



**HAL**  
open science

# Texture Synthesis and Modification with a Patch-Valued Wavelet Transform

Gabriel Peyré

► **To cite this version:**

Gabriel Peyré. Texture Synthesis and Modification with a Patch-Valued Wavelet Transform. Scale Space and Variational Methods in Computer Vision (SSVM'07), Jun 2007, Ischia, Italy. pp.640-651, 10.1007/978-3-540-72823-8 . hal-00365612

**HAL Id: hal-00365612**

**<https://hal.science/hal-00365612>**

Submitted on 3 Mar 2009

**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.

# Texture Synthesis and Modification with a Patch-valued Wavelet Transform

Gabriel Peyré

Ceremade, Université Paris Dauphine,  
Place du Marchal De Lattre De Tassigny,  
75775 Paris Cedex 16 France  
[gabriel.peyre@ceremade.dauphine.fr](mailto:gabriel.peyre@ceremade.dauphine.fr),  
<http://www.ceremade.dauphine.fr/~peyre/>

**Abstract.** This paper models a texture as a 2D mapping onto a nonlinear manifold representing the local structures of the image. This manifold is learned from the set of local patches from an exemplar texture. A multiscale decomposition of this manifold valued representation is computed that mimics the orthogonal wavelet transform. The key ingredient of this decomposition is a geometric association field that drives the computations along the manifold. Iterated predictions leads to the computation of details coefficients over the features manifold. The resulting transform is invertible, non-linear and represents efficiently the local geometric structures of the exemplar. The multiscale coefficients of this transform are used to perform analysis and synthesis of textures.

*Keywords:* Manifold of patches, non linear wavelet transform, image geometry, texture synthesis, texture modification.

## 1 Geometric Modeling of Images

*Multiscale decompositions.* The classical isotropic wavelet transform [10] cannot capture efficiently geometrical structures of textures. The local anisotropy of edges requires specific constructions such as the curvelets frame of Candès and Donoho [1] or the bandelets framework of Le Pennec and Mallat [6].

*Geometric structure propagation.* The local geometry of images can be described as points on a curved manifold. Lee et al. [7] show how the edge manifold emerges from the set of patches extracted from natural images. To handle more complex texture features we propose to learn this manifold from an exemplar image. Computations are performed along this manifold using a local connexion describing how structures propagate in the image plane. Mumford first introduced this notion for edges propagation with the elastica model [12]. Computer vision scientists such as Williams and Jacobs [17] proposed in some cases efficient approximations of this edge propagation field.

*Texture synthesis.* Pioneer work of Julesz [5] states filtering rules for the probabilistic characterization of textures. The wavelet domain modeling of Heeger and Bergen [4] synthesizes cloudy textures.

Most successful approaches for texture synthesis in graphics are based on non-parametric copy of small patches from an original texture, see for example Efros and Leung [3]. Recent approaches such as the method of Lefebvre and Hoppe in [8] have re-casted this non-parametric patches sampling into a multiscale framework. A random shuffling of patches coordinates allows to deviate from the original texture. These modifications can be seen roughly as wavelet details that are added at various scales during the synthesis.

Our manifold description of textures paves the way between multiscale texture descriptions [4] and more complex, non parametric sampling [3,8].

*Manifold valued analysis.* Modeling data using a manifold structure can be performed using non-linear estimation algorithms such as Isomap [15] or LLE [13]. These dimensionality reduction procedures enable the analysis of libraries of images such as the ones studied by Donoho and Grimes [2].

Processing of manifold valued functions is studied by Ur-Raman et al. [16] who propose a wavelet-like decomposition for such data. Our multiscale feature valued analysis presented in section 4 is inspired by this work but is adapted to geometric images and textures. In particular, our scheme takes into account the relative position of two features in the image, which enables the generation of anisotropic geometries such as elongated features.

Recent approaches to image synthesis in computed graphics use manifold modeling of textures. Matusik et al. define a manifold from a set of textures [11]. Lefebvre and Hoppe introduce a mapping of an image into a higher dimensional appearance space [9]. This embedding allows a synthesis with high fidelity and spacial variations.

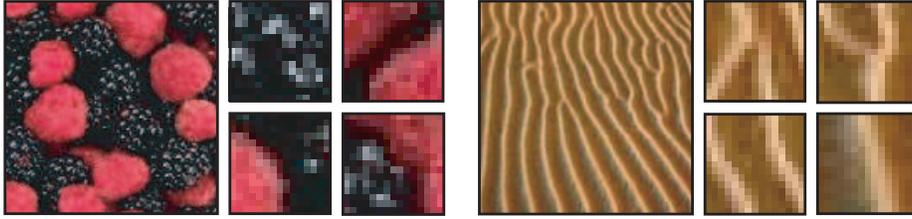
## 2 Manifolds as Image Models

We model the local structure of textures using a set of manifolds  $\{\mathcal{M}^w\}_w$  computed from an exemplar texture  $f_e$ . The manifold  $\mathcal{M}^w$  is composed of patches of fixed size  $w$  that are extracted from the original texture  $f_e$

$$\mathcal{M}^w \stackrel{\text{def.}}{=} \{p_t^w \mid t \in [0, 1]^2\} \subset L^2([-w/2, w/2]^2) \quad (1)$$

$$\text{where } \forall x \in [-w/2, w/2]^2, \quad p_t^w(x) = f_e(x - t). \quad (2)$$

As explained by Grimes and Donoho [2], the set  $\mathcal{M}^w$  might have a complex structure and be non-differentiable even for a simple images  $f_e$ . Figure 1 shows some examples of patches extracted from a discretized texture. In numerical applications, the image  $f_e$  is composed of  $n \times n$  pixels, so  $\mathcal{M}^w$  is a set of patches considered as vectors of  $\mathbb{R}^{m \times m}$  with  $m = wn$ .



**Fig. 1.** Two examples of textures and some local geometric structures.

*Models and estimation.* A gray scale image is a mapping  $f : \{0, \dots, n-1\}^2 \rightarrow \mathbb{R}$  from pixels to real numbers. Our manifold description of images corresponds to a factorization

$$f : \{0, \dots, n-1\}^2 \xrightarrow{\tilde{f}} \mathcal{M}^{w_0} \xrightarrow{\Phi_{w_0}} \mathbb{R} \quad (3)$$

where the manifold  $\mathcal{M}^{w_0}$  is estimated from an exemplar  $f_e$  (that might be different from  $f$ ) and the scale parameter  $w_0 > 0$  represents the smallest size of a typical feature. Note that the mapping  $\Phi_{w_0}$  depends only on  $\mathcal{M}^{w_0}$  and thus on  $f_e$  and can be computed independently of  $f$ . Factorization (3) extends to color images by considering each channel independently.

In this paper, we are interested by the mapping  $\tilde{f}$  which locates an image feature  $\tilde{f}(x) \in \mathcal{M}^{w_0}$  of size  $w_0$  at location  $x$  in the image plane. The function  $\Phi_{w_0}$  controls the display of each feature, and we assume a simple sampling process

$$\Phi_{w_0} : p \in \mathcal{M}^{w_0} \mapsto p(0) \in \mathbb{R}.$$

We note that recent works in harmonic analysis has focussed on the processing of such function  $\Phi_{w_0} \in L^2(\mathcal{M})$  defined over a manifold domain  $\mathcal{M}$ . In particular Fourier analysis of this function can be carried using eigenvector of the Laplace Beltrami operator. Szlam et al. [14] use this spectral decomposition to perform a non-linear filtering.

In this paper, instead of modifying  $\Phi_{w_0}$ , we propose to modify  $\tilde{f}$  which involves two key ingredients.

- Estimating  $\tilde{f}$  such that factorization (3) hold at least approximately. To that end we perform a simple local best fit

$$\tilde{f}[k] \stackrel{\text{def.}}{=} \underset{g \in \mathcal{M}^{w_0}}{\operatorname{argmin}} \sum_{t_1=-nw_0/2}^{nw_0/2} \sum_{t_2=-nw_0/2}^{nw_0/2} |f[k+t] - g[t]|^2. \quad (4)$$

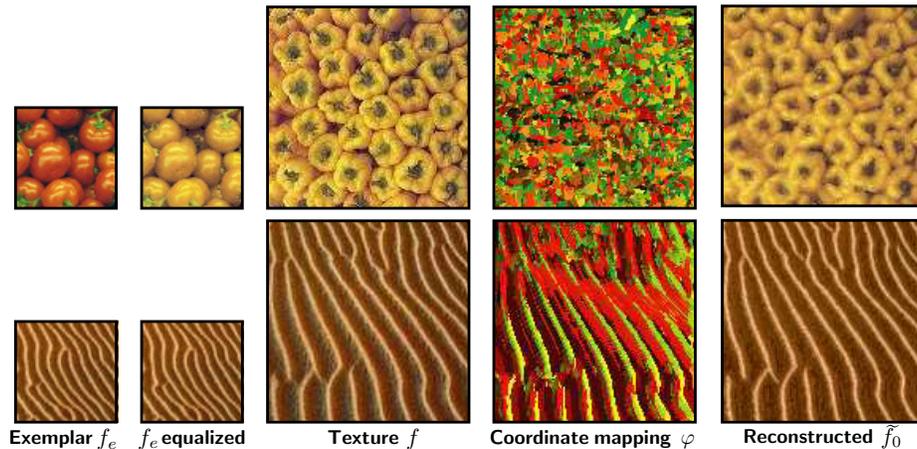
The minimization of equation (4) extends to color images. In practice a color equalization is performed on  $f_e$  prior to the matching as shown in image 2.

- Performing signal processing for manifold-valued functions. The general frameworks of Ur Uaman et al. [16] is not suited for processing patches that have some spacial connexions. We develop our own computation model and derive a multiscale decomposition in this paper.

*Examples of texture representations.* In the case of a texture manifold  $\mathcal{M}^{w_0}$  learned from an exemplar  $f_e$  using equation (1), the estimation of  $\tilde{f}$  from a given texture  $f$  defines a 2D-valued mapping

$$\varphi : [0, 1]^2 \rightarrow [0, 1]^2, \quad \text{where} \quad \forall x \in [0, 1]^2, \quad \tilde{f}(x) = p_{\varphi(x)}^{w_0} \in \mathcal{M}^{w_0},$$

where  $p_t^w$  is the patch extracted from  $f_e$  at some position  $t$ , as defined by equation (1). A similar coordinate mapping  $\varphi$  is generated by the texture synthesis algorithm of Lefebvre and Hoppe [8,9].



**Fig. 2.** Example of coordinate mapping  $\varphi$  estimated for two different pairs of textures  $(f_e, f)$ . The mapping  $\varphi(x) = (\varphi_1(x), \varphi_2(x)) \in \mathbb{R}^2$  is depicted using a red color for  $\varphi_1$  and green for  $\varphi_2$ . Prior to computing  $\varphi$ , the colors of  $f_e$  are equalized to match those of  $f$ . The reconstructed texture is  $\Phi_{w_0} \circ \tilde{f}$ .

Figure 2 shows two examples of mappings  $\varphi$  estimated for target textures  $f$  of a size twice bigger (in pixels) than the size of the exemplar  $f_e$ . For natural images, the mapping  $\varphi$  is usually piecewise linear and exhibit jump discontinuities. The right column of figure 2 shows the function  $\Phi_{w_0} \circ \tilde{f} \approx f$  reconstructed from the manifold representation  $\tilde{f}$ . This reconstruction satisfies  $(\Phi_{w_0} \circ \tilde{f})(x) = f_e(\varphi(x))$ . If the learned manifold  $\mathcal{M}^{w_0}$  does not contains the local geometric structures of  $f$ , the reconstruction might differ from the original texture. On the example of figure 2 (upper row), one can see that the green line features of the pepper image are lacking.

### 3 Association Field over the Patch Manifold

The mean computation of two features  $g, g' \in \mathcal{M}^w$  must take into account the relative positions of the features in the image to process. The two features

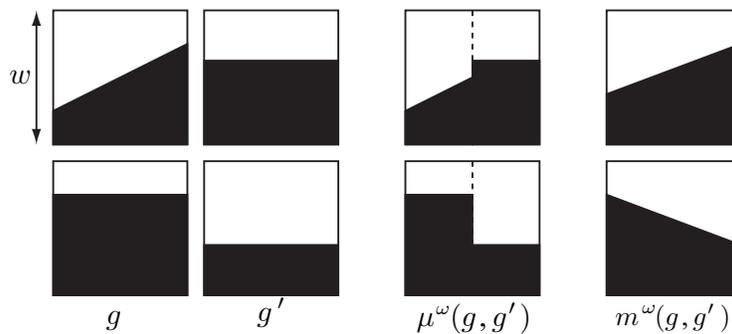
are thus assumed to be samples of some manifold valued function  $\tilde{f}$  such that  $\tilde{f}(x) = g \in \mathcal{M}^w$  and  $\tilde{f}(x') = g' \in \mathcal{M}^w$ . For the purpose of our multiscale processing framework, one only needs to compute such average for horizontal  $\omega = (1, 0)$  and vertical  $\omega = (0, 1)$  alignments of features. We thus assume that  $x - x' = w\omega$ . We describe the computation of the mean mapping in the horizontal case, the vertical one being similar. Starting from two feature elements  $g, g' \in \mathcal{M}^w$ , a new patch is computed by mixing the right side of  $g$  with the left side of  $g'$

$$\mu^\omega(g, g')(x) \stackrel{\text{def.}}{=} \begin{cases} g(x_1 + w/2, x_2) & \text{if } x_1 < 0, \\ g'(x_1 - w/2, x_2) & \text{if } x_1 \geq 0. \end{cases}$$

The mean  $m^\omega(g, g') \in \mathcal{M}^w$  of  $g$  and  $g'$  in the direction  $\omega$  is then computed as the feature that best matches this mixing image

$$m^\omega(g, g') \stackrel{\text{def.}}{=} \operatorname{argmin}_{h \in \mathcal{M}^w} \|\mu^\omega(g, g') - h\|.$$

Figure 3 illustrates the process of mean features computations  $\mathcal{M}^w$ .



**Fig. 3.** Computation of the mean  $m^\omega(g, g')$  of two edge features  $g, g' \in \mathcal{M}^w$  along direction  $\omega = (1, 0)$ .

This numerical algorithm is a simple way to estimate an association field between feature elements. Similar ideas are used in graphics for texture synthesis using copy of patches, see for instance [3,8]. More complex approaches relying on a biological or variational modeling of the association field could be used [12,17].

## 4 Multiscale Feature-valued Transform

The multiscale transform decomposes a given patch valued image  $\tilde{f} : [0, 1]^2 \mapsto \mathcal{M}^{w_0}$  of  $n \times n$  values into a set of  $n^2$  coefficients  $\{d_j\}_{j=1}^J \cup \{\tilde{f}_J\}$  that encodes at each scale  $j \leq J \stackrel{\text{def.}}{=} \log_2(n^2)$  the details needed to reconstruct the features of the original texture. This transform is inspired by the interpolating manifold

wavelet decomposition of Ur-Raman et al. [16] but with important differences that make it suitable for the analysis of local image features.

*Manifold parameterization.* In order to carry computations along the manifold  $\mathcal{M}^w$  we assume that it is globally parameterized by a bijective mapping

$$\gamma_w : \Omega \subset \mathbb{R}^p \rightarrow \mathcal{M}^w. \quad (5)$$

Such a parameterization is hardly available in practice. In numerical applications, we compute an approximate mapping using the Locally Linear Embedding [13]. The goal of this dimensionality reduction is to compute wavelet coefficients as differences in the parameter domain  $d = \gamma_w(g) - \gamma_w(g') \in \mathbb{R}^p$  of two patches  $g, g' \in \mathcal{M}^w$ . Ur-Raman et al. [16] define a wavelet difference  $d$  as a vector in the tangent plane of the manifold that is tangent to the geodesic joining  $g$  and  $g'$  and whose magnitude is the length of the geodesic. We use a parametric approach to cope with our patch-valued data. The value of the dimensionality  $p$  should reflect the complexity of the local geometry of  $f_e$  and we use  $p = 5$  in our numerical applications.

<p>Function <math>[\{d_j^\omega\}_{j,\omega}, \tilde{f}_J] = \text{multiscale\_transform\_fwd}(\tilde{f})</math>.</p> <p><i>Input</i> : feature valued image <math>\tilde{f}</math>,</p> <p><i>Output</i> : detail coefficients <math>d_j^\omega</math> for <math>j = 1, \dots, J</math> and <math>\omega \in \{(1, 0), (0, 1)\}</math>, coarse approximation at scale <math>J</math>, <math>f_J</math>.</p> <p><i>Initialization</i> : set <math>\tilde{f}_0 = \tilde{f}</math>.</p> <p>For <math>j = 1, \dots, J = \log_2(n)</math>,</p> <ul style="list-style-type: none"> <li>Set <math>w_j = 2^j/n</math>.</li> <li>(Expand) Consider <math>f_j[k] = f_{j-1}[k] \in \mathcal{M}^{w_j}</math>.</li> <li>For <math>\omega = (1, 0)</math> and then <math>\omega = (0, 1)</math> <ul style="list-style-type: none"> <li>(Split) Compute even and odd set of coefficients:  <math>f_j^e[k_1, k_2] = f_j[2k_1, k_2]</math> and <math>f_j^o[k_1, k_2] = f_j[2k_1 + 1, k_2]</math>.</li> <li>(Extract) Compute <math>\mu = m^\omega(f_j^e[k_1, k_2], f_j^o[k_1 + 1, k_2])</math> and set  <math>d_j^\omega[k_1, k_2] = \gamma_{w_j}(\mu) - \gamma_{w_j}(f_j^o[k_1, k_2])</math> and <math>f_j[k_1, k_2] = f_j^e[k_1, k_2]</math>.</li> <li>(Swap) Switch lines and columns : <math>f_j[k_1, k_2] = f_j[k_2, k_1]</math>.</li> </ul> </li> </ul> <p>End</p>
--

**Table 1:** Pseudo code for the forward feature valued transform.

Reconstruction from the wavelet coefficients requires the computation of the inverse mapping  $\gamma_w^{-1}$ . In numerical applications where  $\mathcal{M}^w$  is estimated using a discrete set of points, we use a nearest neighbor search

$$\forall x \in \mathbb{R}^d, \quad \gamma_w^{-1}(x) \stackrel{\text{def.}}{=} \underset{g \in \mathcal{M}}{\operatorname{argmin}} \|\gamma_w(g) - x\|_{\ell^2}^2.$$

*Feature-valued Wavelet Decomposition* In the following, we use cyclic boundary conditions, which means that we formally write  $v[k] = v[k + n]$  for a vector  $v \in \mathbb{R}^n$ . For the analysis of non-periodic textures, we use an extension by symmetrization along the boundaries of the image.

The algorithm starts with an initial feature valued mapping  $\tilde{f}_0 = \tilde{f}$  of  $n \times n$  values. Each step generates successively coarser features valued images  $\tilde{f}_j[k] \in \mathcal{M}^{w_j}$  of  $n/2^j \times n/2^j$  coefficients, with  $w_j = 2^j/n$ .

Each column of the array  $\tilde{f}_j$  is processed and we drop the column index to simplify the notations. The same computation is then performed on the rows of the array, as described in the pseudo code 1. The current vector  $\tilde{f}_j$  is split into even and odd coefficients

$$\forall k = 0, \dots, n/2^j - 1, \quad \tilde{f}_j^e[k] \stackrel{\text{def.}}{=} \tilde{f}_j[2k] \quad \text{and} \quad \tilde{f}_j^o[k] \stackrel{\text{def.}}{=} \tilde{f}_j[2k + 1].$$

The even coefficients are used to predict the values of the odd coefficients using the averaging operator described in section 3

$$\tilde{f}_j^o[k] \approx \mu \stackrel{\text{def.}}{=} m^\omega(\tilde{f}_j^e[k], \tilde{f}_j^e[k + 1]) \quad \text{where} \quad \omega = (1, 0).$$

The wavelet decomposition at scale  $j$  along direction  $\omega$  encodes the prediction error, measured using differences computed over the parameter domain

$$d_j^\omega[k] \stackrel{\text{def.}}{=} \gamma_{w_j}(\mu) - \gamma_{w_j}(\tilde{f}_j^o[k]) \in \mathbb{R}^p.$$

The even coefficients are then copied  $\tilde{f}_j = \tilde{f}_j^e$  in order to be further processed along the rows. Once  $\tilde{f}_j$  as been processed, the resulting feature valued image of  $n/2^{j+1} \times n/2^{j+1}$  values is copied into  $\tilde{f}_{j+1}$  considered as taking values into the coarser manifold  $\mathcal{M}^{w_{j+1}}$ . Note that at each iteration of the analysis, the assignment  $\tilde{f}_j \leftarrow \tilde{f}_{j-1}$  assumes a switch from elements  $\tilde{f}_{j-1}[k] \in \mathcal{M}^{w_{j-1}}$  to elements  $\tilde{f}_j[k] \in \mathcal{M}^{w_j}$ .

The resulting wavelet transform can be inverted by retrieving the odd coefficients  $\tilde{f}_j^o$  from the available details  $d_j^\omega$  and even coefficients  $\tilde{f}_j^e = \tilde{f}_j$

$$\tilde{f}_j^o[k] = \gamma_{w_j}^{-1}(\gamma_{w_j}(\tilde{f}_j^e[k]) - d_j^\omega[k]).$$

The pseudo code 1 implements the forward transform, whereas the pseudo code 2 implements the reverse transform. Figure 7 (b) shows the magnitude of the wavelet coefficients packed similarly to the orthogonal wavelet coefficients [10].

<p>Function <math>\tilde{f} = \text{multiscale\_transform\_bwd}(\{d_j^\omega\}_{j,\omega}, f_J)</math>.</p> <p><i>Input</i> : detail coefficients <math>d_j^\omega</math> for <math>j = 1, \dots, J</math> and <math>\omega \in \{(1, 0), (0, 1)\}</math>, coarse approximation at scale <math>J</math>, <math>f_J</math>.</p> <p><i>Output</i> : feature valued image <math>\tilde{f}</math>.</p> <p><i>Initialization</i> : set <math>\tilde{f}_J = \tilde{f}</math>.</p> <p>For <math>j = J = \log_2(n), \dots, 1</math>,</p> <table border="1"> <tr> <td>Set <math>w_j = 2^j/n</math>.</td> </tr> <tr> <td>For <math>\omega = (0, 1)</math> and then <math>\omega = (1, 0)</math></td> </tr> <tr> <td>(Reconstruct) Compute <math>\mu = m^\omega(f_j[k_1, k_2], f_j[k_1 + 1, k_2])</math> and set <math>f_j^o[k_1, k_2] = \gamma_{w_j}^{-1}(\mu - d_j^\omega[k_1, k_2])</math> and <math>f_j^e[k_1, k_2] = f_j[k_1, k_2]</math>.</td> </tr> <tr> <td>(Join) Compute coefficients: <math>f_j[2k_1, k_2] = f_j^e[k_1, k_2]</math> and <math>f_j[2k_1 + 1, k_2] = f_j^o[k_1, k_2]</math>.</td> </tr> <tr> <td>(Swap) Switch lines and columns : <math>f_j[k_1, k_2] = f_j[k_2, k_1]</math>.</td> </tr> <tr> <td>End</td> </tr> <tr> <td>(Contract) Consider <math>f_{j-1}[k] = f_j[k] \in \mathcal{M}^{w_{j-1}}</math>.</td> </tr> <tr> <td>End</td> </tr> </table> <p>Output <math>\tilde{f} = \tilde{f}_0</math>.</p>	Set $w_j = 2^j/n$ .	For $\omega = (0, 1)$ and then $\omega = (1, 0)$	(Reconstruct) Compute $\mu = m^\omega(f_j[k_1, k_2], f_j[k_1 + 1, k_2])$ and set $f_j^o[k_1, k_2] = \gamma_{w_j}^{-1}(\mu - d_j^\omega[k_1, k_2])$ and $f_j^e[k_1, k_2] = f_j[k_1, k_2]$ .	(Join) Compute coefficients: $f_j[2k_1, k_2] = f_j^e[k_1, k_2]$ and $f_j[2k_1 + 1, k_2] = f_j^o[k_1, k_2]$ .	(Swap) Switch lines and columns : $f_j[k_1, k_2] = f_j[k_2, k_1]$ .	End	(Contract) Consider $f_{j-1}[k] = f_j[k] \in \mathcal{M}^{w_{j-1}}$ .	End
Set $w_j = 2^j/n$ .								
For $\omega = (0, 1)$ and then $\omega = (1, 0)$								
(Reconstruct) Compute $\mu = m^\omega(f_j[k_1, k_2], f_j[k_1 + 1, k_2])$ and set $f_j^o[k_1, k_2] = \gamma_{w_j}^{-1}(\mu - d_j^\omega[k_1, k_2])$ and $f_j^e[k_1, k_2] = f_j[k_1, k_2]$ .								
(Join) Compute coefficients: $f_j[2k_1, k_2] = f_j^e[k_1, k_2]$ and $f_j[2k_1 + 1, k_2] = f_j^o[k_1, k_2]$ .								
(Swap) Switch lines and columns : $f_j[k_1, k_2] = f_j[k_2, k_1]$ .								
End								
(Contract) Consider $f_{j-1}[k] = f_j[k] \in \mathcal{M}^{w_{j-1}}$ .								
End								

**Table 2:** Pseudo code for the backward feature valued transform.

## 5 Texture Analysis and Synthesis

In this section, an input texture  $f_e$  is used to provide an estimation of the feature manifolds  $\mathcal{M}^w$  for dyadic sizes  $w$ . The multiscale transform is used with these manifolds to perform texture synthesis and modification.

### 5.1 Texture Synthesis

An ideal interpolation of a coarse scale sets of coefficients can be constructed using the inverse multiscale transform. The user defines the four coefficients composing  $\tilde{f}_{j_0}$  for the scale  $j_0 = 1$  (they are chosen at random in our examples). The wavelet coefficients  $\{d_j^\omega\}_{j,\omega}$  are set to zero in order to deviate as little as possible from the exemplar texture  $f_e$ . Coefficients at scale  $j = 1$  define four corners of the texture and successive steps in the backward transform (pseudo-code 2) perform dyadic refinements which compute feature valued functions  $\tilde{f}_j$  of increasing dyadic sizes. The corresponding images  $f_j \stackrel{\text{def.}}{=} (\Phi_{w_j} \circ \tilde{f}_j)$  have  $2^j \times 2^j$  pixels and can be seen as a non linear interpolation. The final synthesized texture is  $f \stackrel{\text{def.}}{=} (\Phi_{w_J} \circ \tilde{f})$ . The pixels of these intermediate images are sampled from the exemplar texture via a coordinate mapping  $f_j[k] = f_e[\varphi_j[k]]$  and the mapping  $\varphi_j$  is refined through the iterations. Figure 4 shows the iterations of the synthesis algorithm which refines  $f_j$  and  $\varphi_j$ .

Figure 5 shows two examples of texture synthesis. Note that the size of the synthesized textures can be arbitrary large. We use periodic boundary conditions for the backward transform so that the resulting texture is periodic (although the exemplar needs not be) and tiles the plane.

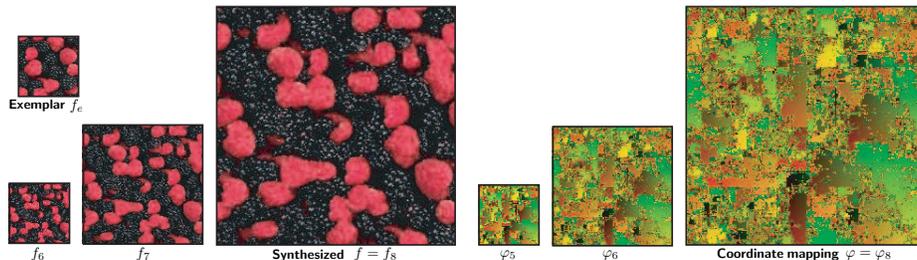


Fig. 4. Progression of the synthesis algorithm.

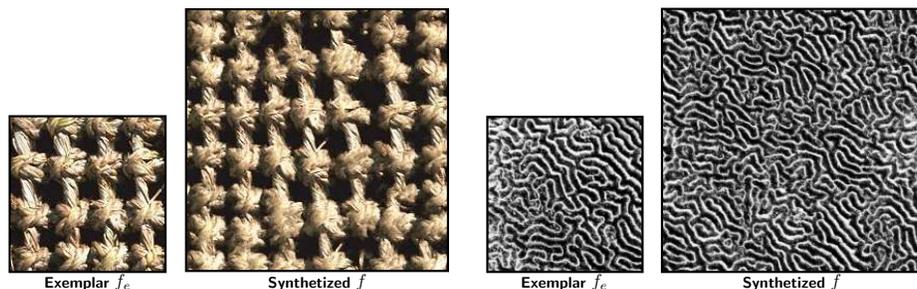


Fig. 5. Two examples of texture synthesis using zero padding in the multiscale domain.

In order to deviate from the original texture, one can use non-zero wavelet coefficients. More precisely, we define a parametric texture model over the wavelet coefficients using

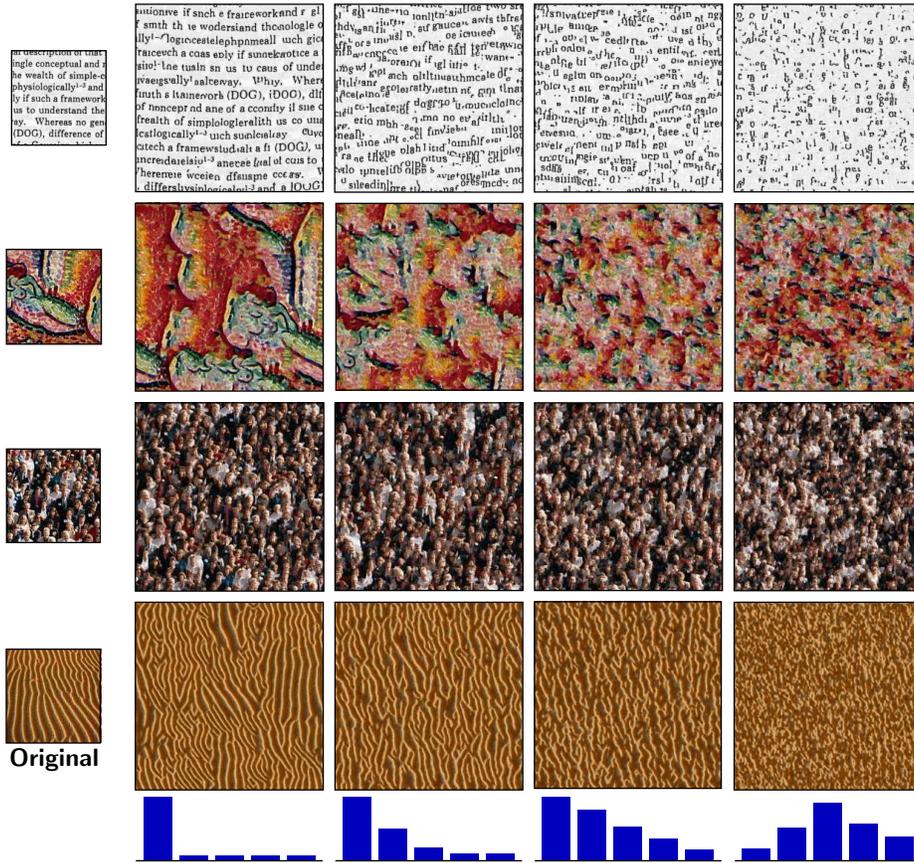
$$\forall j > 0, \quad \forall \omega, \quad \forall k \in \{0, \dots, n/2^j - 1\}^2 \quad d_j^\omega[k] \sim X(\sigma_j),$$

where  $\sim$  means that each coefficient is drawn from a gaussian random variable  $X(\sigma_j)$  of mean 0 and variance  $\sigma_j$  which allows a user control of the synthesis. Figure 6 shows different examples of texture synthesis using various spectral content  $\sigma_j$ .

## 5.2 Texture Modification

In contrast to texture synthesis, texture modification requires both the forward and backward multiscale transform. The modification process takes as input an original texture  $f$ , which is to be modified according to some exemplar texture  $f_e$ . This exemplar texture is used as the exemplar model for the feature valued multiscale transform.

In a pre-processing step, a feature valued mapping  $\tilde{f}$  is computed from  $f$  following equation (4). The forward transform described in listing 1 is used to decompose  $\tilde{f}$  as wavelets details  $\{d_j^\omega\}_{j,\omega}$ . The texture can be modified in the wavelet domain by applying a non-linear thresholding that removes small



**Fig. 6.** Texture generation with varying multiscale spectral distributions. The bottom line shows the spectral variation  $\sigma_j$  (coarse scales on the left). A small band indicate little deviation with respect to an ideal synthesis for the considered scale.

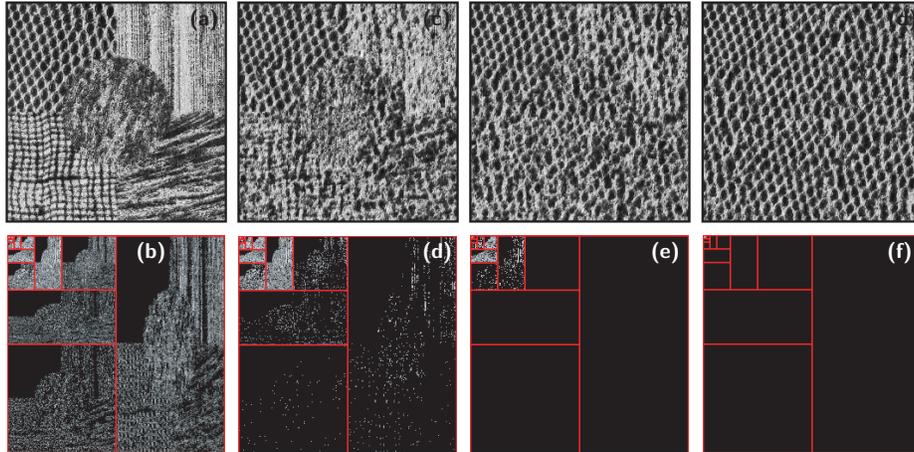
coefficients whose magnitude is below a threshold  $T$

$$d_{j,T}^\omega[n] \stackrel{\text{def.}}{=} \begin{cases} d_j^\omega[n] & \text{if } \|d_j^\omega[n]\| \leq T, \\ 0 & \text{otherwise.} \end{cases}$$

A more complex thresholding strategy could be used to include a normalization of the wavelet coefficients but we do not use it in our numerical experiments. Note that each wavelet detail  $d_j^\omega[n]$  is a  $p$ -dimensional vector so its magnitude  $\|d_j^\omega[n]\|$  is computed using the usual norm in  $\mathbb{R}^p$ . The thresholded wavelet coefficients  $d_{j,T}^\omega$  are used to synthesize a modified feature valued function  $\tilde{f}_T$  using the backward transform described in listing 2. The modified texture is computed as  $f_T \stackrel{\text{def.}}{=} (\Phi_{2^j} \circ \tilde{f}_T)$ .

On figure 7, bottom row, one can see how the wavelets coefficients are progressively shrunk toward zero. When  $T = \max(d_j^\omega)$ , the modification algorithm performs a pure texture synthesis since all wavelets coefficients are set to zero.

On figure 8 one can see how the modification process smoothly interpolates between the two textures by progressively adding and removing texture structures when wavelet coefficients are thresholded to zero.



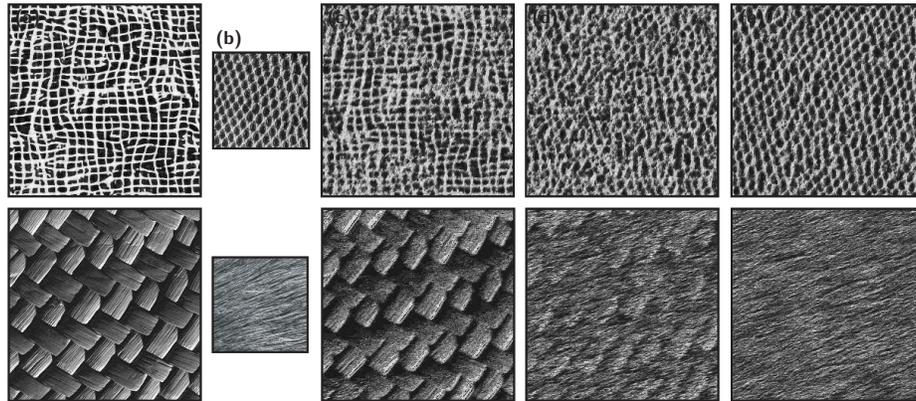
**Fig. 7.** *Progressive modification of a texture. (b,d,f,h) Magnitude  $S$  of the thresholded wavelets coefficients for  $T/\max d_j^{\sigma} = 0, 0.1, 0.5$  and 1 respectively. (a,c,e,g) Corresponding reconstructed textures.*

## 6 Conclusion

In this paper, we have proposed a new texture model as a mapping into a non-linear features manifold. A multiscale decomposition of such a mapping is performed by using a pairwise association field. This non-linear analysis and synthesis framework allows to perform feature preserving texture generation and modification. This new model is promising for capturing multiple structures that exists in non-homogeneous textures using several target manifolds. Taking into account non-manifold constraints such as symmetries is another avenue for future work.

## References

1. E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise  $c_2$  singularities. *Comm. Pure Appl. Math.*, 57(2):219–266, 2004.
2. D.L. Donoho and C. Grimes. Image manifolds isometric to euclidean space. *J. Math. Imaging Computer Vision*, 23, July 2005.
3. A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1033. IEEE Computer Society, 1999.



**Fig. 8.** Two examples of texture modification. (a) Input texture  $f$ . (b) Exemplar texture  $f_e$ . (c,d,e) Reconstruction from the thresholded wavelet coefficients for thresholds  $T/\max(d_j^w) = 0.1, 0.5$  and  $1$  respectively.

4. D. J. Heeger and J. R. Bergen. Pyramid-Based texture analysis/synthesis. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 229–238. ACM SIGGRAPH, Addison Wesley, August 1995.
5. B. Julesz. Visual pattern discrimination. *IRE Trans. Inform. Theory*, 8(2):84–92, 1962.
6. E. Le Pennec and S. Mallat. Bandelet Image Approximation and Compression. *SIAM Multiscale Modeling and Simulation*, page to appear, 2005.
7. A.B. Lee, K.S. Pedersen, and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1-3):83–103, August 2003.
8. S. Lefebvre and H. Hoppe. Parallel controllable texture synthesis. *ACM Trans. Graph.*, 24(3):777–786, 2005.
9. S. Lefebvre and H. Hoppe. Appearance-space texture synthesis. *ACM Trans. Graph.*, 2006.
10. S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
11. W. Matusik, M. Zwicker, and F. Durand. Texture design using a simplicial complex of morphable textures. *ACM Trans. Graph.*, 24(3):787–794, 2005.
12. D. Mumford. Elastica and computer vision. In C. L. Bajaj (Ed.), *Algebraic geometry and its applications*, pages 491–506, 1994.
13. S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, Dec. 2000.
14. A. Szlam, M. Maggioni, and R. R. Coifman. A general framework for adaptive regularization based on diffusion processes on graphs. *Yale technical report*, July 2006.
15. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, Dec. 2000.
16. I. Ur-Raman, I. Drori, V. Stodden, and P. Schroeder. Multiscale representations of manifold-valued data. *to appear in SIAM Multiscale Modeling and Simulation*, 2005.
17. L.R. Williams and D. W. Jacobs. Stochastic completion fields: a neural model of illusory contour shape and salience. *Neural Comput.*, 9(4):837–858, 1997.