

Synthèse de textures volumiques basée sur la vraisemblance de patches

RADU-DRAGOS URS^{1,2}, JEAN-PIERRE DA COSTA^{1,2}, CHRISTIAN GERMAIN^{1,2}

¹ Univ. Bordeaux, IMS, UMR 5218

² CNRS, IMS, UMR 5218

radu.urs@ims-bordeaux.fr, jean-pierre.dacosta@ims-bordeaux.fr, christian.germain@ims-bordeaux.fr

Résumé - Nous proposons une approche probabiliste non-paramétrique basée sur la vraisemblance de patches pour la synthèse de textures volumiques. Une seule image 2D est utilisée pour contraindre plusieurs vues orthogonales de la texture 3D. Considérant que les textures sont des réalisations d'un processus de Markov, l'objectif de l'algorithme est de reproduire, sur le volume texturé, les interactions entre pixels estimées de façon non-paramétrique sur l'image d'entrée. Le processus de synthèse s'appuie sur l'échantillonnage d'une seule image 2D en entrée mais garantit la cohérence selon les différentes vues de la texture 3D. Différentes variantes sont proposées, relatives aux stratégies de gestion simultanée des tranches orthogonales contenant le voxel courant. L'approche est mise en œuvre sur un jeu de textures structurées, de régularité et d'anisotropie variées. Une analyse comparée et une étude de sensibilité sont menées, mettant en évidence les atouts et faiblesses des différentes variantes, et permettant de les comparer à certains algorithmes non-paramétriques de la littérature.

1 Introduction

1.1 Contexte

Le domaine de la synthèse de texture est particulièrement dynamique avec des applications nombreuses dans l'extrapolation, la compression d'images, l'*inpainting* ou le *mapping* mais aussi dans d'autres domaines comme la fusion et le montage vidéo, la description de la géométrie d'une surface ou encore la synthèse de matériaux virtuels. Parmi les approches existantes, les techniques de synthèse de textures à partir d'un exemple sont parmi les plus populaires. Ces techniques, souvent 2D, demandent à être adaptées en vue de la modélisation de structures volumiques telles que celles obtenues par des techniques d'imagerie 3D (e.g. imagerie médicale, tomographie, imagerie sismique). Toutefois, pour des raisons de coût, de praticité ou simplement de résolution, l'obtention de textures volumiques par des techniques d'imagerie 3D n'est pas toujours envisageable. La synthèse de textures 3D doit alors se faire par des approches 2D/3D extrapolant en 3D une information capturée en 2D sur une texture exemple. Outre les difficultés liées à leur complexité calculatoire, de telles approches posent des problèmes stéréologiques d'inférence 2D/3D et ne sont envisageables que pour des textures isotropes ou, dans le cas de textures anisotropes – lamellaires ou filaires – sous certaines hypothèses bien définies.

1.2 Objectif

Parmi les techniques de synthèse d'images texturées, les approches dites *par patch* sont parmi les plus populaires. Les approches non paramétriques basées sur la recherche de voisinages fixes, sont connues pour leur simplicité et leurs performances, certaines extensions 2D/3D ayant été proposées [1-4]. Bien qu'efficaces, ces dernières montrent cependant des limites, se révélant parfois incapables de synthétiser certaines textures. Leur

défaut le plus connu est leur propension à répéter verbatim certains motifs de l'image d'entrée.

Nous traitons ici d'une approche probabiliste, qui réalise de façon explicite ce que les méthodes basées sur la recherche de voisinage fixe tentent d'accomplir indirectement - distribuer les pixels et les configurations locales d'entrée dans la texture de sortie, en reproduisant le plus fidèlement possible les caractéristiques de la texture d'entrée. Notre méthode étend l'approche 2D markovienne initialement proposée par Paget et Longstaff [5]. Dans l'approche 3D que nous explorons, la synthèse est effectuée voxel par voxel, en maximisant la vraisemblance de patches 2D extraits de tranches orthogonales. L'approche s'appuie sur l'échantillonnage d'un seul modèle 2D d'entrée en garantissant la cohérence selon plusieurs vues de la texture 3D.

2 Algorithme 2D/3D basé sur la vraisemblance

2.1 Principes théoriques

S'appuyant sur les propriétés de stationnarité et de localité de la texture, l'approche markovienne [5] consiste à générer une texture pixel par pixel, en maximisant la vraisemblance de chaque pixel au sens de la fonction de densité de probabilité conditionnelle locale (FDPCL) [5-7]. Autrement dit, on affecte à chaque pixel s le niveau de gris le plus probable connaissant son voisinage V_s . Ce procédé requiert la connaissance de la FDPCL, modélisant de façon unique les interactions entre pixels voisins, et révélant ainsi la markovianité : *la variable aléatoire X_s décrivant l'intensité d'un pixel s peut prendre n'importe quelle valeur x_s de l'ensemble Λ de niveaux de gris de l'image d'entrée, mais la probabilité que $X_s = x_s$ ne dépend que des valeurs des pixels voisins $r \in V_s$ [6,7] :*

$$\mathcal{L}(p) = P(X_s = x_s | X_r = x_r, r \neq s) = P(x_s | x_r, r \in V_s) \quad (1)$$

La FDPCL est estimée à la volée sur l'image d'entrée, de façon non paramétrique par la technique de

fenêtrage de Parzen [8], l'estimateur de densité étant défini par:

$$\hat{f}(z) = \frac{1}{n \cdot h^d} \sum_{p \in Y, N_p \subset Y} K\left(\frac{1}{h}(z - z_p)\right) \quad (2)$$

où K est un noyau multi-varié. Le paramètre de fenêtre h définit la portée du noyau, n le nombre de voisinages possibles V_p de taille d dans l'image d'entrée Y . Le modèle markovien s'appuie ainsi sur l'estimation de la FDPCL [5]:

$$\hat{L}(s) = \hat{P}(x_s | x_r, r \in V_s) = \frac{\hat{f}(z_{x_s} = Col[x_s, x_r, r \in V_s])}{\sum_{\lambda_s \in \Lambda} \hat{f}(z_{\lambda_s} = Col[\lambda_s, x_r, r \in V_s])} \quad (3)$$

où le vecteur colonne $z_p = Col[y_p, y_q, q \in V_p]$ contient le pixel y_p et toutes les valeurs à l'intérieur de son voisinage V_p de taille d et $z_{x_s} = Col[x_s, x_r, r \in V_s]$ est une configuration de l'image de sortie X , sur laquelle l'estimateur de Parzen est appliqué. En choisissant un noyau gaussien multi-varié sphérique et en exploitant l'équation (3), la FDPCL devient alors:

$$\hat{L}(s) = \frac{\sum_{p \in Y, N_p \subset Y} \exp\left[-\frac{1}{2h^2}(z_{x_s} - z_p)^T(z_{x_s} - z_p)\right]}{\sum_{\lambda_s \in \Lambda} \sum_{p \in Y, N_p \subset Y} \exp\left[-\frac{1}{2h^2}(z_{\lambda_s} - z_p)^T(z_{\lambda_s} - z_p)\right]} \quad (4)$$

2.2 Extension 2D/3D

2.2.1 Décomposition de voisinage

L'approche 2D/3D que nous proposons repose sur les mêmes principes. Elle consiste à synthétiser la texture volumique voxel par voxel, en mettant à jour un voxel de façon à maximiser sa vraisemblance:

$$\lambda_s^* = \arg \max \{L_{3D}(\lambda_s)\} = \arg \max \{\hat{P}(\lambda_s | x_r, r \in V_s^{3D})\} \quad (5)$$

où V_s^{3D} définit un système de voisinage 3D.

Toutefois, l'image modèle étant bidimensionnelle, la FDPCL 3D, ne peut pas être estimée. La synthèse 3D est alors envisagée selon une approche multi-2D, à l'instar des approches basées sur la recherche de voisinages fixes [4]. On propose de considérer un système de voisinage 3D composé de plusieurs plans orthogonaux, comme illustré à la Fig. 1:

$$V_{s=(i,j,k)}^{3D} = V_{s=(i,j)}^{face} \cup V_{s=(j,k)}^{droite} \cup V_{s=(i,k)}^{haut} \quad (6)$$

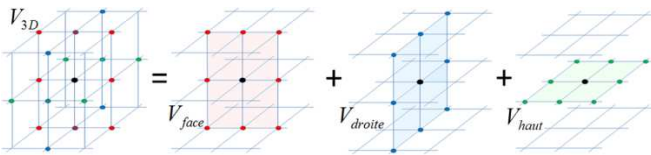


Figure 1 : Remplacement du voisinage 3D par trois voisinages 2D, selon les vues orthogonales (face, droite, haut), sur lesquelles seront estimées les FDPCL 2D

La fonction densité de probabilité conditionnelle 3D est remplacée par une combinaison des FDPCL 2D relatives aux plans orthogonaux. Selon la nature de la texture à synthétiser, trois vues ou deux vues seulement peuvent être exploitées. La valeur de voxel choisie doit maximiser la vraisemblance exprimée sur chacune des vues orthogonales auxquelles le voxel appartient, i.e. vues de face, de droite et de haut. Ces FDPCL 2D sont estimables sur l'image d'entrée. Il reste toutefois à trouver une stratégie permettant la maximisation des vraisemblances exprimées selon les différentes vues.

2.2.2 Heuristique H1

Le niveau de gris à affecter au voxel v_s peut être construit comme une combinaison des niveaux de gris les plus probables trouvés séparément pour chacune des vues orthogonales, par trois recherches 2D/2D indépendantes. Les couples niveaux de gris / FDPCL notés $(\lambda_{face}, \hat{P}_{face})$, $(\lambda_{droite}, \hat{P}_{droite})$ et $(\lambda_{haut}, \hat{P}_{haut})$, sont obtenus grâce à des maximisations distinctes:

$$\lambda_{vue}^* = \arg \max \{\hat{P}_{vue}(\lambda_{vue} | x_r, r \in V_s^{vue})\}, \lambda_{vue} \in \Lambda \quad (7)$$

Différentes stratégies peuvent être utilisées pour la combinaison comme par exemple:

- le choix de la plus faible FDPCL, stratégie *H1a*;
- la moyenne des trois solutions, stratégie *H1b*.

Cette dernière stratégie rappelle les algorithmes basés sur la recherche du plus proche voisinage, où la valeur de mise à jour pour un voxel est calculée comme une combinaison des solutions trouvées pour les vues orthogonale [1-4].

2.2.3 Heuristique H2

Afin d'éviter d'effectuer plusieurs recherches et de combiner plusieurs solutions, une deuxième heuristique est proposée. On utilise une fonction \mathcal{F} des estimations 2D associés aux vues orthogonales, par exemple:

$$\mathcal{F}(L_{face}, L_{droite}, L_{haut}) = \min(L_{face}, L_{droite}, L_{haut}) \quad (8)$$

$$\mathcal{F}(L_{face}, L_{droite}, L_{haut}) = \text{produit}(L_{face}, L_{droite}, L_{haut}) \quad (9)$$

La maximisation de cette fonction permet de trouver un niveau de gris λ_s^* de mise à jour unique:

$$\lambda_s^* = \arg \max \left\{ \mathcal{F}(L_{face}(\lambda_s), L_{droite}(\lambda_s), L_{haut}(\lambda_s)) \right\}, \lambda_s \in \Lambda$$

où $L_{vue}(\lambda_s) = \hat{P}_{vue}(\lambda_s | x_r, r \in V_s^{vue})$ (10)

Ces heuristiques sont identifiées comme suit:

- *H2a* pour l'expression (8) qui maximise la plus petite des probabilités de façon à les maximiser toutes;
- *H2b* pour \mathcal{F} défini en (9) afin de maximiser le produit, comme un compromis pour les trois vues.

2.2.4 Algorithme générique 2D/3D

Les différentes solutions proposées reposent sur une variante multi-échelle [5] de l'algorithme de relaxation déterministe de type ICM (*Iterated Conditional Modes*) [7]. Le principe de la synthèse est de débiter, à l'échelle la plus haute, par un bloc 3D aléatoire, de distribution marginale identique à celle de l'image d'entrée à la même échelle. L'optimisation est gérée voxel par voxel, selon un ordre de parcours aléatoire. La valeur d'un voxel est modifiée par maximisation des vraisemblances 2D par vue. Une fois le volume synthétisé à l'échelle la plus haute, il sert d'initialisation à l'échelle inférieure: les voxels d'une échelle sont recopiés sur les voxels de coordonnées paires à l'échelle inférieure.

Le calcul des vraisemblances est réalisé en donnant d'autant plus d'importance à un voxel voisin que la confiance qui lui est accordée est forte. La gestion de cette confiance est assurée par une fonction de température [5], qui donne à chaque pixel une certaine température $t_s \in [0,1]$ (où 0 signifie confiance totale) et qui conditionne la FDPCL de la manière suivante:

$$(z_{x_s} - z_p) = Col[x_s - y_p, (x_r - y_r)(1 - t_r), r \in V_s] \quad (11)$$

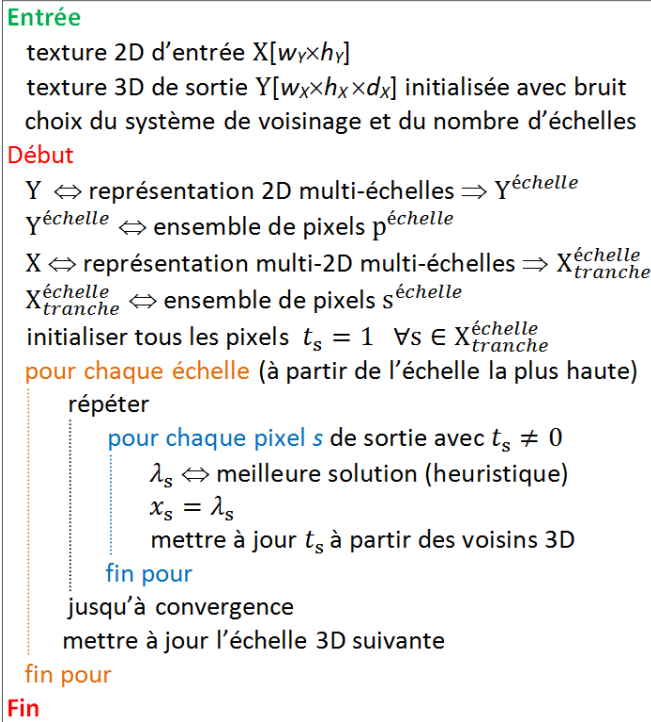


Figure 2 : Algorithme 2D/3D basé sur la vraisemblance de patches 2D

La synthèse démarre à une température globale élevée (méfiance) et le processus est considéré comme terminé lorsque la température est nulle pour tous les pixels. Les schémas de décroissance des températures des voxels sont gérés individuellement. Chaque fois qu'un voxel est traité, sa température est mise à jour :

$$t_s = \max\{0, (\xi + \sum_{r \in V_s} t_r) / |V_s|\}, \quad \xi < 0 \quad (12)$$

où ξ est utilisé pour contrôler la vitesse à laquelle un pixel est refroidi sur la base des températures voisines. Les voxels hérités de l'étage supérieur se voient doter d'une température plus basse.

L'algorithme multi-échelle complété par le schéma de gestion de confiances est donné en Fig. 2.

3 Expérimentation

Les différentes stratégies proposées ont été mises en œuvre pour la synthèse d'un jeu de textures variées, principalement de textures de Brodatz [9]. L'objectif est ici de discuter du potentiel des différentes stratégies mais également du rôle et de l'effet des paramètres communs. En termes de qualité, un point critique consiste à prendre en compte la taille des motifs de la texture d'entrée pour déterminer la taille du voisinage et le nombre d'échelles. Le schéma de décroissance de la température et la condition d'arrêt sont également des points importants.

Pour montrer la capacité des algorithmes proposés, les résultats sont obtenus en utilisant un cadre générique qui empiriquement s'est avéré produire des résultats satisfaisants : un voisinage carré de taille 9×9 , trois échelles et une synthèse des voxels selon un parcours aléatoire. Afin d'assurer un temps de calcul raisonnable et des résultats de bonne qualité, la température des pixels est diminuée avec un pas $\xi = -3$ et le processus de relaxation est arrêté après avoir atteint une diminution

de 95% de la température globale, pour faire la synthèse d'un bloc 3D de 64^3 pixels à partir d'un échantillon de 64^2 pixels.

Des précautions sont prises pour s'assurer que l'extrapolation de la texture en 3D a du sens en termes d'anisotropie. L'existence d'une structure 3D réaliste, donnant lieu à des coupes 2D similaires à l'exemple, doit être possible. La plupart des textures 2D employées ne peuvent pas être observées sur les trois vues 2D orthogonales d'une même texture volumique, mais peuvent l'être sur deux vues seulement. Par conséquent, la synthèse de textures 3D est effectuée en contraignant seulement deux vues du bloc de sortie avec l'échantillon (vue de face et de droite). Cette base est considérée pour les résultats suivants, en mentionnant pour chaque cas, les paramètres pris en compte.

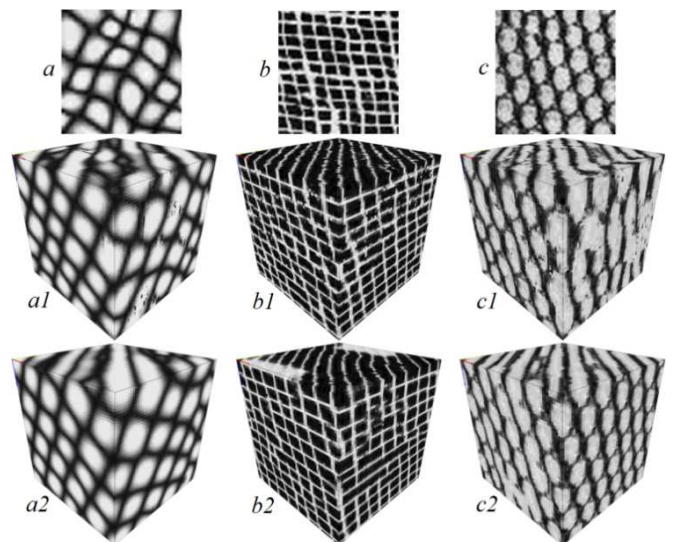


Figure 3 : Résultats volumiques – chaque colonne montre de haut en bas les résultats synthétisés à partir de l'exemple en 1^{ère} ligne, en utilisant les heuristiques *H1b* (ligne 2) et *H2b* (ligne 3)

Quelques résultats obtenus avec l'heuristique *H1b* et *H2b* sont présentés à la figure 3. En utilisant *H1b*, des artefacts indésirables apparaissent sur presque tous les résultats (Fig. 3a1-c1). La stratégie de moyennage n'est pas payante : elle affecte aux voxels des niveaux de gris qui n'existent pas dans l'échantillon 2D, ce qui perturbe le processus de synthèse. Les textures produites présentent des cassures. L'heuristique *H2b* semble mieux capturer les caractéristiques visuelles de la texture d'entrée. Les textures synthétiques (Fig. 3a2-c2) sont plus nettes, presque sans artefacts indésirables (mis à part quelques petits défauts de structure). Elles sont très similaires aux échantillons, donnant l'impression qu'elles ont été produites par le même processus.

Les remarques ci-dessus justifient le choix de la stratégie *H2b* pour la suite, comme l'heuristique la plus convaincante pour les approches fondées sur la maximisation de la vraisemblance. A la figure 4, nous comparons cette stratégie à la méthode proposée par Kopf et al. [2]. Cette dernière s'est avérée être représentative des méthodes basées sur la recherche de voisinage fixe [1-4].

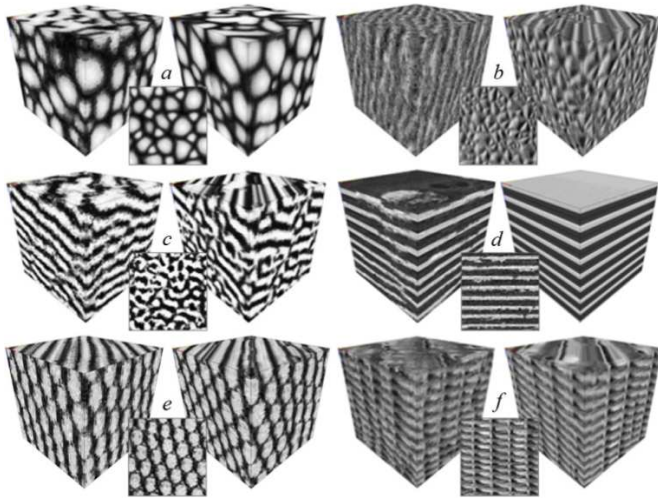


Figure 4 : Exemples de textures volumiques - chaque triplet contient au-dessous l'échantillon 2D, à gauche le résultat obtenu avec la méthode [2] et à droite avec notre algorithme 2D/3D

Les textures volumiques produites avec l'extension probabiliste sont d'une qualité visuelle convaincante, notamment en termes de dynamique. Par rapport à la méthode de Kopf et al. [2], l'approche de type maximisation de vraisemblance dispose de paramètres supplémentaires, parfois délicats à contrôler. Lorsqu'ils sont choisis correctement, des résultats très satisfaisants peuvent toutefois être obtenus, surpassant ceux obtenus avec les méthodes de recherche de voisinage, au prix toutefois d'un coût calculatoire plus élevé.

Dans de nombreux cas, comme en Fig. 4e et 4f, les deux types d'approches fournissent des résultats satisfaisants, à la fois en termes de structure et de dynamique. Pour les textures présentées en Fig. 4a-c notre approche se comporte mieux que l'approche basée sur la recherche de voisinage fixe qui s'avère incapable de capturer la structure d'entrée.

La tendance des algorithmes de recherche de voisinage à répéter certains motifs texturaux, rendant ainsi les résultats de synthèse plus réguliers que l'échantillon [4] (phénomène observé ici en Fig. 4c) est atténuée avec l'approche basée sur la vraisemblance. Pour celle-ci, les propriétés de structure et de périodicité des images sont souvent mieux respectées même si dans certains cas les textures 3D sont trop ordonnées (comme l'exemple en Fig. 4d, où on observe une simplification de la structure et une réduction de la dynamique). C'est un effet lié à l'algorithme de relaxation déterministe ICM, qui tend à ne retenir que les configurations de niveaux de gris les plus présentes (i.e. les plus probables) rendant la texture 3D plus simple. C'est pourquoi l'emploi d'un algorithme de relaxation stochastique semble une piste digne d'intérêt.

L'intégration dans la synthèse d'un algorithme de relaxation stochastique de type recuit simulé, a permis d'obtenir des résultats préliminaires satisfaisants (voir les exemples en Fig. 5). Mais en l'état actuel des implémentations, ces résultats restent toutefois comparables à ceux obtenus avec l'algorithme ICM. Les textures synthétisées par des relaxations déterministe ou stochastique peuvent difficilement être différenciées.

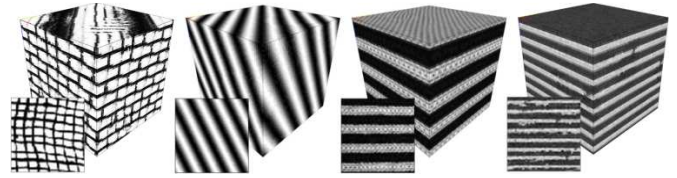


Figure 5 : Textures volumiques obtenues avec notre extension 2D/3D et une relaxation stochastique de type recuit simulé

4 Conclusions

Nous avons consacré notre réflexion à l'élaboration d'un algorithme probabiliste pour la synthèse de textures 3D à partir d'un exemple 2D. S'appuyant sur un paradigme markovien 2D, nous avons proposé une extension 2D/3D inédite visant à générer des textures volumiques. On maximise la vraisemblance de chaque voxel au sens de la densité de probabilité conditionnelle locale, au moyen d'un algorithme de relaxation. Différentes variantes sont proposées, relatives aux stratégies de gestion simultanée des tranches orthogonales contenant le voxel. Elles ont été appliquées à un jeu de textures structurées, de régularité et d'anisotropie variées, puis comparées aux algorithmes de type recherche de voisinage. Des résultats visuellement satisfaisants ont été obtenus. Pour autant, certains résultats laissent entrevoir des perspectives d'améliorations. Une attention particulière devra également être portée à la complexité calculatoire.

5 Bibliographie

- [1] L.-Y. Wei et M. Levoy. *Texture synthesis from multiple sources*. Proceedings of ACM SIGGRAPH Sketches & Applications, New York, 2003.
- [2] J. Kopf, C. W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski et T. T. Wong. *Solid Texture Synthesis from 2D Exemplars*. SIGGRAPH, vol. 26, issue 3, 2007.
- [3] J. Chen et B. Wang. *High quality solid texture synthesis using position and index histogram matching*. The Visual Computer, vol.26, pp. 253-262, 2010.
- [4] R. Urs, J.-P. Da Costa, J.-M. Leyssale, G. Vignoles et C. Germain. *Non-parametric synthesis of laminar volumetric textures from a 2D sample*. Proceedings of British Machine Vision Conference, pp.54.1-54.11, 2012.
- [5] R. Paget et I. Longstaff. *Texture synthesis via a noncausal nonparametric multiscale Markov random field*. IEEE Trans. on Image Processing 7, 6, pp. 925-931, 1998.
- [6] S. Geman et D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 6, no. 6, pp. 721-741, 1984.
- [7] J. E. Besag. *On the statistical analysis of dirty pictures*. Journal Of The Royal Statistical Society, vol. B-48, pp. 259-302, 1986.
- [8] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986.
- [9] P. Brodatz. *A Photographic Album for Artists and Designers*. Dover, New York, 1966.