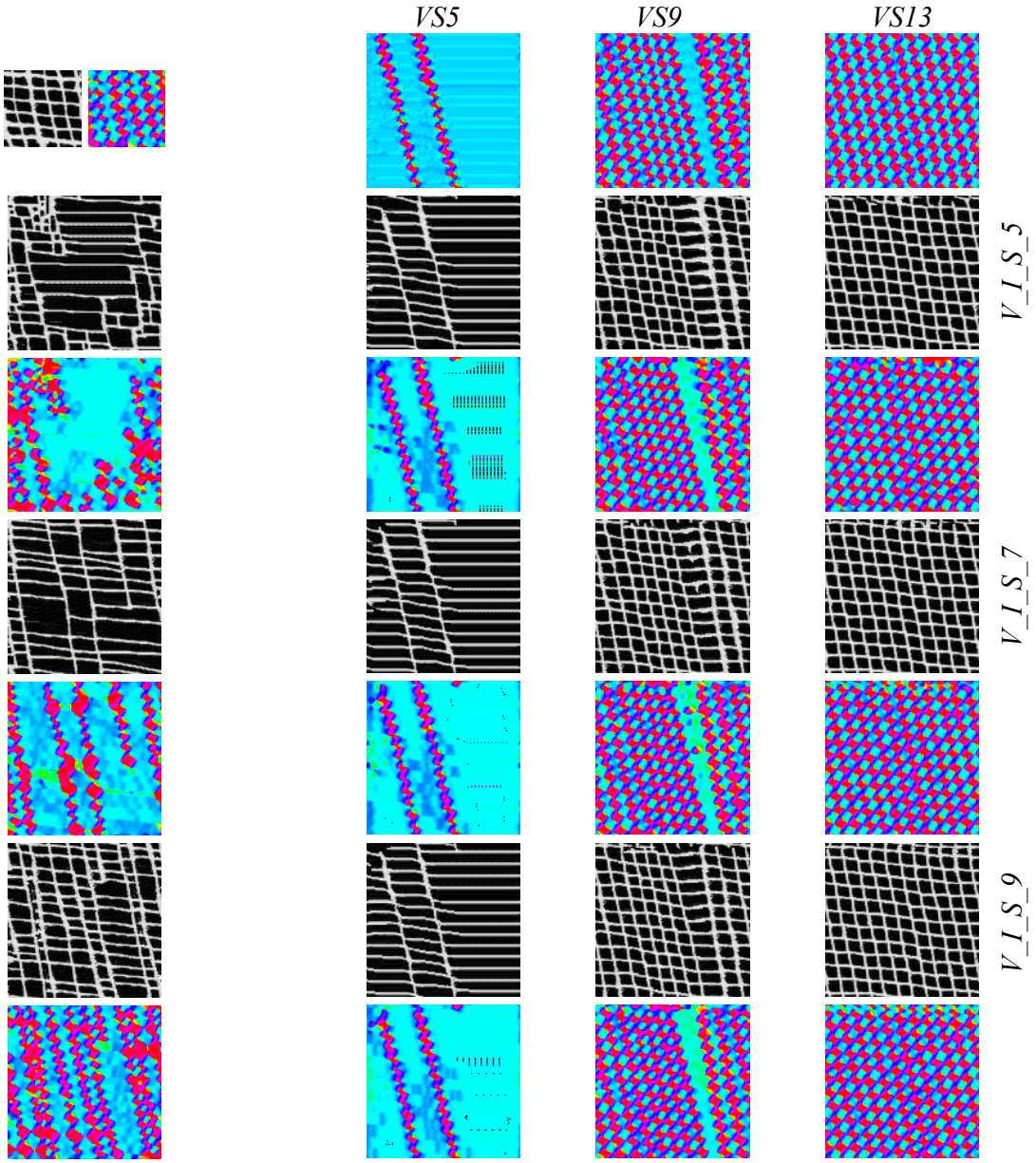


Figure G.1 – *Influence de la taille du voisinage sur la synthèse des couches de structure et de texture. La 1ère ligne (gauche à droite) : l'exemplaire, son image d'orientation, les images d'orientation des champs de tenseurs synthétisés avec des voisinages de structure de taille 5×5, 9×9 et 13×13. Les textures synthétisées en utilisant les contraintes de la 1ère ligne sont montrées à la 2ème ligne avec un voisinage de texture de taille 5×5, à la 4ème ligne avec un voisinage de taille 7×7 et à la 6ème ligne avec un voisinage de texture 9×9, tandis que la 1ère colonne montre les résultats obtenus par l'algorithme de W&L. L'image d'orientation de chaque texture synthétisée est montrée juste au-dessous.*

Annexe H
Algorithme proposé vs. méthodes existantes -1-

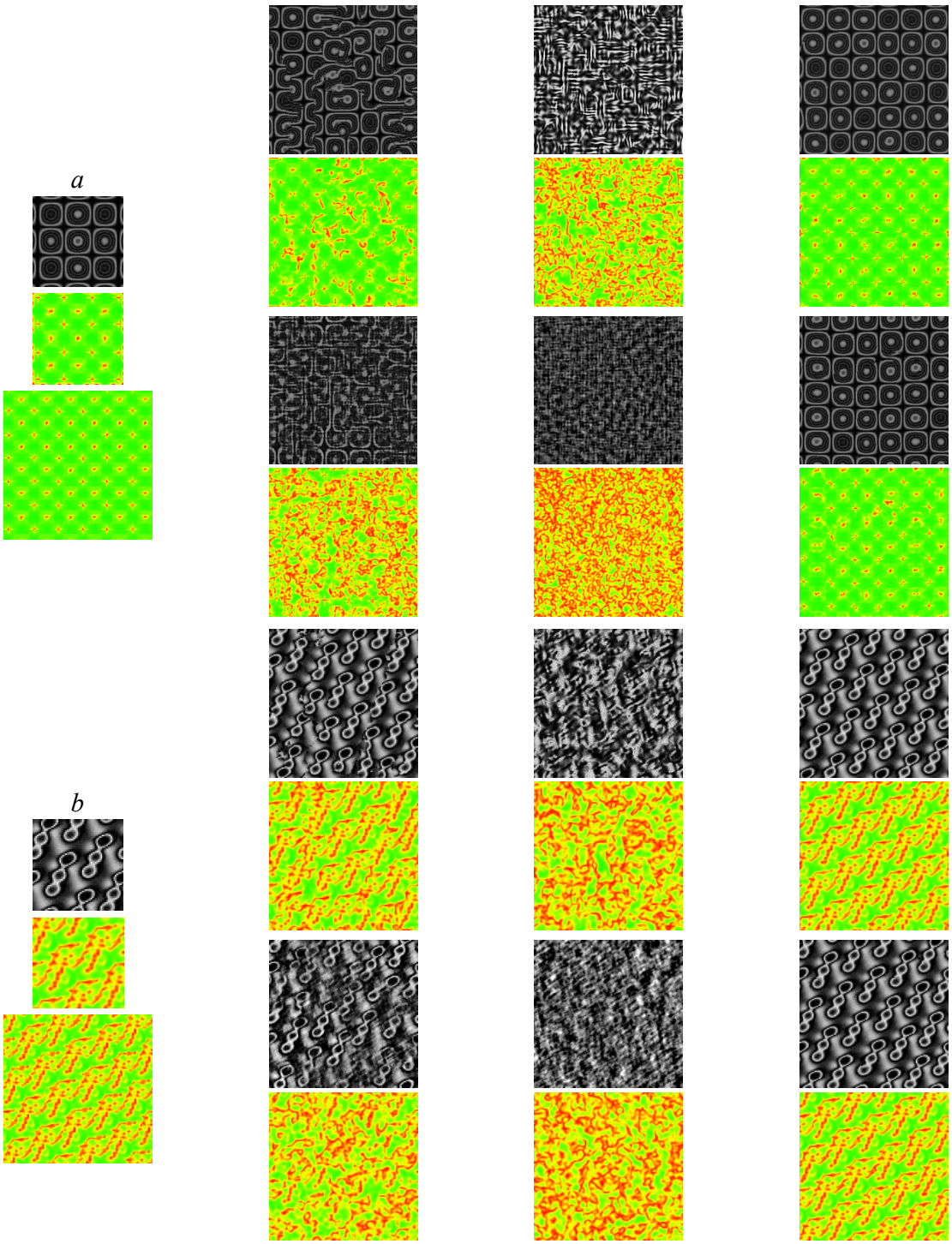


Figure H.1 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, la cohérence du champ de tenseurs d'entrée et la cohérence du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec la cohérence de chaque texture qui est représentée juste au-dessous.*

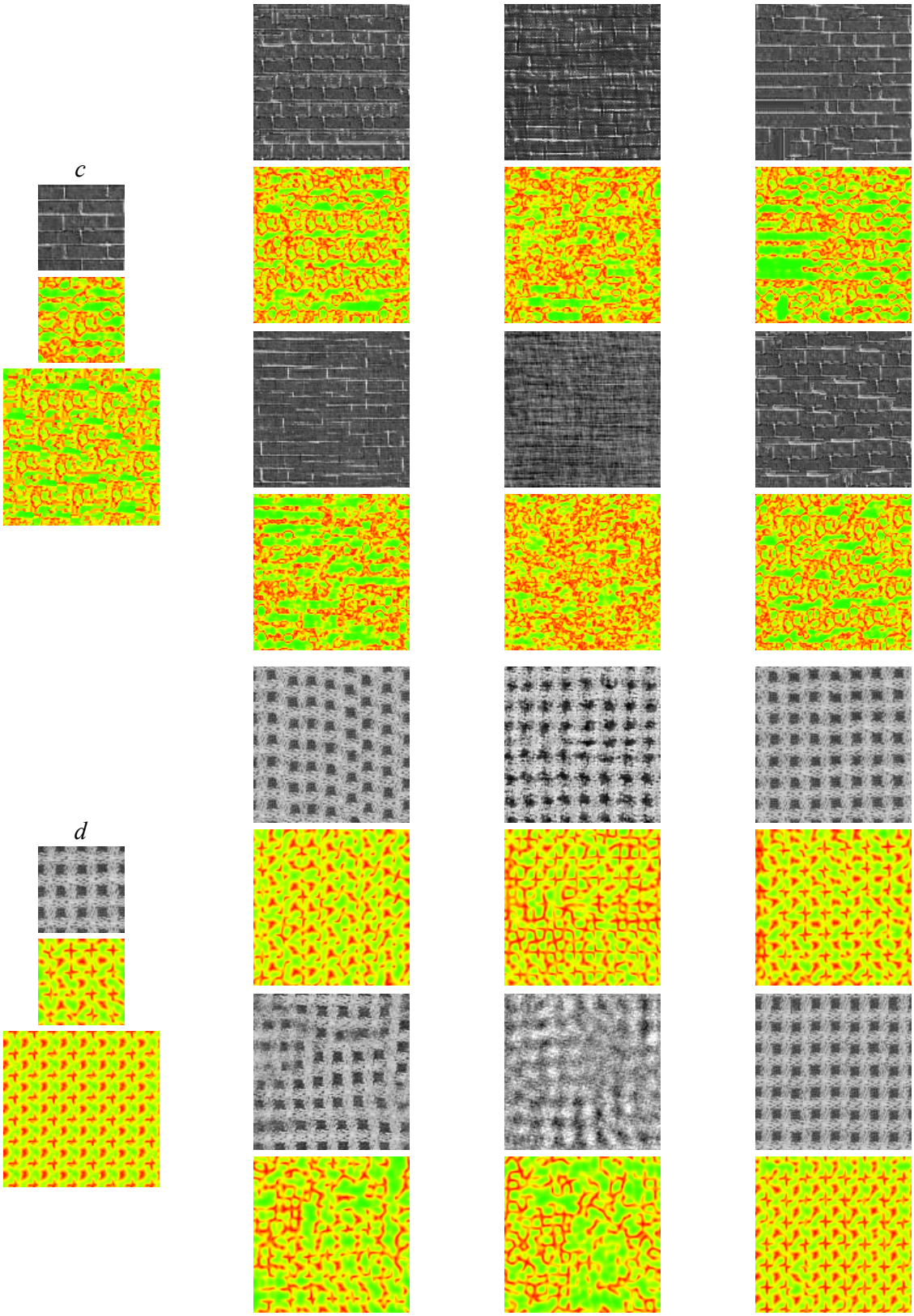


Figure H.2 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, la cohérence du champ de tenseurs d'entrée et la cohérence du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec la cohérence de chaque texture qui est représentée juste au-dessous.*

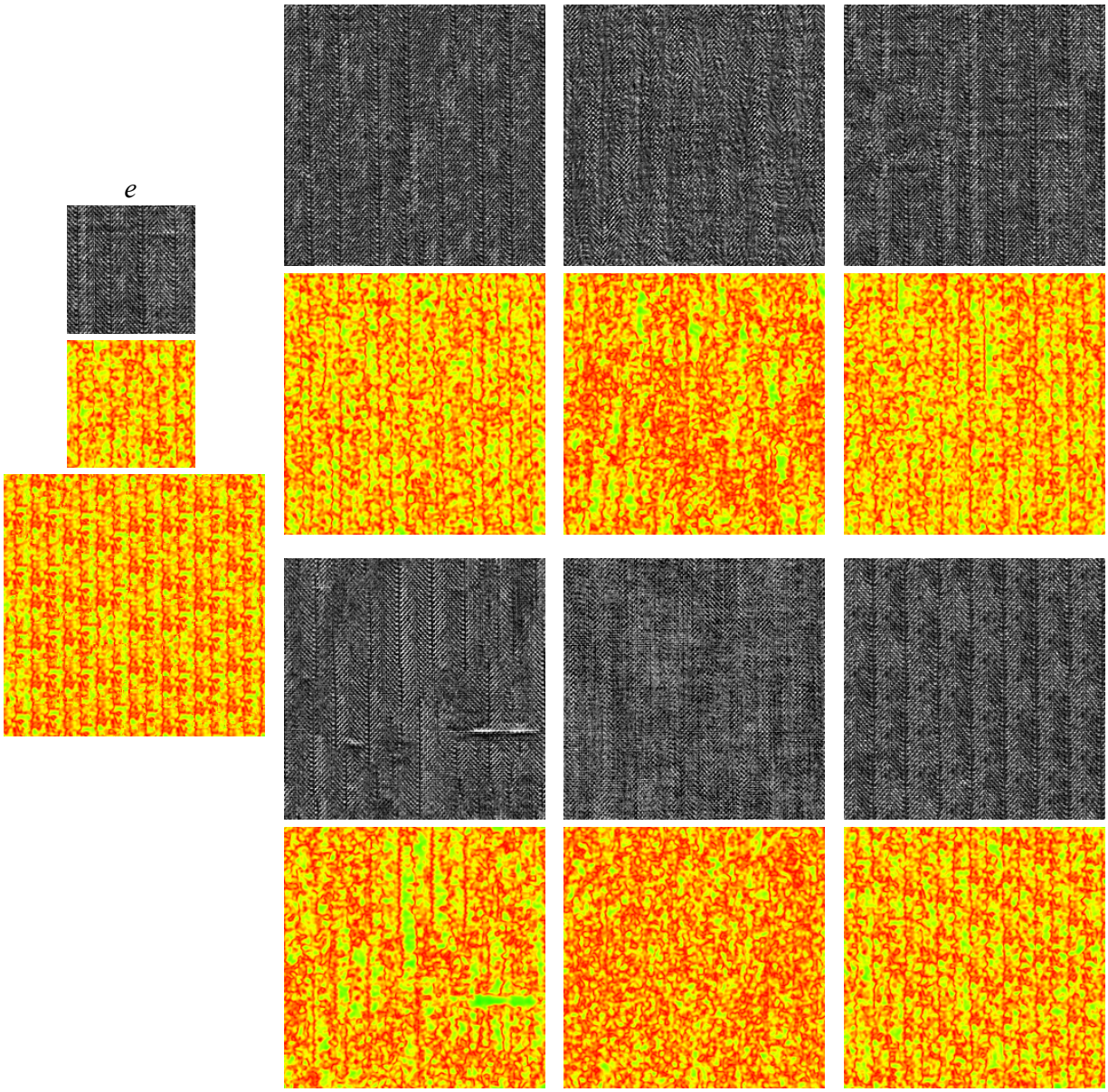


Figure H.3 – *Résultats de synthèse sur une texture régulière structurée. La 1ère colonne montre (de haut en bas) l'exemplaire, la cohérence du champ de tenseurs d'entrée et la cohérence du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec la cohérence de chaque texture qui est représentée juste au-dessous.*

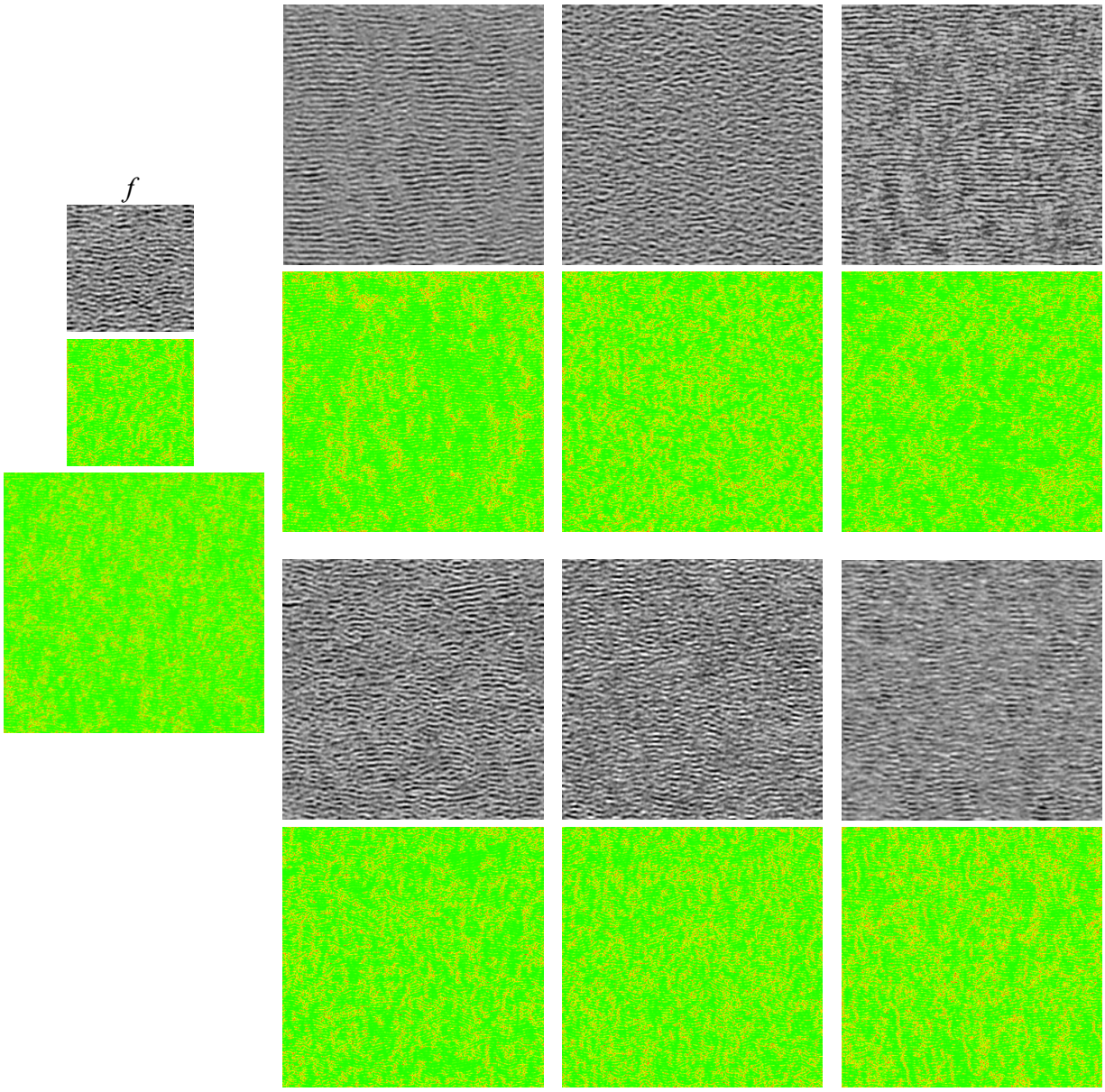


Figure H.4 – *Résultats de synthèse sur une texture irrégulière obtenue par microscopie électronique en transmission à haute résolution. La 1ère colonne montre (de haut en bas) l'exemplaire, l'image de cohérence du champ de tenseurs de structure d'entrée et la cohérence du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec la cohérence de chaque texture qui est représentée juste au-dessous.*

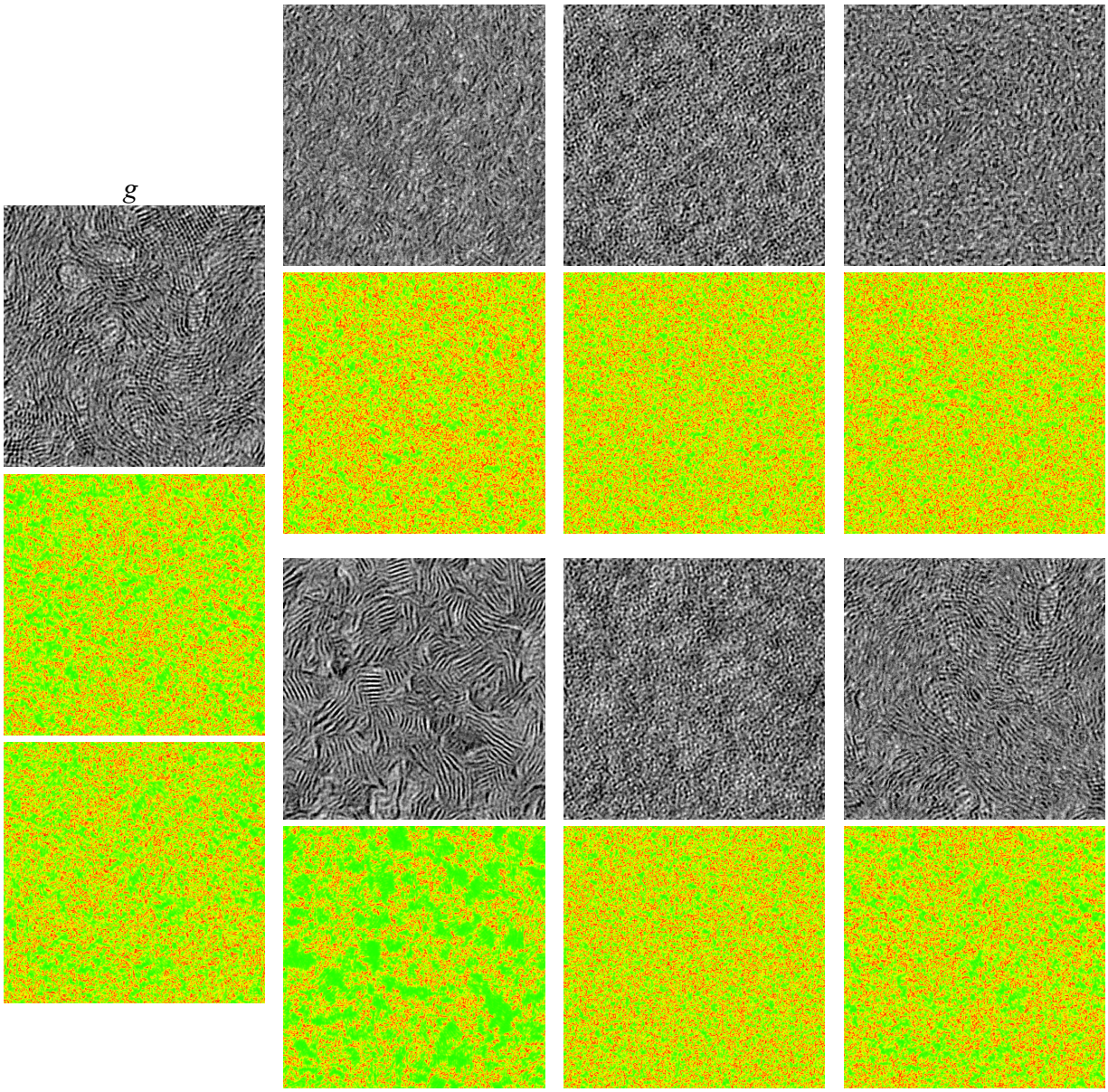


Figure H.5 – *Résultats de synthèse sur une texture irrégulière obtenue par microscopie électronique en transmission à haute résolution. La 1ère colonne montre (de haut en bas) l'exemplaire, l'image de cohérence du champ de tenseurs de structure d'entrée et la cohérence du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec la cohérence de chaque texture qui est représentée juste au-dessous.*

Annexe I
Algorithme proposé vs. méthodes existantes -2-

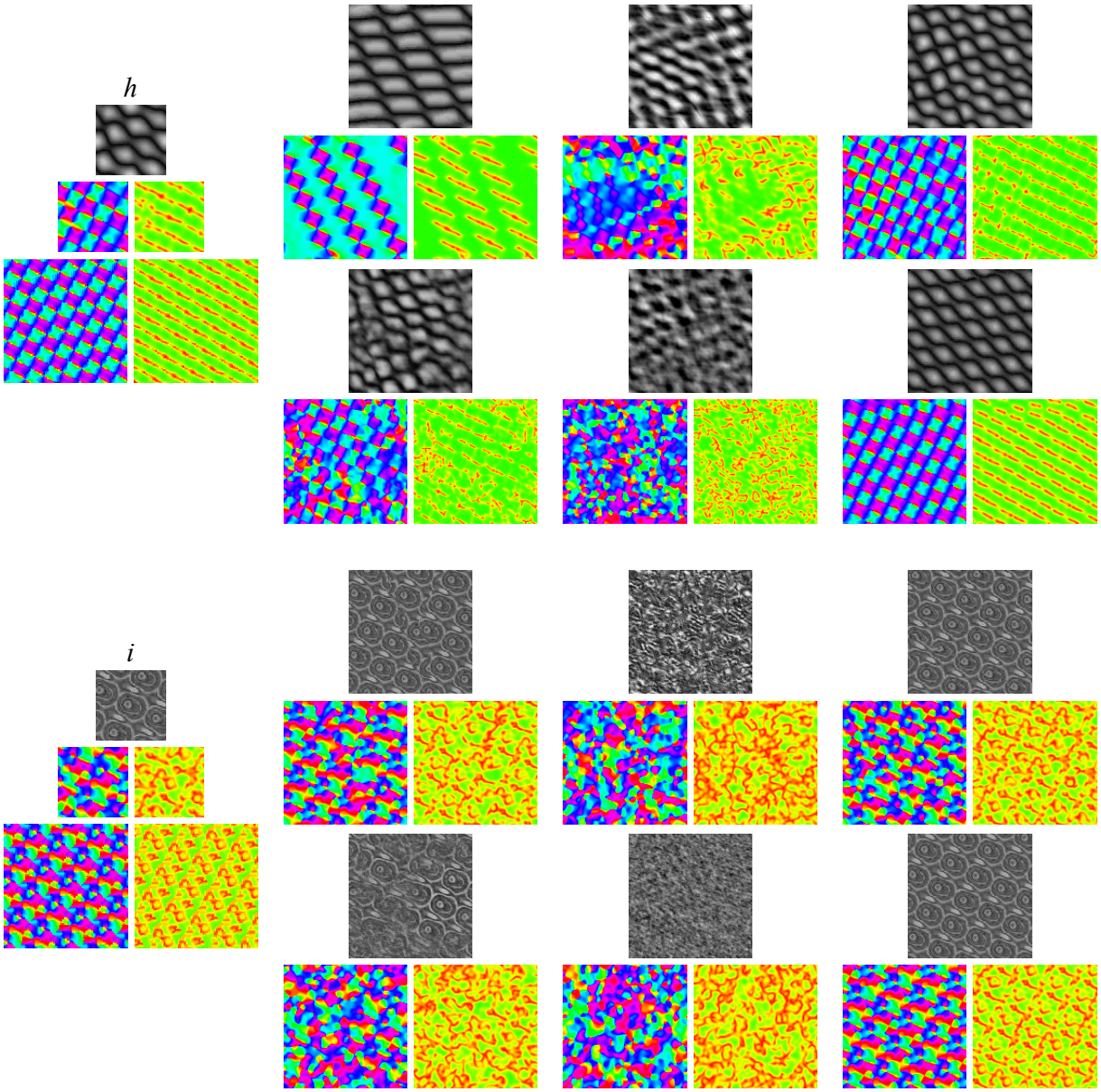


Figure I.1 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, les images d'orientation et de cohérence du champ de tenseurs de structure d'entrée et du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation et la cohérence de chaque texture qui sont représentées juste au-dessous. Les images d'orientation sont montrées à gauche et les images de cohérences sont à droite.*

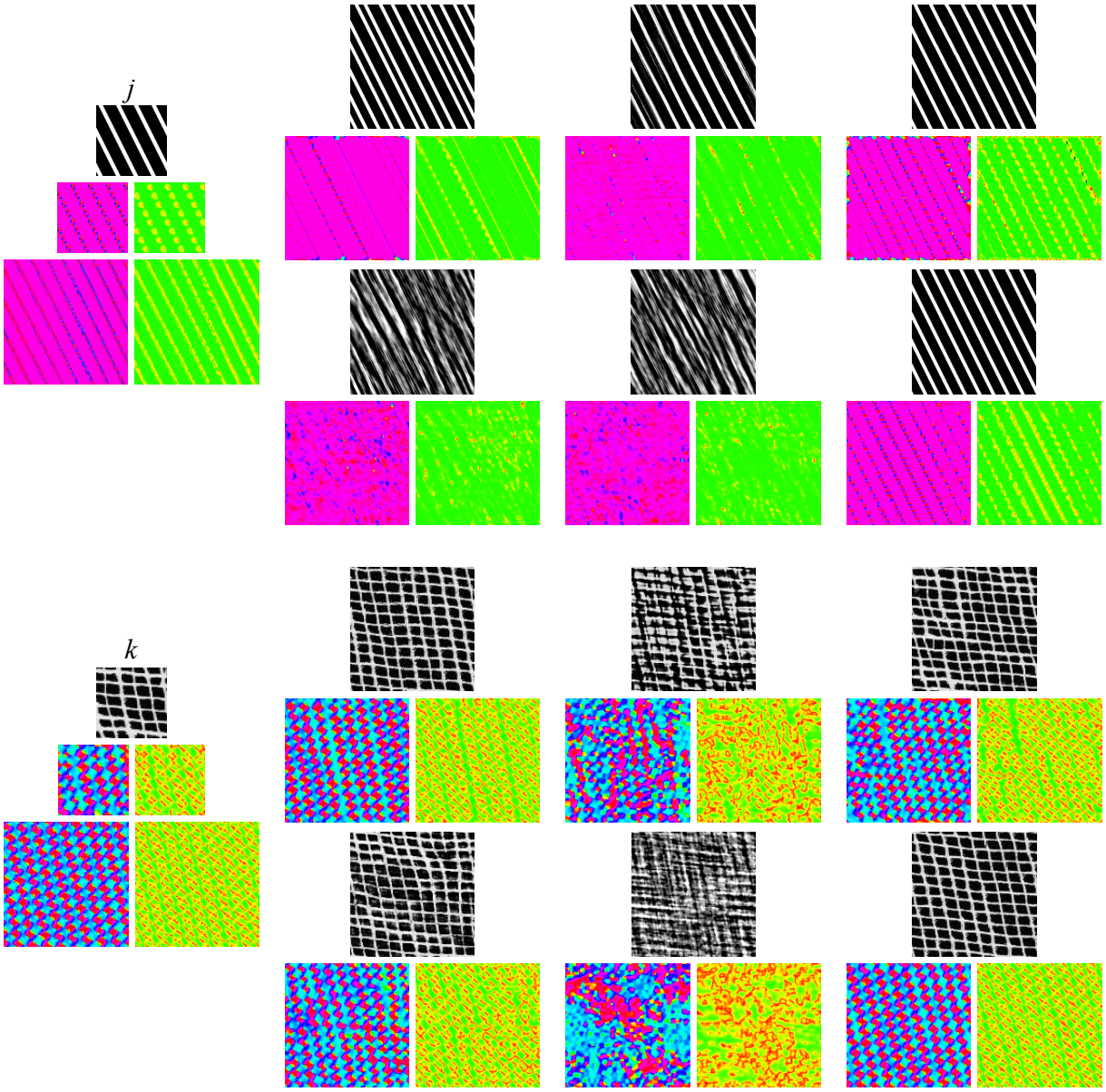


Figure I.2 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, les images d'orientation et de cohérence du champ de tenseurs de structure d'entrée et du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation et la cohérence de chaque texture qui sont représentées juste au-dessous. Les images d'orientation sont montrées à gauche et les images de cohérences sont à droite.*

Annexe J
Algorithme proposé vs. méthodes existantes -3-

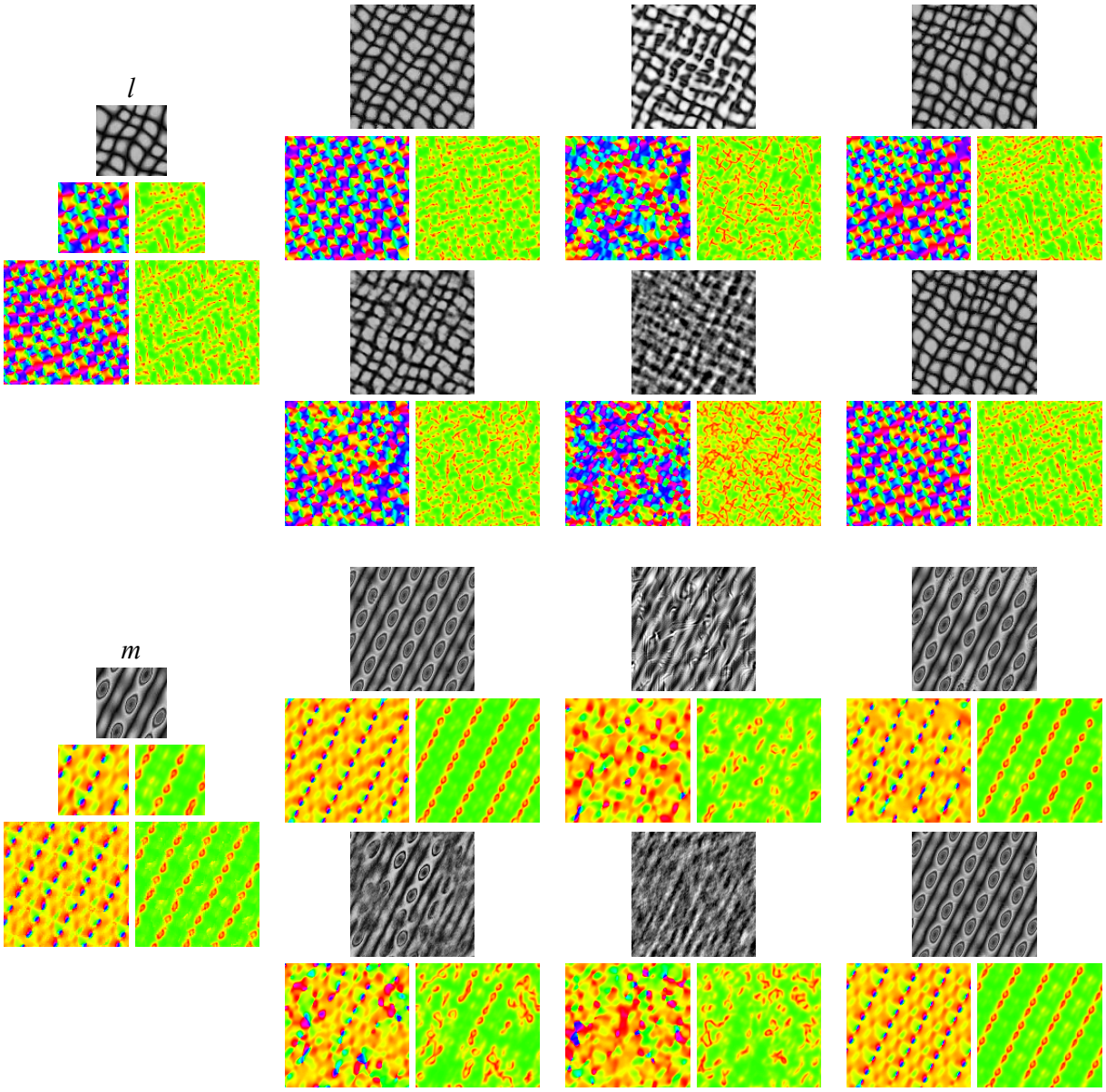


Figure J.1 – *Résultats de synthèse sur des textures régulières structurées. Pour chaque résultat, la 1ère colonne montre (de haut en bas) l'exemplaire, les images d'orientation et de cohérence du champ de tenseurs de structure d'entrée et du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation et la cohérence de chaque texture qui sont représentées juste au-dessous. Les images d'orientation sont montrées à gauche et les images de cohérences sont à droite.*

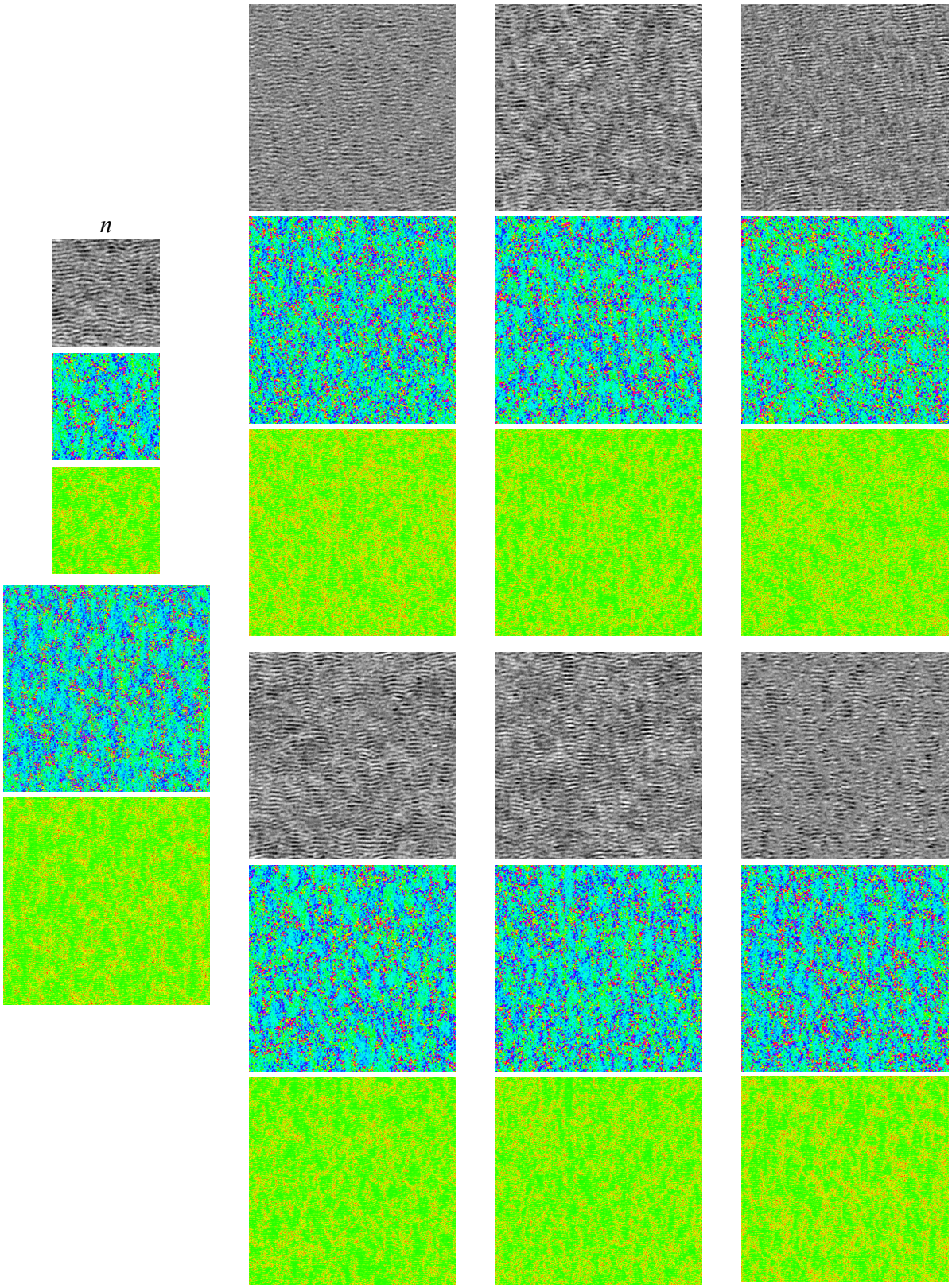


Figure J.2 – *Résultats de synthèse sur une texture irrégulière obtenue par microscopie électronique en transmission à haute résolution. La 1ère colonne montre (de haut en bas) l'exemplaire, les images d'orientation et de cohérence du champ de tenseurs de structure d'entrée et du champ de tenseurs synthétisé. Les 2ème à 4ème colonnes montrent (d'en haut à gauche jusqu'en bas à droite) les textures obtenues avec W&L, P&S, E-L, Tar, Gal et l'approche de synthèse structure/texture proposée (S/T) avec l'orientation et la cohérence de chaque texture qui sont représentées juste au-dessous. Les images d'orientation sont montrées en haut et les images de cohérences sont en bas.*

Annexe K
Echantillons de pyrocarbone observées par microscopie électronique en transmission à haute résolution

Les Figures K.1, K.2 et K.3 présentent les échantillons de matériaux de pyrocarbone de grande taille (1024×1024 pixels) à partir desquelles on a extrait les exemplaires *f*, *g* et *n* de la Figure 35, respectivement.

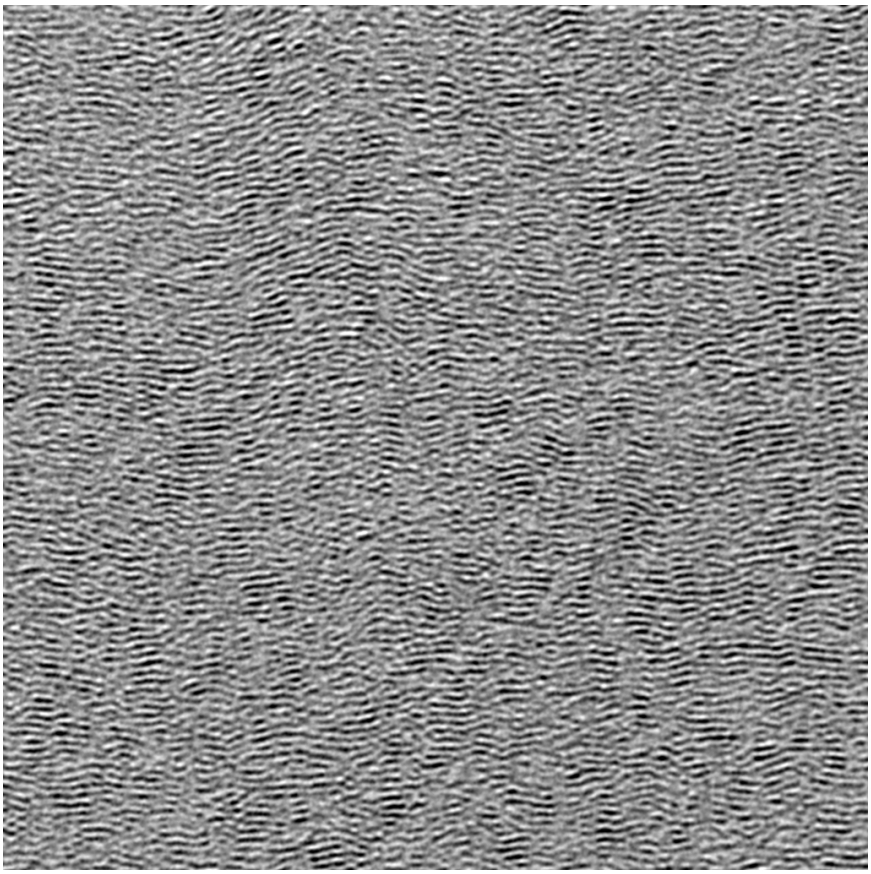


Figure K.1 – *Echantillon de pyrocarbone de la Figure 35.f.*

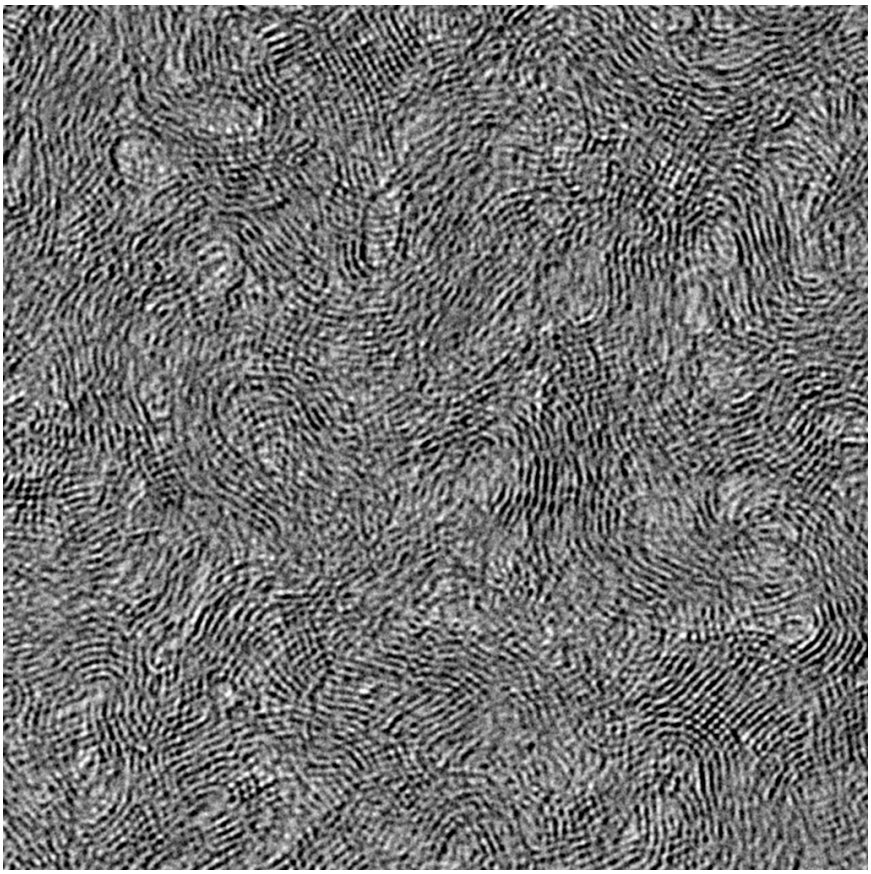


Figure K.2 – *Echantillon de pyrocarbone de la Figure 35.g.*

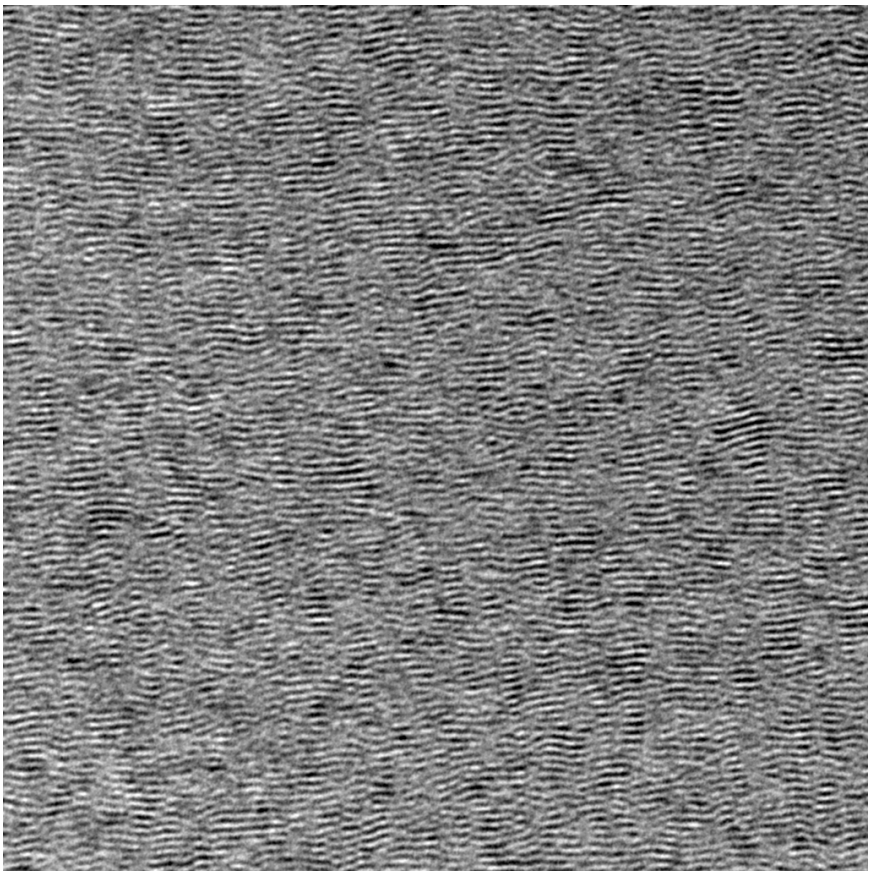


Figure J.K – *Echantillon de pyrocarbone de la Figure 35.n.*

Annexe L
Résultats de synthèse supplémentaires sur des textures couleur

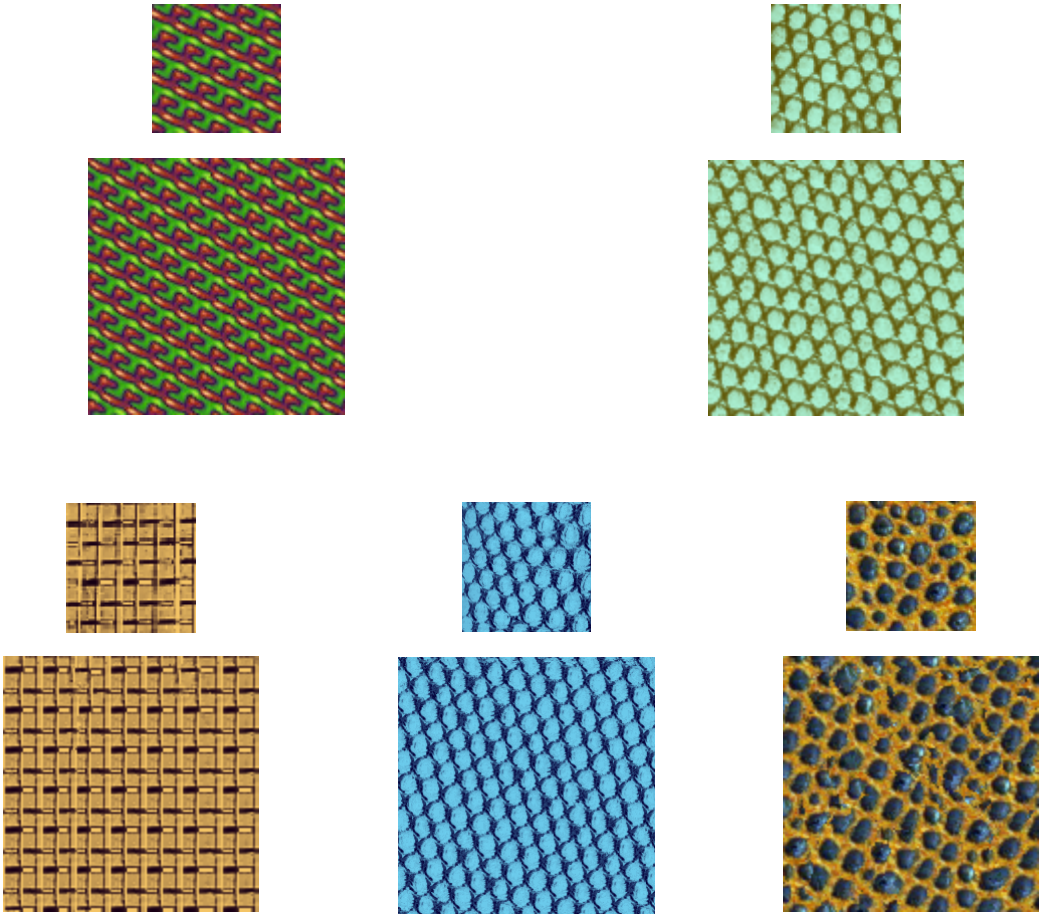


Figure L.1 – *Résultats de synthèse supplémentaires sur des textures couleur en utilisant l'algorithme de synthèse à deux étapes structure/texture proposé. Chaque résultat montre la texture d'entrée (en haut) et la texture synthétisée (en bas).*

Annexe M
Comparaison des approches *A_FS*, *M_FS* et *CT_LS*

La première ligne de la Figure M.1 montre un échantillon de matériau composite anisotrope (I_{in}), son champ de tenseurs de structure (S_{in}) représenté par son image d'orientation et une texture concentrique arbitrairement imposée (I_{arb}), utilisés en entrées des algorithmes *A_FS*, *M_FS* et *CT_LS*. L'image S_{con_1} représente le champ d'orientation de la texture I_{arb}, et I_{out_1} est la texture obtenue par l'approche *A_FS*, i.e. en utilisant le champ de tenseurs S_{con_1}comme contrainte pour la synthèse de la texture. La troisième ligne montre l'image d'orientation du champ de tenseurs instauré par modulation (S_{con_2}) et la texture résultante (I_{out_2}) avec l'approche *M_FS*. La quatrième ligne montre l'orientation du champ de tenseurs instauré par synthèse (S_{con_3}) et la texture résultante (I_{out_3}) avec l'approche *CT_LS*. Tous les résultats sont obtenus à partir du même exemplaire (I_{in}), en utilisant le carré de la distance Euclidienne. Il est important de noter que le champ d'orientation du champ de tenseurs S_{con_1}utilisé pour contraindre la synthèse de I_{out_1}, est à la base de la modulation du champ de tenseurs d'entrée S_{in} pour aboutir à S_{con_2}. En outre, ce même champ d'orientation joue le rôle de la *consigne* pointant vers la partie du dictionnaire concernée dans le processus de synthèse de S_{con_3} et I_{out_3}. Dans tous les résultats, l'approche mono-échelle est utilisée avec un voisinage causal et un parcours lexicographique, et les images synthétiques sont obtenues après deux itérations.

On peut remarquer d'après la Figure M.1 que la synthèse contrainte par le champ de tenseurs de la texture arbitraire conduit à une texture ayant des motifs trop réguliers et qui n'existent pas dans l'exemplaire. En revanche, l'approche *M_FS* réussit à construire, à partir du champ de tenseurs d'entrée, un champ de tenseurs qui représente la structure (les orientations) de la texture arbitrairement imposée, et par conséquent aboutit à une texture restructurée, de forme concentrique, et habillée par les motifs de l'exemplaire. Cela justifie l'importance de l'instauration du champ de tenseurs *contrainte*. Toutefois, du fait que seule l'orientation est modulée (sans tenir compte de l'énergie et du degré d'anisotropie), on peut observer que cette approche fournit des résultats satisfaisants tant que l'incrément angulaire n'est pas trop important. En revanche, la synthèse est moins fidèle quand l'orientation s'écarte largement de celle de l'exemplaire.

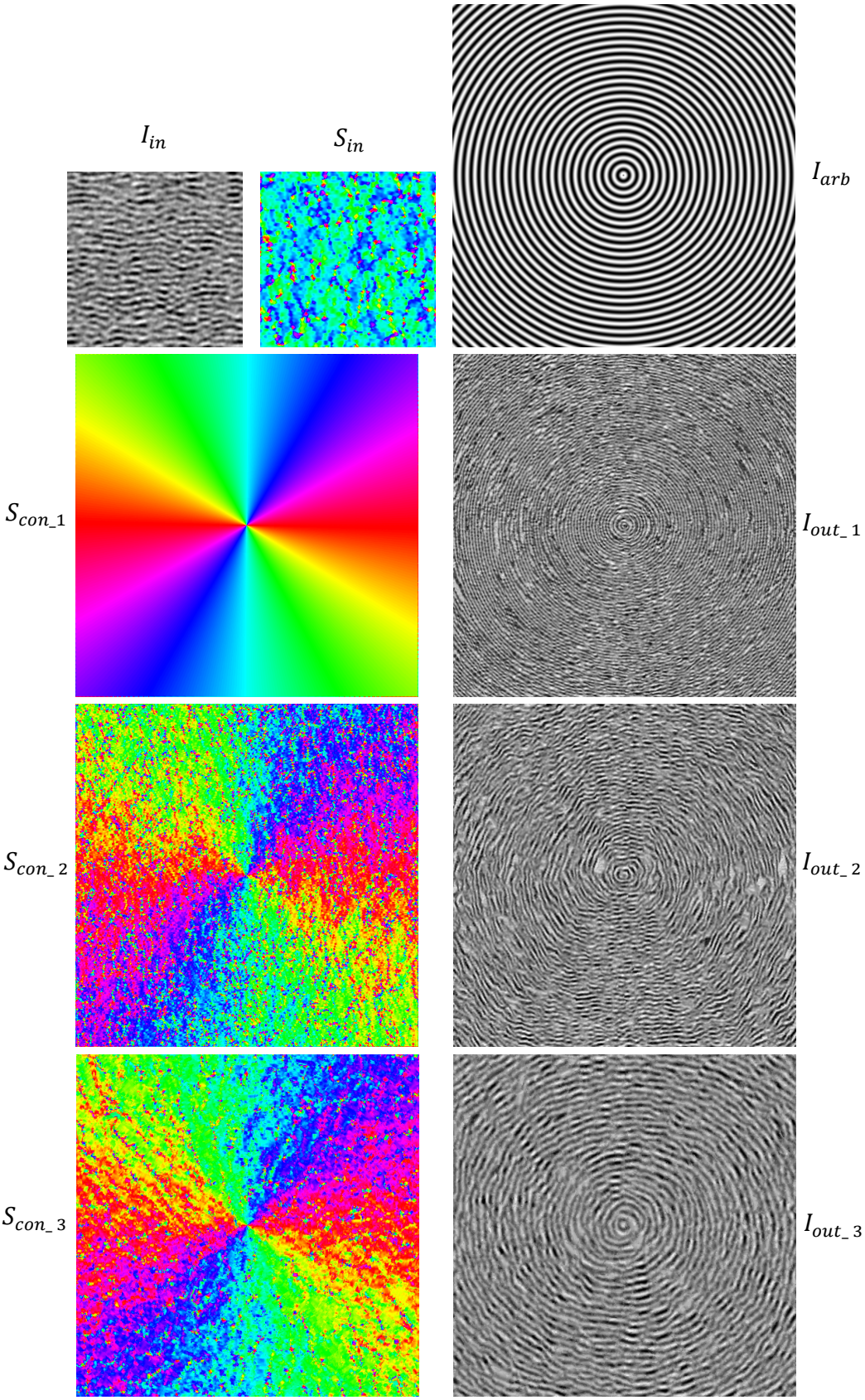


Figure M.1 – *Résultats obtenus avec les approches A_FS, M_FS et CT_LS. 1ère ligne (de gauche à droite) : l'échantillon d'entrée, son champ de tenseurs de structure représenté par son image d'orientation et la texture arbitraire imposée. 2ème ligne : champ d'orientation (S_{con_1}) de la texture arbitraire (I_{arb}), et la texture (I_{out_1}) obtenue par l'approche A_FS. 3ème et 4ème lignes : les orientations des champs de tenseurs générés par M_FS et CT_LS (à gauche) et les textures résultantes (à droite), respectivement.*

Enfin, l'approche *CT_LS* arrive à synthétiser un champ de tenseurs de structure *contrainte* plus lisse, et de bonne qualité en termes de respect des orientations de la texture arbitraire. La texture résultante ayant donc la forme de la texture arbitraire, respecte mieux les dynamiques, les caractéristiques locales et les aspects visuels de l'exemplaire.

Bibliographie

[Agu13] C. Aguerrebere, Y. Gousseau and G. Tartavel. *Exemplar-based Texture Synthesis: the Efros-Leung Algorithm*. Image Processing On Line, vol. 3, pp. 223-241, 2013.

[Akl15] A. Akl, C. Yaacoub, M. Donias, J.-P. Da Costa and C. Germain. *Synthèse de Texture Contrainte par Champ de Structure Arbitraire*. 25th colloquium GRETSI, Lyon-France, September, 2015.

[Ang14] J. Angulo. *Structure Tensor Image Filtering using Riemannian L^1 and L^∞ Center-of-mass*. Image Analysis and Stereology, vol. 33, no. 2, pp. 95-105, 2014.

[Ars06] S. Arseneau and J. Cooperstock. *An Improved Representation of Junctions through Asymmetric Tensor Diffusion*. International Symposium on Visual Computing, November 2006.

[Ash01] M. Ashikhmin. *Synthesizing natural textures*. In SI3D: Proceedings of the 2001 symposium on Interactive 3D graphics, pp. 217-226, New York, NY, USA, ACM Press, 2001.

[Bar06] A. W. Bargteil, F. Sin, J. E. Michaels, T. G. Goktekin and J. F. O'Brien. *A Texture Synthesis Method for Liquid Animations*. Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006.

[Bat81] E.B. Batschelet. *Circular statistics in biology*. Academic Press, New York, 1981.

[Bee96] A. C. Beers, M. Agrawala and N. Chaddha. *Rendering from compressed textures*. Proceedings of SIGGRAPH, pp. 373-378, August 1996.

[Ber00] M. Bertalmio , G. Sapiro , V. Caselles and C. Ballester. *Image inpainting*. Proc. of the 27th annual conference on Computer graphics and interactive techniques, pp. 417-424, July 2000.

[Big87] J. Bigun and G. Granlund. *Optimal Orientation Detection of Linear Symmetry*. First int. Conf. on Computer Vision, ICCV, (London). Piscataway: IEEE Computer Society Press, Piscataway. pp. 433-438, 1987.

[Bor06] C. Bordenave, Y. Gousseau, and F. Roueff. *The dead leaves model: a general tessellation modeling occlusion*. Adv. Appl. Prob., vol. 38, no. 1, pp. 31-46, 2006.

[Bro] Colored Brodatz Texture Database. [Online]. Available: *http://multibandtexture.recherche.usherbrooke.ca/colored%20_brodatz_more.html*

[Bro66] P. Brodatz. *A Photographic Album for Artists and Designers*. Dover, New York, 1966.

[Bro06] T. Brox, R. van den Boomgaard, F. B. Lauze, J. van de Weijer, J. Weickert, P. Mrázek and P. Kornprobst. *Adaptive structure tensors and their applications*. in Visualization and Processing of Tensor Fields, Part 1, J. Weickert and H. Hagen, Springer Berlin Heidelberg, pp. 17-47, 2006.

[Bur83] P. J. Burt and E. H. Adelson. *A Multiresolution Spline With Application to Image Mosaics*. ACM Trans. on Graphics, vol. 2, no. 4, pp. 217-236, 1983.

[Cha14] C. Chapoullie. *Analyse/synthèse tridimensionnelle de textures fibreuses*. PhD dissertation, University of Bordeaux, France, 2014.

[Che85] R. Chellappa and R. L. Kashyap. *Texture synthesis using 2-D noncausal autoregressive models*. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP 33, no. 1, pp. 194-203, 1985.

[Che93] R. Chellappa, R. Kashyap and B. Manjunath. *Model based texture segmentation and classification*. Handbook of Pattern Recognition and Computer Vision, World Scientifc Publishing, pp. 277-310, 1993.

[Che98] H. Chen, L. F. Pau and P.S.P Wang. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, chapter 2.1, pp. 207-248, World Scientific Publishing Co., 1998.

[Cla87] M. Clark, A. C. Bovik and W. S. Geisler. *Texture segmentation using Gabor modulation/demodulation*. Pattern Recognition Letters, vol. 6, no. 4, pp. 261-267, 1987.

[Con80] R. W. Conners and C. Harlow. *A Theoretical Comparison of Texture Algorithms.* IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 2, no. 3, pp. 204-222, 1980.

[Cro83] G. Cross and A. Jain. *Markov Random Field Texture Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5, pp. 25-39, 1983.

[Dac01] J.-P. Da Costa. *Analyse statistique de textures directionnelles. Application à la caractérisation de matériaux composites*. PhD dissertation, University of Bordeaux I, France, 2001.

[Dac10] J.-P. Da Costa and C. Germain. *Synthesis of solid textures based on a 2D example: application to the synthesis of 3D carbon structures observed by transmission electronic microscopy*. Proc. SPIE, vol. 7538, Image Processing: Machine Vision Applications III, pp. 10, 2010.

[DeB97] J. S. DeBonet. *Multiresolution sampling procedure for analysis and synthesis of texture images*. In SIGGRAPH: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., pp. 361– 368, 1997.

[Dry09] I. L. Dryden, A. Koloydenko and D. Zhou. *Non-Euclidean Statistics for Covariance Matrices, with Applications to Diffusion Tensor Imaging*. The Annals of Applied Statistics, vol. 3, no. 3, pp. 1102-1123, 2009.

[Dub12] S. Dubois, R. Peteri and M. Menard. *Decomposition of dynamic textures using morphological component analysis*. IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 2, pp.188 -201, 2012.

[Dun00] J. S. Duncan and N. Ayache. *Medical image analysis: Progress over two decades and the challenges ahead*. IEEE Trans on PAMI. vol. 22, pp.85–106, 2000.

[Ebe02] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin and S. Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[Efr99] A. Efros and T. Leung. *Texture synthesis by non-parametric sampling*. In International Conference on Computer Vision, vol. 2, pp. 1033–1038, September 1999.

[Efr01] A. Efros and W. T. Freeman. *Image quilting for texture synthesis and transfer*. In SIGGRAPH: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 341-346, New York, NY, USA, ACM Press, 2001.

[Far14] B. Farbos, P. Weisbecker, H.E. Fischer, J-P. Da Costa, M. Lalanne, G. Chollon, C. Germain, G.L. Vignoles and J-M. Leyssale. *Nanoscale structure and texture of highly anisotropic pyrocarbons revisited with transmission electron microscopy, image processing, neutron diffraction and atomistic modeling*. Carbon 80, pp. 472–489, 2014.

[Fil05] P. Fillard, V. Arsigny, N. Ayache and X. Pennec. *A Riemannian framework for the processing of tensor-valued images*. in: Fogh Olsen, O., Florack, L.M.J., Kuijper, A. eds. DSSCV 2005. LNCS, vol. 3753, pp. 112–123. Springer, Heidelberg, 2005.

[For03] D. A. Forsyth and J. Ponce. *"Texture," in Computer Vision, A Modern Approach*. Pearson Education, 2003.

[Fra93] J. M. Francos, A. Z. Meiri and B. Porat. *A Unified Texture Model Based on a 2-D Wold-Like Decomposition*. IEEE Trans. Signal Processing, vol. 41, pp. 2665-2678, 1993.

[Gag81] A. P. Gagalowicz. *A new method for texture fields synthesis: Some applications to the study of human vision*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 3, no. 5, pp. 520-533, 1981.

[Gal11] B. Galerne, Y. Gousseau, and J.-M. Morel. *Micro-Texture Synthesis by Phase Randomization*. Image Processing On Line, vol. 1, 2011.

[Gal11 bis] B. Galerne, Y. Gousseau and J. -M. Morel. *Random Phase Textures: Theory and Synthesis*. IEEE Trans. on Image Processing, vol. 20, pp. 257-267, 2011.

[Gar08] R. de L.-García, R. Deriche, and C. Alberola-López. *Texture and color segmentation based on the combined use of the structure tensor and the image components*. Signal Processing, vol. 88, pp. 776-795, 2008.

[Gem84] S. Geman and D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 6, pp. 721-741, 1984.

[Ger97] C. Germain. *Caractérisation multi-échelle de l'anisotropie des textures*. PhD dissertation, University of Bordeaux I, France, 1997.

[Hai91] M. Haindl. *Texture synthesis*. CWI Quarterly, vol.4, pp. 305-331, 1991.

[Har79] R. M. Haralick. *Statistical and structural approaches to textures*. Proceedings of the IEEE, vol. 67, n.5, pp. 786–804, May 1979.

[Har92] R. M. Haralick and L.G. Shapiro. *"Texture," in Computer and Robot Vision*, vol. 1, Addison Wesley, 1992.

[Hec86] P. Heckbert. *Survey of Texture Mapping*. Pixar, IEEE Computer Graphics and Applications, Nov. 1986.

[Hee95] D. J. Heeger and J. R. Bergen. *Pyramid-based texture analysis/synthesis*. In SIGGRAPH , pp. 229-238, New York, NY, USA, 1995.

[Her01] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless and D. H. Salesin. *Image analogies*. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 327-340, New York, NY, USA, ACM Press, 2001.

[Ige97] H. Igehy and L. Pereira. *Image replacement through texture synthesis*. In International Conference on Image Processing, vol. 3, pp. 186-189, Oct 1997.

[ITU11] ITU-R Recommendation BT.601-7. *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios*. ITU-R, March 2011.

[Jac11] L. Jacques, L. Duval, C. Chaux and G. Peyré. *A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity*. Signal Processing, vol. 91(2011), pp. 2699-2730, Elsevier, 2011.

[Jäh93] B. Jähne. *Spatio-Temporal Image Processing: Theory and Scientific Applications*. vol. 751. Springer-Verlag, Berlin, 1993.

[Jay13] S. Jayasumana1, R. Hartley, M. Salzmann, H. Li and M. Harandi. *Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices*. IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[Kaa87] M. Kaas and A. Witkin. *Analyzing oriented patterns*. Comput.Vision Graph. Image Process., vol. 37, no. 3, pp. 362-385, 1987.

[Knu89] H. Knutsson. *Representing local structure using tensors*. Proc. the 6th Scandinavian Conf. on Image Analysis, Oulu, Finland, pp. 244–251, 1989.

[Kop07] J. Kopf, C.W. Fu, D. Cohen-Or, O. Deussenn, D. Lischinski and T.T. Wong. *Solid Texture Synthesis from 2D Exemplars.* ACM SIGGRAPH, TOG, vol. 26, no. 3, 2007.

[Köp12] M. Köppel, X. Wang, D. Doshkov, T. Wiegand and P. Nd jiki-Nya. *Depth Image-based Rendering with Spatio-Temporally Consistent Texture Synthesis for 3-D Video with Global Motion.* Proceedings of the IEEE International Conference on Image Processing (ICIP), Orlando, USA, September 2012.

[Kul51] S. Hullback and R. A. Leibler. *On Information and Sufficiency*. Annals of Mathematical Statistic, vol. 22, no. 1, pp. 79-86. doi:10.1214/aoms/1177729694 .MR 39968, 1951.

[Kwa03] V. Kwatra, A. Schödl, I. A. Essa, G. Turk and A. F. Bobick. *Graphcut textures: Image and video synthesis using graph cuts*. In ACM Transactions on Graphics, SIGGRAPH, vol. 22, no. 3, pp. 277-286, July 2003.

[Lag09] A. Lagae, S. Lefebvre, G. Drettakis and P. Dutre. *Procedural noise using sparse Gabor convolution*. In SIGGRAPH, vol. 28, no. 3, August 2009.

[Ley09] J. -M. Leyssale, J. -P. Da Costa, C. Germain, P. Weisbecker and G. Vignoles. *An image-guided atomistic reconstruction of pyrolytic carbons.* Applied Physics Letters, vol. 95, 2009.

[Ley12] J.-M. Leyssale, J.-P. Da Costa, C. Germain, P. Weisbecker and G. L. Vignoles. *Structural features of pyrocarbon atomistic models constructed from transmission electron microscopy images*. Carbon, vol. 50, no. 12, pp. 4388-4400, 2012.

[Lia01] L. Liang, C. Liu, Y. Q. Xu, B. Guo and H. Shum. *Real-time texture synthesis by patch-based sampling*. ACM Transactions on Graphics, vol. 20, no. 3, pp. 127– 150, 2001.

[Lin04] W. -C. Lin, J. H. Hays, C. Wu, V. Kwatra and Y. Liu. *A comparison study of four texture synthesis algorithms on regular and near-regular textures.* Technical report, Carnegie Mellon University, 2004.

[Lyv88] E.P. Lyvers and O.R. Mitchell. *Precision edge contrast and orientation estimation*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pp. 927-937, 1988.

[Mal89] S. Mallat. *Multifrequency Channel Decomposition of Images and Wavelet Models*. IEEE Trans. Acoustic, Speech and Signal Processing, vol. 37, no. 12, pp. 2091-2110, 1989.

[Mar00] K. Mardia, J. Jupp. *Directional Statistics*. John Wiley, England, 2000.

[Merr] Merriam-Webster's Collegiate Online Dictionary. [Online]. Available: *www.merriam-webster.com*

[Mun08] E. Munch, M.E. Launey, D.H. Alsem, E. Saiz, A.P. Tomsia and R.O. Ritchie. *Tough, Bio-Inspired Hybrid Materials*. Science Magazine, vol. 322, no. 5907, pp. 1516-1520, December 2008.

[Pag98] R. Paget and I. Longstaff. *Texture synthesis via a noncausal nonparametric multiscale Markov random field*. In IEEE Transactions on Image Processing, vol. 7, no. 6, pp. 925-931, 1998.

[Pei88] H. Peitgen and D. Saupe. *The Science of Fractal Images*. Springer-Verlag, New York, 1988.

[Pen06] X. Pennec, P.Fillard, and N. Ayache. *A Riemannian Framework for Tensor Computing*. International Journal of Computer Vision, vol. 66, no. 1, pp. 41-66, January 2006.

[Per90] P. Perona and J. Malik. *Scale-space and edge detection using anisotropic diffusion*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629-639, July 1990.

[Pey09] G. Peyré. *Texture Synthesis with grouplets*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 733-746, 2009.

[Pop93] K. Popat and R. W. Picard. *Novel cluster-based probability model for texture synthesis, classification, and compression*. In Proceedings SPIE visual Communications and Image Processing, Boston, 1993.

[Por00] J. Portilla and P. Simoncelli. *A parametric texture model based on joint statistics of complex wavelet coefficients*. International Journal of Computer Vision, vol. 40, no. 1, pp. 49-71, 2000.

[Pre70] J.M.S. Prewitt. *Object Enhancement and Extraction*. in Picture processing and Psychopictorics, Academic Press, 1970.

[Rab10] J. Rabin, G. Peyre, J. Delon and M. Bernot. *Wasserstein barycenter and its application to texture mixing*. Technical report, 2010.

[Rao89] A.R. Rao and B.G. Schunck. *Computing Oriented Texture Fields*. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'89, San Diego, CA, pp. 61-68, June 4-8, 1989.

[Rao90] A.R. Rao. *A taxonomy for Texture Description and Identification*, Springer Verlag, New York, 1990.

[Rao91] A.R. Rao and B.G. Schunck. *Computing oriented texture fields*. CVGIP: Graphical Models Image Process, vol. 53, no. 2, pp. 157-185, March 1991.

[Sav93] G. Savage. *Carbon-carbon composites*. Eds. Chapman & Hall, London, 1993.

[Sch00] A. Schodl, R. Szeliski, D. H. Salesin and I. Essa. *Video textures*. In Computer Graphics, ACM SIGGRAPH Proceedings, pp. 489-498, New Orleans, July 2000.

[Ser82] J. Serra, Ed., *Image Analysis and Mathematical Morphology*. London: Academic press, vol. 1, 1982.

[Sil86] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986.

[Sim92] E.P. Simoncelli, W.T. Freeman, E.H. Adelson and D.J. Heeger. *Shiftable Multi-Scale Transforms.* IEEE Trans. Information Theory, vol. 38, no. 2, pp. 587-607, March 1992.

[Skl78] J. Sklansky. *Image Segmentation and Feature Extraction*. IEEE Transactions on Systems, Man, and Cybernetics, SMC-8, pp. 237-247, 1978.

[Sra12] S. Sra. *Positive Definite Matrices and the Symmetric Stein Divergence*. the Neural Information Processing Systems (NIPS) Conference, December 2012.

[Sti07] Y. Stitou, F. Turcu, Y. Berthoumieu and M. Najim. *Three-Dimensional Textured Image Blocks Model Based on Wold Decomposition*. IEEE Transactions on Signal Processing, vol. 55, no. 7, pp. 3247-3261, 2007.

[Sun10] W. Sun, Y. Lu, F. Wu, S. Li and J. Tardif. *High-Dynamic-Range Texture Compression for Rendering Systems of Different Capacities*. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 1, pp. 57-69, Jan.-Feb. 2010.

[Szu96] M. Szummer and R. W. Picard. *Temporal texture modeling*. In International Conference on Image Processing, vol. 3, pp. 823–826, September 1996.

[Tar14] G. Tartavel, Y. Gousseau, and G. Peyré. *Variational texture synthesis with sparsity and spectrum constraints.* To appear in Journal of Mathematical Imaging and Vision, 2014. [Online]. Available: *http://hal.archives-ouvertes.fr/hal-00881847/.*

[Ton02] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo and H. Shum. *Synthesis of bidirectional texture functions on arbitrary surfaces*. In SIGGRAPH: Proceedings of the 29th annual conference on Computer graphics and interactive techniques. ACM Press, pp. 665–672, 2002.

[Tou10] V. Toujas, M. Donias and Y. Berthoumieu. *Structure Tensor Field Regularization Based on Geometric Features*. 18th European Signal Processing Conference (EUSIPCO), August 23-27, 2010.

[Tur86] M. R. Turner. *Texture discrimination by Gabor functions*. Biological Cybernetics, vol. 55, no. 2-3, pp. 71-82, 1986.

[Uns84] M. Unser. *Description statistique de la texture. Application à l'inspection automatique*. PhD dissertation, EPFL, Lausanne, 1984.

[Urs12] R. Urs, J.-P. Da Costa, J. -M. Leyssale, G. Vignoles and C. Germain. *Non-parametric synthesis of laminar volumetric textures from a 2D sample*. Proc. British Machine Vision Conference, pp. 54.1-54.11, 2012.

[Urs13] R. Urs. *Non-parametric synthesis of volumetric textures from a 2D sample*. PhD dissertation, University of Bordeaux, France, 2013.

[Urs14] R. Urs, J.-P. Da Costa, and C. Germain. *Maximum-Likelihood Based Synthesis of Volumetric Textures from a 2D Sample*. IEEE Trans. On Image Processing, vol. 23, no. 4, pp. 1820-1830, 2014.

[Wan04] Z. Wang, B. Vemuri, Y. Chen and T. Mareci. *A constrained variational principle for direct estimation and smoothing of the diffusion tensor field from complex DWI*. IEEE Trans. Med. Imaging. vol. 23, no. 8, pp. 930-939, 2004.

[Wei00] L. -Y. Wei and M. Levoy. *Fast texture synthesis using tree-structured vector quantization*. In SIGGRAPH, 27th International Conference on Computer Graphics and Interactive Techniques, pp. 479-488, 2000.

[Wei01] L. -Y. Wei. *Texture synthesis by fixed neighborhood searching*. PhD dissertation, Stanford University, November 2001.

[Wei03] L. -Y. Wei. *Texture synthesis from multiple sources*. Proceedings of ACM SIGGRAPH Sketches & Applications, New York, 2003.

[Wei04] J. van de Weijer and T. Gevers. *Tensor Based Feature Detection for Color Images*. 12th Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, pp. 100-105, November 2004.

[Wij91] J. J. van Wijk. *Spot noise texture synthesis for data visualization*. In SIGGRAPH '91. New York, NY, USA: ACM, pp. 309-318, 1991.

[Wik01] K. Wikantika, A. Harto and R. Tateishi. *The use of spectral and textural features from Landsat TM image for land cover classification in mountainous area*. Proceedings of the IECL Japan workshop, Tokyo, 2001.

[Win94] G. Winkenbach and D. H. Salesin. *Computer-generated pen-and-ink illustration*. Proc. of SIGGRAPH 94, pp. 91–100, 1994.

[Wor96] S. Worley: *A cellular texture basis function*. In SIGGRAPH: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM Press, pp. 291-294, 1996.

[XuG00] Y. Q. Xu, B. Guo and H. Shum. *Chaos mosaic: Fast and memory efficient texture synthesis*. In Tech. Rep. MSRTR-2000-32, Microsoft Research, 2000.

[Yam03] H. Yamauchi, J. Haber and H-P. Seidel. *Image restoration using multiresolution texture synthesis and image inpainting*. in Proc. Int. Conf. Comput. Graph., 2003.

[Zel02] S. Zelinka and M. Garland. *Towards real - time texture synthesis with the jump map*. In EGRW: Proceedings of the 13th Eurographics workshop on Rendering, Eurographics Association, pp. 99–104, 2002.

[Zen86] S. D. Zenzo. *A note on the gradient of a multi-image*. Int'l J. Computer Vision, Graphics Image Process., vol. 33, pp. 116-125, 1986.

Liste des publications de l'auteur

Articles de conférence et de journal

A. Akl, C. Yaacoub, M. Donias, J.-P. Da Costa and C. Germain. *Texture Synthesis Using the Structure Tensor*. IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 4082-4095, 2015.

A. Akl, C. Yaacoub, M. Donias, J.-P. Da Costa and C. Germain. *Synthèse de Texture Contrainte par Champ de Structure Arbitraire*. 25th colloquium GRETSI, Lyon-France, September 2015.

A. Akl, C. Yaacoub, M. Donias, J.-P. Da Costa and C. Germain. *Two-Stage Color Texture Synthesis Using the Structure Tensor Field*. the 10th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), Berlin-Germany, March 2015.

A. Akl, C. Yaacoub, M. Donias, J.-P. Da Costa and C. Germain. *Structure Tensor Based Synthesis of Directional Textures for Virtual Material Design*. invited paper - the 21st IEEE International Conference on Image Processing (ICIP), Special Session – Image Processing for Materials Characterization, Paris-France, October 2014.

Rapports d'avancement

J.-P. Da Costa, B. Farbos, C. Germain, R. Urs and **A. Akl**. *PyroMaN project. Progress in 2D/3D texture synthesis.* Fungraph/PyroMaN joint meeting, March, 2014.

Deux rapports, concernant l'état d'avancement des travaux de thèse, ont été rédigés dans le cadre de présentations annuelles à l'Ecole Doctorale - Sciences Physiques et de l'Ingénieur (SPI), en Mai 2014 et Juin 2015.

Publications de l'auteur sortant du cadre de la thèse

A. Akl and C. Yaacoub. *Accelerated Joint Image Despeckling Algorithm in the Wavelet and Spatial Domains*. International Journal of Image Processing (IJIP), vol. 9, no. 3, pp. 102-126, 2015.

A. Akl. *Theory and Practical Fusion in Engineering Education*. European Journal of Educational Sciences (EJES), vol. 1, no. 1, pp. 7-18, 2014.

A. Akl and C. Yaacoub. *A Hybrid Wavelet-Spatial Denoising Filter*. 18th International Conference on Digital Signal Processing (DSP), Santorini-Greece, July 2013.

A. Akl and C. Yaacoub. *A Hybrid Filter for Image Despeckling with Wavelet-Based Denoising and Spatial Filtering*. 3rd International Conference on Communications and Information Technology (ICCIT), AUB-Beirut-Lebanon, June 2013.

A. Akl, K. Tabbara and C. Yaacoub. *An Enhanced Kuan Filter for Suboptimal Speckle Reduction*. 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), NDU-Lebanon, December 2012.

A. Akl, K. Tabbara and C. Yaacoub. *Edge-Based Automatic Parameter Selection for Sub-optimal Image Despeckling using Kuan Filter*. 8th International Conference on Signal Image Technology and Internet Based Systems (SITIS), Naples-Italy, November 2012.

A. Akl, K. Tabbara and C. Yaacoub. *Edge-Based Suboptimal Kuan Filtering for Speckle Noise Reduction*. the 2012 International Conference on Engineering and Technology (ICET), Cairo-Egypt, October 2012.