

Synthèse de Texture Contrainte par Champ de Structure Arbitraire

ADIB AKL^{1,2}, CHARLES YACCOUB², MARC DONIAS¹, JEAN-PIERRE DA COSTA¹, CHRISTIAN GERMAIN¹

¹ Laboratoire d'Intégration du Matériau au Système (IMS)
351 Cours de la libération, UMR CNRS 5218, 33405 Talence Cedex, France

² Faculté d'Ingénierie, Université Saint-Esprit de Kaslik (USEK)
Kaslik, BP 446, Jounieh, Liban

¹marc.donias@ims-bordeaux.fr, jean-pierre.dacosta@ims-bordeaux.fr, christian.germain@ims-bordeaux.fr
²adibakl@usek.edu.lb, charlesyacoub@usek.edu.lb

Résumé – Ce papier propose un algorithme de synthèse en deux étapes permettant de générer des textures de formes arbitraires à partir d'un exemplaire d'entrée. Dans la première étape, un champ de tenseurs de structure est construit à partir d'une image arbitraire dite *texture de contrainte* et utilisé dans la seconde étape pour contraindre la synthèse de la texture. Afin d'évaluer si la méthode reproduit fidèlement l'aspect visuel de l'exemplaire, plusieurs tests sont effectués avec différentes textures de contrainte. Les résultats obtenus montrent la capacité de l'algorithme proposé à synthétiser des textures de formes arbitraires tout en respectant les caractéristiques locales de l'exemplaire.

Abstract – This paper proposes a two-stage synthesis algorithm which allows the synthesis of arbitrary-shaped textures. In the first stage, a structure tensor field is constructed using an arbitrary image and used in the second stage to constrain the synthesis of the texture. To ensure that the method reproduces as faithfully as possible the visual aspect and statistics of the input sample, several tests using various constraint images are implemented. The obtained results highlight the capability of the proposed algorithm to synthesize arbitrary-shaped textures while respecting the local characteristics of the exemplar.

1 Introduction

La synthèse de textures joue un rôle important en traitement d'images, en infographie, dans différents domaines d'applications tels que l'extrapolation d'images, la synthèse de vidéos ou la simulation de matériaux virtuels [1]. La diversité des textures [2] a conduit au développement de nombreux algorithmes. Par exemple, Paget et Longstaff [3] proposent une méthode de modélisation de texture par champ de Markov basée sur l'estimation non-paramétrique de la fonction de densité de probabilité conditionnelle locale. Wei et Levoy [4] proposent un algorithme non paramétrique qui s'appuie sur les hypothèses d'un champ de Markov aléatoire. La méthode modélise la texture en tant que réalisation d'un processus aléatoire local et stationnaire. La texture de sortie est synthétisée pixel par pixel selon un balayage lexicographique. L'algorithme à base de patch de Efros et Leung [5] génère une texture en échantillonnant des valeurs dans l'échantillon d'entrée. Tartavel et al. [6] proposent un algorithme qui impose des statistiques d'ordre 1 à des éléments d'un dictionnaire adaptatif, ainsi que des contraintes d'ordre 2 sur les valeurs des pixels. La structure synthétique résultante contraint ensuite la synthèse de la texture elle-même.

Récemment présentée dans [8], une approche de synthèse en deux étapes consiste à construire une couche dite *de structure* à partir de l'analyse de l'exemplaire, puis à l'utiliser pour contraindre la synthèse de la texture elle-même. Dans cette méthode, la couche de structure est représentée par un champ de

tenseurs de structure qui exprime les orientations dominantes et le degré d'anisotropie locale dans la texture. S'appuyant sur cette dernière méthode, cet article propose un algorithme de synthèse de textures de formes arbitraires qui procède en trois étapes : (i) un champ de tenseurs *contrainte* est créé à partir des orientations d'une texture de base, choisie arbitrairement ; (ii) la texture initiale est utilisée pour construire un dictionnaire d'intensités et de tenseurs ; (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.

L'article est organisé comme suit : la section 2 détaille la création du champ de tenseurs *contrainte* à partir d'une texture arbitrairement imposée et la construction du dictionnaire de recherche. L'algorithme effectuant la synthèse contrainte par le champ de tenseurs obtenu est présenté dans la section 3. Finalement, des résultats sont présentés et discutés dans la section 4.

2 Construction de la contrainte et du dictionnaire

2.1 Instauration du champ de tenseur contrainte

La première étape de l'algorithme de synthèse proposé consiste à construire un champ de tenseurs de structure S_{con} à partir d'une texture arbitraire imposée I_{con} . Cette étape comporte trois parties :

- calcul du champ de tenseurs de structure S_{con} de I_{con} et du champ d'orientation O_{con} associé.

L'orientation d'un tenseur S est donnée par :

$$O(S) = \tan^{-1}(e_y / e_x), \quad (1)$$

où $[e_x, e_y]$ est un vecteur propre de S associé à la plus grande valeur propre.

- calcul du champ de tenseurs S_{en} de la texture d'entrée I_{en} .
- modulation des orientations de S_{en} par O_{con} : à chaque position (x, y) du champ S_{en} , application d'une rotation du tenseur $S_{en}(x, y)$ d'un angle $O_{con}(x, y)$, i.e. l'angle à la position correspondante dans le champ O_{con} :

$$S_{con}(x, y) \leftarrow R(x, y) * S_{en}(x, y) * R(x, y)^T, \quad (2)$$

où $*$ est la multiplication matricielle, T désigne l'opérateur transposée et R la matrice de rotation :

$$\begin{bmatrix} \cos\{O_{con}(x, y)\} & \sin\{O_{con}(x, y)\} \\ -\sin\{O_{con}(x, y)\} & \cos\{O_{con}(x, y)\} \end{bmatrix} \quad (3)$$

Cette méthode permet d'assurer que les tenseurs du champ instauré S_{con} sont des versions « tournées » des tenseurs du champ d'entrée S_{en} pour représenter la structure de la texture imposée I_{con} .

2.2 Construction du dictionnaire de recherche

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

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

$$V_k = VI_k \cup VS_k, \quad (4)$$

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

Afin de garantir que les différentes structures de la carte de tenseurs imposée S_{con} existent dans le dictionnaire, ce dernier est remplacé par un dictionnaire $D_M(I_{en})$ incluant $D(I_{en})$. $D_M(I_{en})$ est constitué de voisinages extraits d'images $\{Rot_m(I_{en}), m = 0, \dots, M-1\}$ où $Rot_m(I_{en})$ est la rotation de I_{en} d'un angle $\varphi_m = 2\pi m/M$ ainsi que des champs de tenseur de structure correspondants.

La première ligne de la figure 1 illustre un exemple de création de deux versions d'une même texture tournée

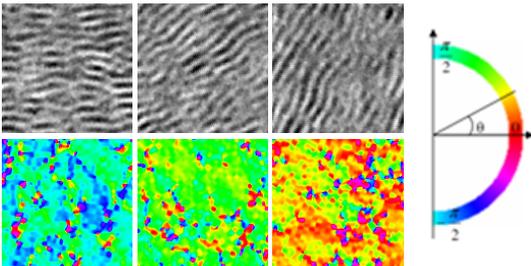


Figure 1 : Exemple de synthèse de textures et de champ de tenseurs avec différents angles de rotation. 1^{ère} ligne (gauche à droite) : l'exemplaire, la version tournée de 30° et celle tournée de 60°. 2^{ème} ligne : images d'orientation du champ de tenseur correspondant aux textures de la 1^{ère} ligne. La dernière colonne présente la palette utilisée pour les images d'orientation.

de $\pi/6$ et $\pi/3$. La 2^{ème} ligne présente les images d'orientation des champs de tenseurs correspondants. Ces images sont à la base de la création du dictionnaire utilisé dans le processus de synthèse. La dernière colonne présente la palette utilisée pour les images d'orientation. Il est à noter que cette palette est également utilisée pour tous les résultats de la section 4.

3 Le processus de synthèse

Le champ de tenseurs *contrainte* S_{con} et le dictionnaire $D_M(I_{en})$ étant créés, la texture de sortie I_{sor} est initialisée aléatoirement (par des valeurs choisies dans la texture d'entrée I_{en}) et modifiée durant la synthèse comme suit : pour chaque position $p=(x, y)$ dans I_{sor} , le voisinage $V_p = VI_p \cup VS_p$ (VI_p est le voisinage d'intensité à la position p dans I_{sor} et VS_p le voisinage de tenseurs de structure à la position correspondante dans S_{con}) est extrait. La meilleure ressemblance est identifiée dans $D_M(I_{en})$ et le pixel correspondant est copié à la position cible $p=(x, y)$ de I_{sor} .

La métrique utilisée pour le calcul de la ressemblance des voisinages est la somme des erreurs quadratiques (*SSD*) pour les voisinages d'intensité combinée à la somme des valeurs de dissimilarités de tenseurs (*STD*) définie dans (6) pour les voisinages tensoriels :

$$R(k, p) = W \cdot SSD(VI_k, VI_p) + (1 - W) \cdot STD(VS_k, VS_p), \quad (5)$$

où VI_k est la composante du $k^{\text{ème}}$ voisinage d'intensité dans $D_M(I_{en})$, VI_p est le voisinage d'intensité à la position p dans I_{sor} , VS_k est la composante du $k^{\text{ème}}$ voisinage de tenseurs dans $D_M(I_{en})$ et VS_p est le voisinage de tenseurs à la position p dans S_{con} . W est un facteur de pondération tandis que la dissimilarité s'exprime par :

$$STD(VS^1, VS^2) = \sum_{i=1}^{N_i} Dist(VS_i^1, VS_i^2), \quad (6)$$

où N_i est le nombre de tenseurs dans chaque voisinage, VS_i représente le $i^{\text{ème}}$ tenseur du voisinage et $Dist$ est l'une des quatre métriques de ressemblance de tenseurs proposées dans [8]. Dans ce papier, la métrique Euclidienne est utilisée. Un voisinage causal associé à un parcours lexicographique, aussi bien qu'un voisinage carré avec un parcours aléatoire, peuvent être utilisés.

Le pseudocode de l'algorithme de synthèse tout entier est fourni dans la figure 2.

Dans le cas de textures contenant des structures à grande échelle, l'utilisation de larges voisinages est coûteuse en termes de complexité calculatoire. Aussi, des pyramides Gaussiennes [8] des images et des champs de tenseurs sont utilisées pour retranscrire les structures à différentes échelles. Le dictionnaire $D_M(I_{en})$ est ainsi construit à l'aide de voisinages multi-échelles extraits dans les pyramides d'intensité et de tenseurs. Les pyramides sont 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_{sor} et S_{con} , respectivement. La synthèse commence au niveau de la

pyramide le plus élevé jusqu'au niveau le plus bas. Plus précisément, pour s'assurer que les détails à haute fréquence ajoutés sont compatibles avec les structures basses fréquences déjà synthétisés, le voisinage multi-résolution à la position $p=(x, y)$ au niveau L contient son voisinage au même niveau ainsi que le voisinage de la position correspondante $p'=(x/2, y/2)$ au niveau $L+1$ précédemment synthétisé [4].

Entrée: I_{en} & I_{con}

$S_{con} \leftarrow \text{CREATIONCONTRAINTE}(I_{en} \& I_{con})$

$D_M(I_{en}) \leftarrow \text{CREATIONDICTIONNAIRE}(I_{en})$

$I_{sor} \leftarrow \text{INITIALISATIONBRUIT}(I_{en})$

Pour chaque position $p=(x, y)$ dans I_{sor}

$PIX_k \leftarrow \text{argmax}\{\text{MEILLEURRESSEMBLANCE}(V_p \text{ VS } V_k)\}$

$I_{sor}(x, y) \leftarrow PIX_k$

Fin pour

Sortie: I_{sor}

Figure 2 : Pseudocode de l'algorithme proposé.

4 Résultats

Pour des raisons de format, nous montrons dans cette partie un sous-ensemble de résultats obtenus par l'algorithme proposé. La figure 3 présente les entrées de l'algorithme de synthèse. La 1^{ère} ligne montre un échantillon de matériau composite anisotrope (I_{en}) et son champ de tenseurs de structure (S_{en}) représenté par son image d'orientation. Les 2^{ème} et 3^{ème} lignes montrent les textures arbitrairement imposées (texture en forme de vagues I_{con_1} et texture concentrique I_{con_2}). La figure 4 présente les résultats obtenus avec les deux textures arbitraires de la figure 3. La 1^{ère} colonne montre les champs d'orientations (O_{con_1} et O_{con_2}) des textures arbitraires (I_{con_1}) et (I_{con_2}) de la figure 3, respectivement et les images d'orientation des champs de tenseurs *contrainte* (S_{con_1}) et (S_{con_2}), obtenus par la méthode proposée en utilisant la même texture d'entrée (I_{en}). La 2^{ème} colonne montre les textures synthétisées ($I_{sor_1/O}$) et ($I_{sor_2/O}$) en utilisant respectivement les champs de tenseurs (O_{con_1}) et (O_{con_2}) pour contraindre la synthèse. La 3^{ème} colonne montre les textures synthétisées ($I_{sor_1/S}$) et ($I_{sor_2/S}$) en utilisant respectivement les champs de tenseurs *contrainte* (S_{con_1}) et (S_{con_2}). Tous les résultats de synthèse sont obtenus à partir du même exemplaire (I_{en}) avec un pas angulaire de 5°. Une pyramide Gaussienne à deux échelles est utilisée dans les résultats ($I_{sor_1/O}$ et $I_{sor_1/S}$) tandis que les résultats ($I_{sor_2/O}$ et $I_{sor_2/S}$) sont obtenus avec l'approche mono-échelle. Tous les résultats présentés sont obtenus après deux itérations en utilisant un voisinage causal et un parcours lexicographique. On peut remarquer d'après les figures 3 et 4 que la méthode proposée 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.

Par conséquent, l'algorithme de synthèse contrainte par le champ de structure obtenu est capable de synthétiser

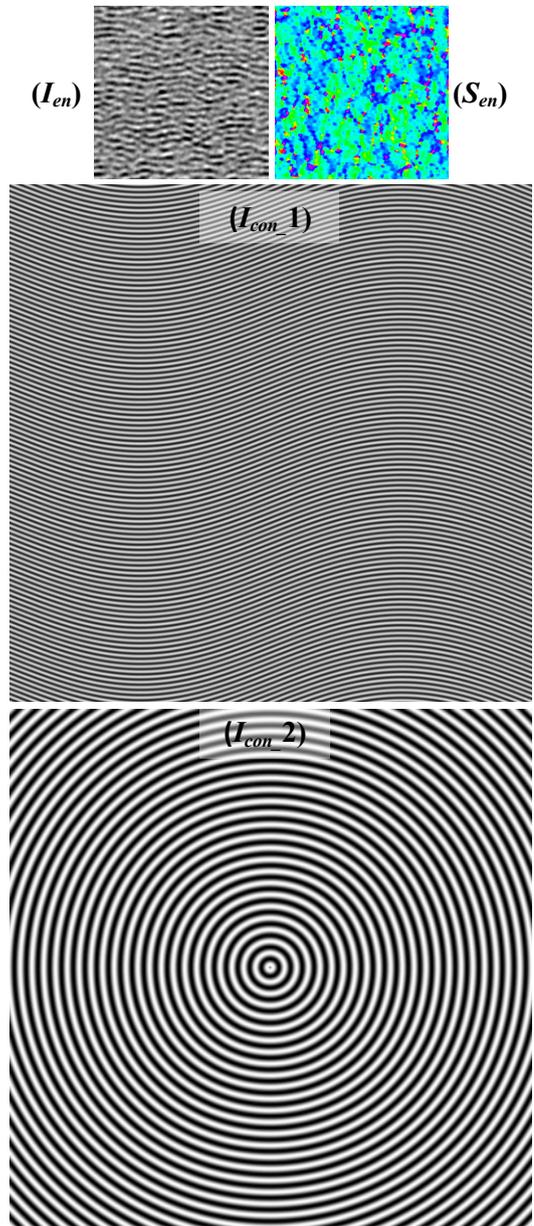


Figure 3 : Les entrées à l'algorithme de synthèse. 1^{ère} ligne : texture d'entrée (I_{en}) et l'orientation de son champ de tenseurs (S_{en}). 2^{ème} et 3^{ème} lignes : textures arbitrairement imposées (I_{con_1}) et (I_{con_2}).

des textures restructurées, et de formes arbitraires en les habillant par les motifs de l'exemplaire, tout en respectant ses dynamiques et ses aspects visuels. En revanche la synthèse contrainte par les champs de tenseurs des textures arbitraires conduit à des textures ayant des motifs trop réguliers et qui n'existent pas dans l'exemplaire. Cela justifie l'importance de l'algorithme d'instauration du champ de tenseur *contrainte*.

Du fait que seule l'orientation est modulée, l'approche proposée 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. On propose donc comme perspective, une solution qui consiste à limiter la recherche de la ressemblance dans le dictionnaire, à l'image tournée selon le même angle que celui imposé par la contrainte.

Par ailleurs, il est important de noter que l'algorithme proposé part du principe qu'on dispose d'une texture d'entrée de même taille que la contrainte arbitraire. Le

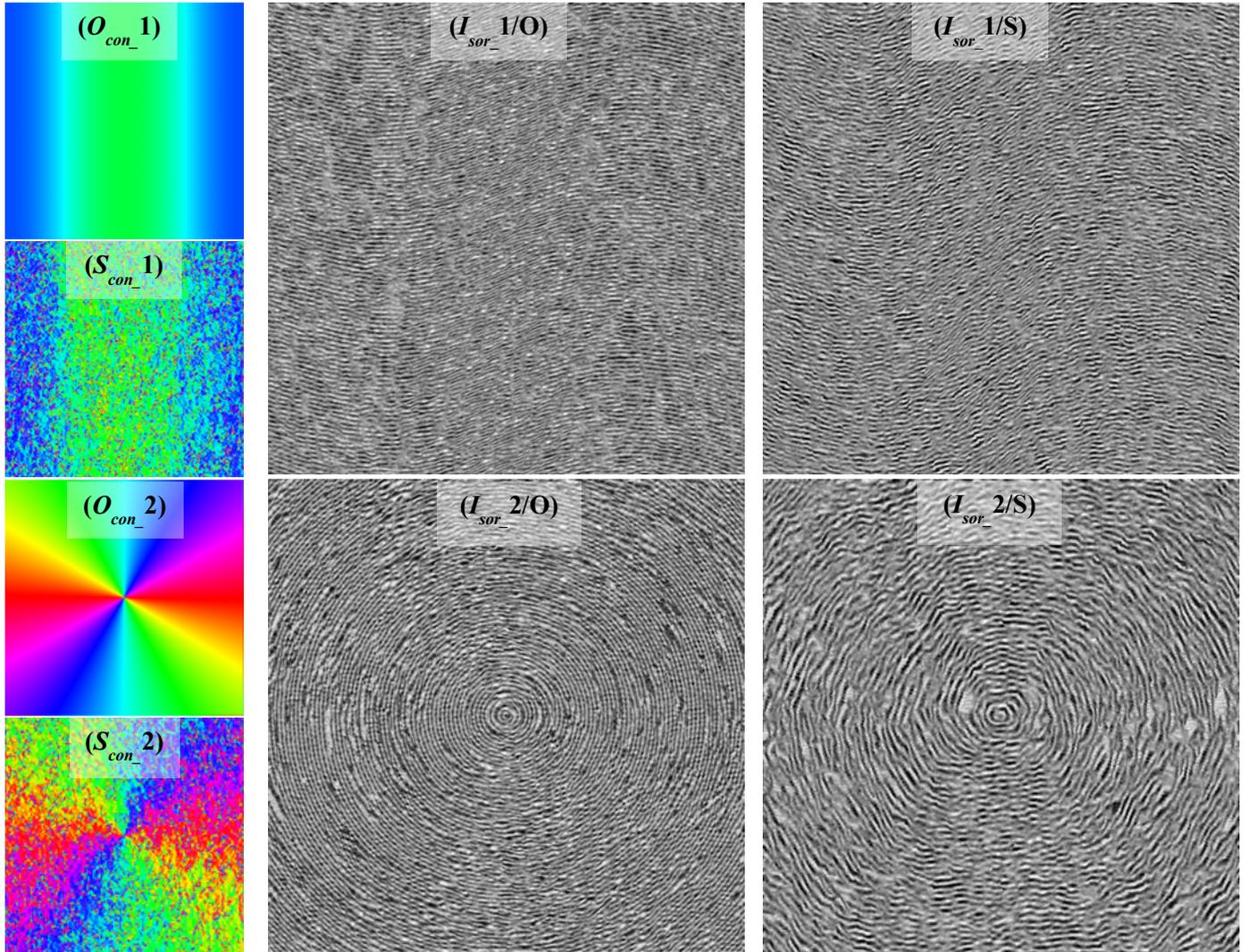


Figure 4 : Résultats de synthèse. 1^{ère} colonne : champs d'orientations (O_{con_1}) et (O_{con_2}) des textures arbitraires (I_{con_1}) et (I_{con_2}) de la figure 3, respectivement, et les images d'orientation des champs de tenseurs *contrainte* (S_{con_1}) et (S_{con_2}), obtenus par la méthode proposée en utilisant la même texture d'entrée (I_{en}). 2^{ème} colonne : textures synthétisées ($I_{sor_1/O}$) et ($I_{sor_2/O}$) en utilisant respectivement les champs de tenseurs (O_{con_1}) et (O_{con_2}) pour contraindre la synthèse. 3^{ème} colonne : textures synthétisées ($I_{sor_1/S}$) et ($I_{sor_2/S}$) en utilisant respectivement les champs de tenseurs *contrainte* (S_{con_1}) et (S_{con_2}). Tous les résultats de synthèse sont obtenus à partir de (I_{en}).

cas échéant, une alternative consiste à synthétiser à partir de l'exemplaire (généralement de petite taille), une texture de taille égale à celle de la texture imposée. La texture résultante serait donc à la base de la construction du champ de tenseurs de structure utilisé comme contrainte dans la synthèse de la texture.

5 Conclusion

Cet article présente un algorithme de synthèse de textures de formes arbitraires où le champ de tenseurs de l'exemplaire est modulé par l'orientation d'une texture imposée et utilisé comme contrainte dans la synthèse de la texture. Les résultats montrent que la méthode est capable de synthétiser des textures de formes spécifiées, tout en préservant l'essentiel des caractéristiques locales de l'exemplaire.

Références

- [1] 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.
- [2] D. A. Forsyth and J. Ponce, "Texture," in *Computer Vision, A Modern Approach*, Pearson Education, 2003.
- [3] R. Paget and I.D. Longstaff, "Texture synthesis via a non-causal nonparametric multiscale markov random field". *IEEE Trans. Image Processing*, vol. 7, no.6, pp. 925-931, 1998.
- [4] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," *Proc. ACM SIGGRAPH 2000*, pp. 479-488, 2000.
- [5] 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.
- [6] 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/>.
- [7] G. Peyré, "Texture Synthesis with grouplets". *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 733-746, 2009.
- [8] A. Akl, C. Yaacoub, M. Donias, J.P. Da Costa, C. Germain, "Structure tensor based synthesis of directional textures for virtual material design", *Proc. of the 21st IEEE International Conference on Image Processing (ICIP)*, Paris, 2014.