



HAL
open science

Synthèse de textures par transport optimal

Rabin Julien, Gabriel Peyré

► **To cite this version:**

Rabin Julien, Gabriel Peyré. Synthèse de textures par transport optimal. GRETSI'11, 2011, France. hal-00744812

HAL Id: hal-00744812

<https://hal.science/hal-00744812>

Submitted on 24 Oct 2012

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.

Synthèse de textures par transport optimal

Julien RABIN¹ et Gabriel PEYRÉ²

¹CMLA, ENS de Cachan – 61 Avenue du Président Wilson, 94230 Cachan Cedex 00, France

²CEREMADE, Univ. Paris Dauphine – Place du Maréchal De Lattre De Tassigny, 75775 Paris cedex 16, France
gabriel.peyre@ceremade.dauphine.fr

Résumé – Cet article présente un formalisme générique pour la synthèse de textures couleurs par méthodes statistiques. L’algorithme développé utilise le transport optimal pour prendre en compte des contraintes statistiques apprises sur une texture donnée en exemple. L’approche proposée généralise des méthodes classiques de synthèse de textures par égalisation d’histogrammes multi-échelles. Elle permet, entre autres, de traiter les images couleurs. Nous verrons que l’utilisation de contraintes statistiques en dimension arbitraire rend possible la synthèse de textures couleurs complexes. Ceci permet également de réaliser un mélange de textures par calcul de barycentres de distributions.

Abstract – This article introduces a generic methodology for the synthesis of color texture using statistical constraints. The proposed algorithm makes use of optimal transport to take into account constraints learned from a given exemplar. This approach generalizes several classical method for statistical texture synthesis using histograms of multiscale coefficients. It allows in particular to take into account color textures in a seamless way. The extraction of high dimensional co-occurrence histograms allows one to synthesis complicated color textures. This framework also enables texture mixing by the computation of the barycenter of distributions.

1 Introduction

Synthèse par l’exemple. La synthèse de textures par l’exemple demande la génération d’une nouvelle image qui doit être visuellement similaire à une texture de départ, tout en présentant un caractère aléatoire. Les méthodes les plus efficaces en infographie sont les méthodes par recopie, voir par exemple [4, 9]. Ces méthodes se contentent de replacer de façon aléatoire les motifs de la texture d’origine, en garantissant une juxtaposition cohérente afin de générer une image plausible. Elles sont peu contraintes, et ne permettent pas d’effectuer de réelles modifications sur l’image. Les méthodes procédurales génèrent des textures simples à partir de bruits filtrés qui sont modifiés par des opérateurs non-linéaires, voir par exemple [14]. Il est difficile de reproduire des motifs textures complexes à l’aide de ces approches, qui nécessitent de plus un ajustement manuel des paramètres. Les méthodes statistiques quant à elles modélisent les textures à synthétiser par des contraintes statistiques, comme par exemple la distribution de coefficients multi-échelles [6, 21, 2, 13]. Il est possible d’établir des modèles plus complexes en prenant en compte la distribution de paires de coefficients, voir par exemple [16].

Modification de textures. Heeger et Bergen [6] et Bar-Joseph *et al.* [1] effectuent un mélange de textures en combinant des coefficients d’ondelettes. Le mélange des statistiques de coefficients dans une trame de grouplets permet de mélanger des texture turbulentes [15]. Les méthodes par recopie permettent d’effectuer des mélanges de textures à l’aide de patches (sous-images) extraits d’un ensemble d’images d’apprentissage, ce qui crée des textures non-homogènes, voir par exemple [7, 5,

8]. Les méthodes de métamorphose de textures [10, 19, 18] effectuent un mélange en trouvant des correspondances entre des éléments de l’image (les textons). Ces méthodes sont étendues par Matusik *et al.* [11] afin de calculer des combinaisons convexes de textures en déformant les patches et en effectuant un moyennage d’histogrammes 1D.

Contribution. Cet article décrit un cadre générique pour les méthodes statistiques de synthèse de textures dans des dictionnaires (comme par exemple des bases multi-échelles). La méthode ainsi obtenue est plus générale car elle permet de prendre en compte des contraintes en dimension quelconque (comme par exemple des coefficients d’images couleurs, ou des paires de coefficients). Elle permet aussi de définir de façon naturelle des moyennes de textures par un barycentre au sens du transport optimal [20].

2 Synthèse par Projections Alternées

2.1 Architecture de l’algorithme de synthèse

Une architecture générique de synthèse de textures considère un ensemble de contraintes $\{\mathcal{C}_i\}_{i \in I}$, qui sont des sous-ensembles compacts $\mathcal{C}_i \subset \mathbb{R}^{N \times d}$, où N est le nombre de pixels, et $d = 1$ pour les images en niveaux de gris, $d = 3$ pour les images couleurs. L’ensemble des textures admissibles est l’intersection des contraintes

$$\mathcal{T} = \bigcap_{i \in I} \mathcal{C}_i.$$

Les contraintes utilisées portent typiquement sur les niveaux de gris de l’image (moyenne, variance, histogramme), sur la

régularité de l'image (modules de Fourier) ou sur la distribution des réponses à des bancs de filtres multi-échelles [6, 12, 16].

Cet ensemble \mathcal{T} est compact, et, suivant [21], on définit une synthèse comme une réalisation tirée selon la distribution uniforme sur \mathcal{T} , qui est la distribution ayant la plus forte entropie. Générer un tel tirage est très coûteux, car les contraintes \mathcal{C}_i sont en général complexes et non-convexes [21]. Portilla et Simoncelli [16] proposent une méthode de synthèse plus rapide qui calcule des réalisations selon une distribution non-uniforme mais de grande entropie. En partant d'une image $f^{(0)}$ qui est la réalisation d'un bruit blanc (ayant une forte entropie), la synthèse est obtenue par des projections alternées sur chaque contrainte,

$$f^{(\ell+1)} = \text{Proj}_{\mathcal{C}_{i(\ell)}}(f^{(\ell)}), \quad (1)$$

où $i(\ell) \in I$ correspond à un choix de contrainte à l'itération ℓ (un choix usuel consiste à parcourir périodiquement l'ensemble I). Bien que difficile à analyser théoriquement, on constate que pour les contraintes utilisées couramment en synthèse d'images, le processus converge vers une texture $f^{(\infty)} \in \mathcal{T}$, qui correspond à l'image synthétisée.

2.2 Contraintes statistiques

Vecteurs de coefficients. Dans la suite, nous allons considérer des vecteurs de N coefficients $x = (x_m)_{m=0}^{N-1} \in \mathbb{R}^{d \times N}$ où chaque x_m est un vecteur de \mathbb{R}^d . Ces coefficients peuvent être par exemple les valeurs des pixels d'une image en niveaux de gris ($d = 1$), d'une image couleurs ($d = 3$), ou encore des coefficients d'ondelettes.

Distributions discrètes. Nous allons également considérer des distributions discrètes $Y \subset \mathbb{R}^d$, qui sont des nuages de points $Y = \{y_m\}_{m=0}^{N-1}$. Une distribution est par définition un ensemble non ordonné de N points dans \mathbb{R}^d , qu'il ne faut pas confondre avec un vecteur ordonné de N coefficients $x \in \mathbb{R}^{d \times N}$. On passe d'un vecteur x à sa distribution $[x]$ associé en oubliant l'ordonnancement des points, c'est-à-dire $[x] = \{x_m\}_{0 \leq m \leq N-1}$.

Dictionnaires et décomposition. On note $f = (f^1, \dots, f^d) \in \mathbb{R}^{N \times d}$ une image de N pixels $f_m = (f_m^1, \dots, f_m^d) \in \mathbb{R}^d$ ($d = 1$ pour les images en niveaux de gris, $d = 3$ pour les images couleurs). On va également considérer des collections d'atomes orthogonaux dans \mathbb{R}^N , que l'on note $D = (\psi_m)_{m=0}^{P-1}$, où $\psi_m \in \mathbb{R}^N$ et $P \leq N$ est le nombre d'atomes. Un choix classique consiste à choisir pour D un ensemble d'ondelettes à une échelle et une orientation donnée, afin de contraindre les singularités géométriques de la texture [6, 12, 16]. Le coefficient de f associé à ψ_m est un vecteur $\langle f, \psi_m \rangle = (\langle f^r, \psi_m \rangle)_{r=0}^{d-1} \in \mathbb{R}^d$ obtenu en projetant chaque composante $f^r \in \mathbb{R}^N$ de f sur ψ_m . On note alors $Df = (\langle f, \psi_m \rangle)_m \in \mathbb{R}^{P \times d}$ la collection de tous les P coefficients.

Contrainte statistique. Une contrainte statistique définit l'ensemble des vecteurs ayant pour distribution Y , et s'écrit

$$\mathcal{C}_Y = \{x \in \mathbb{R}^N \mid [x] = Y\},$$

où $[x] = \{x_m\}_{0 \leq m \leq N-1}$ correspond au nuage de point non ordonné défini par les coefficients de x . Ainsi, on a $x \in \mathcal{C}_Y$ s'il existe une permutation $\sigma \in \Sigma_N$ des indices $\{0, \dots, N-1\}$ tel que $x_m = y_{\sigma(m)}$.

Contrainte sur les pixels et dans un dictionnaire. Il est possible de contraindre les valeurs de ses pixels en imposant que $f \in \mathcal{C}_Y$ pour une certaine distribution Y . On peut aussi souhaiter contraindre la distribution de ses coefficients $Df \in \mathbb{R}^{P \times d}$ dans un dictionnaire D . Une telle contrainte s'écrit $Df \in \mathcal{C}_Y$, ou de façon équivalente,

$$f \in \mathcal{C}_{Y,D} \quad \text{où} \quad \mathcal{C}_{Y,D} = \{f \in \mathbb{R}^N \mid [Df] = Y\}. \quad (2)$$

2.3 Projection sur des contraintes statistiques

Afin d'appliquer l'algorithme de synthèse (1) à des contraintes statistiques, il est nécessaire de pouvoir calculer la projection sur une contrainte $\mathcal{C}_{Y,D}$.

Projection sur une contrainte statistique. À partir d'un vecteur de coefficients x quelconque, on calcule $\tilde{x} = \text{Proj}_{\mathcal{C}_Y}(x)$ sa projection orthogonale sur \mathcal{C}_Y . Calculer sa projection correspond à calculer le transport optimal (également appelé affectation optimale [3]) $\sigma^* \in \Sigma_N$ entre les nuages $[x]$ et Y , c'est-à-dire $\tilde{x}_m = y_{\sigma^*(m)}$ avec

$$\sigma^* = \underset{\sigma \in \Sigma_N}{\text{argmin}} W_\sigma([x], Y)^2 = \sum_m \|x_m - y_{\sigma(m)}\|^2. \quad (3)$$

La distance $W([x], Y) = W_{\sigma^*}([x], Y)$ est appelé distance de Wasserstein [20], c'est une distance sur l'espace des distributions discrètes.

Projection sur une contrainte de dictionnaire. Étant donné un dictionnaire, on calcule la projection $\tilde{f} = \text{Proj}_{\mathcal{C}_{Y,D}}(f)$ de f sur une contrainte statistique dans le dictionnaire $\mathcal{C}_{Y,D}$ en calculant les coefficients dans le dictionnaire $x = Df$, puis

$$\tilde{f} = \text{Proj}_{D^\perp}(f) + \sum_m \tilde{x}_m \psi_m \quad \text{où} \quad \tilde{x} = \text{Proj}_{\mathcal{C}_Y}(x), \quad (4)$$

où $\text{Proj}_{D^\perp}(f)$ est la projection orthogonale de f sur le complémentaire orthogonal de $\text{Vect}(\psi_m)_m$, l'espace engendré par le dictionnaire (dans le cas où D est une base orthogonale, cette contribution disparaît). La formule (4) signifie que la projection sur $\mathcal{C}_{Y,D}$ se calcule en projetant les coefficients de f dans D sur \mathcal{C}_Y puis en reconstruisant l'image.

Pour les applications numériques à la synthèse de textures, nous avons utilisé un dictionnaire D redondant d'ondelettes orientées [16]. Un tel dictionnaire améliore la qualité des images par rapport à un dictionnaire orthogonal, mais dans ce cas la formule de projection (4) n'est qu'une approximation, ce qui ne pose cependant pas de problème en pratique.

Accélération du calcul de la projection. Le calcul du transport optimal σ^* dans (3) peut s’effectuer en utilisant des algorithmes dédiés d’affectation optimale, qui nécessitent un nombre d’opérations de l’ordre de $O(N^{5/2} \log(N))$, voir [3]. Ce coût est malheureusement prohibitif pour des applications en image, où N peut être très grand (nombre de pixels dans l’image). Afin d’obtenir un algorithme rapide, nous utilisons une méthode de calcul d’affectation approchée rapide, développée récemment dans [17].

3 Synthèse multi-échelles de textures couleurs

3.1 Algorithme de synthèse par projection

Algorithme. L’algorithme exploite un ensemble $\{D_i\}_{i \in I}$ de dictionnaires et un ensemble de distributions $\{Y_i\}_{i \in I}$. La première phase de l’algorithme apprend les distributions à partir d’une texture f donnée en exemple, en calculant pour chaque $i \in I$ les distributions de ses coefficients, $Y_i = [D_i f]$. La deuxième phase de l’algorithme calcule les projections alternées (1) avec $C_i = C_{Y_i, D_i}$.

Dictionnaires multi-échelles. Pour les applications numériques, on choisit $D_1 = (\psi_m)_m$ comme l’identité $\langle f, \psi_m \rangle = f_m$, de telle sorte que $C_{Y_1, D_1} = C_{Y_1}$ contraint la distribution de valeurs (niveaux de gris ou couleurs) des pixels de l’image. Chaque dictionnaire D_i pour $i > 1$ est extrait d’une trame d’ondelettes orientées [16]. Un dictionnaire $D_i = (\psi_m)_m$ est formé d’un ensemble d’atomes à une échelle $j = j(i)$ et une orientation $\theta = \theta(i)$ fixées, c.a.d. $\psi_m(x) \approx \psi(2^j R_\theta(x - m))$, où ψ est une ondelette mère, R_θ est la rotation d’angle θ et m indexe la position de l’atome ψ_m dans l’image.

3.2 Moyenne de textures

On considère désormais plusieurs textures $\{f^{[s]}\}_{s \in S}$ données en exemples. Pour commencer, les distributions $Y_i^{[s]} = [D_i f^{[s]}]$ des coefficients de $f^{[s]}$ dans le dictionnaire D_i sont apprises. On définit ensuite un modèle de textures $\{Y_i\}_{i \in I}$ qui est une moyenne pondérée des modèles, en utilisant un barycentre au sens du transport optimal des distributions

$$\forall i \in I, \quad Y_i = \operatorname{argmin}_Y \sum_{s \in S} \rho_s W(Y_i^{[s]}, Y)^2 \quad (5)$$

où chaque $\rho_s \geq 0$ est un poids de barycentre (avec $\sum_s \rho_s = 1$). Ceci correspond à une généralisation du barycentre Euclidien, où l’on remplace la distance Euclidienne par la distance de Wasserstein. Le problème (5) nécessite la minimisation d’une fonctionnelle convexe dans l’espace des distributions, mais dont l’optimisation est difficile. Nous avons proposé dans [17] une méthode d’approximation qui minimise par descente de gradient une fonctionnelle non-convexe différentiable. Une fois le

modèle moyen $\{Y_i\}_{i \in I}$ appris, il est possible de synthétiser une texture moyenne en utilisant l’algorithme décrit à la section 3.1.

Un exemple de résultat de synthèse et de mélange de trois textures couleurs obtenu avec cet algorithme est présenté en figure 1.

3.3 Prise en compte des dépendances statistiques

Les coefficients multi-échelles d’images naturelles ne sont pas statistiquement indépendants à cause de la présence de structures géométriques, voir par exemple [16]. Il est nécessaire d’étendre le modèle précédent en prenant en compte les dépendances entre paires de coefficients voisins. On note $\gamma = (0, \gamma_1, \dots, \gamma_{K-1})$ un système de voisinage, où chaque déplacement γ_k permet de relier un index m à un index voisin $m + \gamma_k$ dans un dictionnaire $D = \{\psi_m\}_m$. On extrait les coefficients dans le voisinage γ pour obtenir un vecteur

$$f_m^\gamma = (\langle f, \psi_{m+\gamma_k} \rangle)_{k=0}^{K-1} \in \mathbb{R}^{dK}$$

en grande dimension pour chaque index m . La distribution des statistiques jointes dans le voisinage γ est alors calculée comme $D^\gamma f = (f_m^\gamma)_m \in \mathbb{R}^{P \times (Kd)}$.

L’algorithme de synthèse de textures réalise successivement les projections sur les contraintes

$$C_{i,j} = \{f \in \mathbb{R}^N \mid [D_i^{\gamma_j} f] = Y_{i,j}\},$$

où les $\{D_i\}_i$ sont des dictionnaires, $\{\gamma_j\}_j$ des systèmes de voisinages, et $Y_{i,j} = [D_i^{\gamma_j} f^{[0]}]$ des distributions apprises sur une texture $f^{[0]}$ donnée en exemple.

On peut également réaliser une interpolation entre des modèles $\{Y_{i,j}^{[s]} = [D_i^{\gamma_j} f^{[s]}]\}_{s \in S}$ appris à partir d’un ensemble de textures $\{f^{[s]}\}_{s \in S}$. La figure 2 montre un exemple d’interpolation à l’aide de systèmes de voisinages $\{\gamma_j\}_j$ groupant des coefficients d’ondelettes voisins spatialement à une échelle donnée, et des coefficients voisins à travers deux échelles consécutives.

4 Conclusion

Dans ce papier, nous avons introduit un nouveau cadre pour la définition de moyennes de distributions statistiques de coefficients en dimension quelconque. Nous avons démontré la validité de l’approche proposée par une application à la synthèse et au mélange de textures, qui illustre par ailleurs l’intérêt des méthodes de transport optimal pour la modélisation et la synthèse d’images naturelles.

Remerciements Ces travaux ont été partiellement financés par le Centre National d’Études Spatiales (projet MISS), the Office of Naval research (grant N00014-97-1-0839), the European Research Council (advanced grant “Twelve labours”), et l’Agence Nationale de la Recherche (projet ANR NatImage 08-EMER-009).

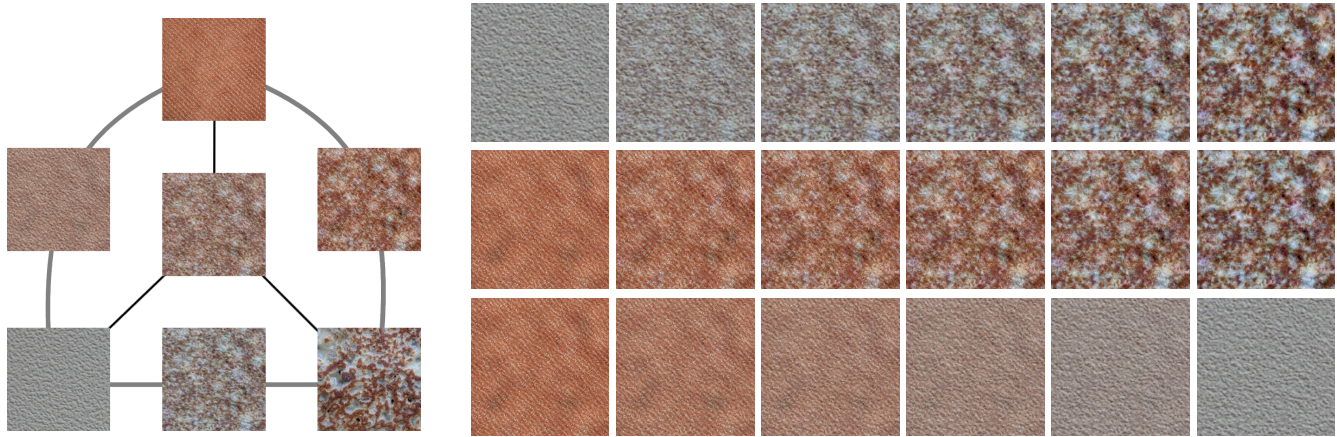


FIGURE 1 – Gauche : les trois textures $\{f^{[1]}, f^{[2]}, f^{[3]}\}$ données en exemple sont les sommets d’un triangle. Les textures au milieu des arêtes et au centre sont des textures moyennes calculées en faisant varier les poids (ρ_1, ρ_2, ρ_3) . Droite : interpolation entre deux textures le long d’une arête, obtenue en faisant varier continûment les poids.

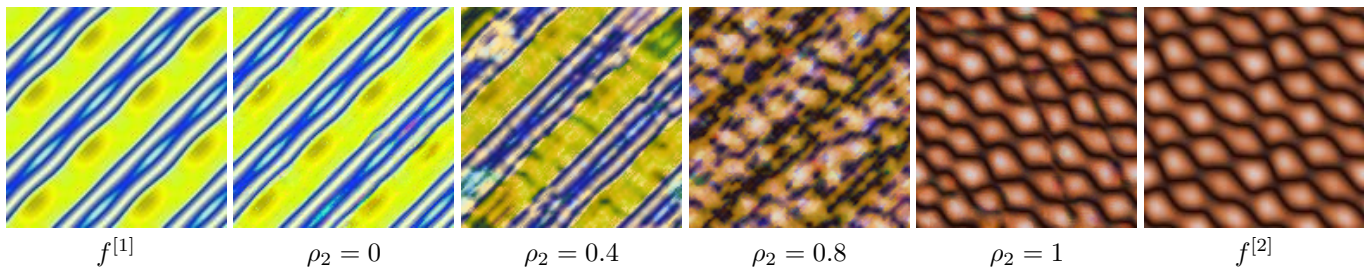


FIGURE 2 – Interpolation entre deux textures à l’aide d’histogramme de co-occurrence de coefficients d’ondelettes.

Références

- [1] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Tr. on Vis. and Comp. Graph.*, 7(2) :120–135, 2001.
- [2] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proc. Siggraph '97*, pages 361–368, 1997.
- [3] R. Burkard, M. Dell’Amico, and S. Martello. *Assignment Problems*. SIAM, 2009.
- [4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. of ICCV '99*, page 1033, 1999.
- [5] A.A. Efros and W.T. Freeman. Image quilting for texture synthesis and transfer. *ACM Trans. on Graphics*, pages 341–346, August 2001.
- [6] D. J. Heeger and J. R. Bergen. Pyramid-Based texture analysis/synthesis. In *Proc. Siggraph '95*, Annual Conference Series, pages 229–238. ACM SIGGRAPH, 1995.
- [7] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In ACM, editor, *Proc. Siggraph'01*, pages 327–340. ACM Press, 2001.
- [8] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures : Image and video synthesis using graph cuts. *ACM Trans. on Graphics*, 22(3) :277–286, July 2003.
- [9] S. Lefebvre and H. Hoppe. Appearance-space texture synthesis. *ACM Trans. on Graphics*, 25(3) :541–548, July 2006.
- [10] Z. Liu, C. Liu, H-Y. Shum, and Y. Yu. Pattern-based texture metamorphosis. In *Proc. Pacific Graphics'02*, pages 184–193. IEEE Computer Society, 2002.
- [11] W. Matusik, M. Zwicker, and F. Durand. Texture design using a simplicial complex of morphable textures. *ACM Trans. on Graphics*, 24(3) :787–794, 2005.
- [12] D. Mumford and B. Gidas. Stochastic models for generic images. *Q. Appl. Math.*, LIV(1) :85–111, 2001.
- [13] R. Paget and I. D. Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE Trans. Image Proc.*, 7(6) :925–931, June 1998.
- [14] K. Perlin. An image synthesizer. In *Proc. Siggraph '85*, pages 287–296, New York, NY, USA, 1985. ACM Press.
- [15] G. Peyré. Texture synthesis with grouplets. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 32(4) :733–746, 2010.
- [16] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. Journal of Computer Vision*, 40(1) :49–70, 2000.
- [17] J. Rabin, G. Peyré, J. Delon, and M. Berton. Wasserstein barycenter and its application to texture mixing. *Proc. SSVN'11*, 2011.
- [18] A. Tal and G. Elber. Image morphing with feature preserving texture. *Comput. Graph. Forum*, 18(3) :339–348, 1999.
- [19] L. Tonietto and M. Walter. Texture metamorphosis driven by texton masks. *Computers and Graphics*, 29(5) :697–703, 2005.
- [20] C. Villani. *Topics in Optimal Transportation*. American Math. Society, 2003.
- [21] S. C. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (FRAME) : Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2) :107–126, 1998.