Oriented Patterns Synthesis

Gabriel Peyré
Ceremade, Université Paris Dauphine,
Place du Marchal De Lattre De Tassigny,
75775 Paris Cedex 16, France
gabriel.peyre@ceremade.dauphine.fr

April 26, 2007

Abstract
This paper explores two different methods to capture the local anisotropy of locally parallel textures. The corresponding models allow for both an analysis and a synthesis of oriented patterns. The first method models such textures as local oscillations that can be analyzed by a windowed Fourier transform. The resulting model can be sampled by alternating projections that enforce the consistency of the local phase with the estimated spectrum of the input texture. The second method models the texture over the domain of local structure tensors. These tensors encode the local energy, anisotropy and orientation of the texture. The synthesis is performed by matching the multiscale histograms of the tensor fields.

Figure 1: Examples of locally parallel textures.

This paper exposes two models for analyzing and synthesizing natural textures that exhibit a local anisotropy. In section 3 we propose a model based on a local Fourier expansion for the analysis and on iterated projections for the synthesis of the phase function. In section 4, we propose a statistical model that captures the variations of the orientation over a tensor domain.

1 Previous Works

Analysis of oriented patterns. Oriented patterns provide key features in computer vision and find applications in many processing problems such as fingerprints analysis [9]. Their analysis is performed through the application of local differential operators averaged over the image plane.
Synthesis of oriented patterns. Computer-graphics approaches to texture synthesis allow to generate high quality textures from a single input image through a consistent recopy of texture pixels and patches [7, 33]. These approaches do not provide a compact model for specific classes of textures such as highly oscillating patterns.

Statistical modeling of natural images [29] seeks a compact representation in order to capture the redundancy inherent to the geometric features of textures. Most of the models are built over a multiscale domain such as wavelets and allow to synthesize images with isotropic singularities such as clouds or marble [23, 8, 4]. In order to incorporate elongated geometrical features such as oriented patterns, one needs to add advanced multiscale statistical models to capture the redundancy of wavelet coefficients [36, 25].

Parametric methods such as particles advection [32, 22] can synthesize turbulent textures. Reaction-diffusion equations [30, 35] is an alternative way to synthesize oriented patterns such as elongated stripes.

Sparse representation of oriented patterns. Harmonic analysis tools such as wavelets [17] have proven useful to perform image processing of natural images. They are however not optimal to compress complex images containing geometrical structures. Oriented and directional decompositions [28, 6, 20] allow a fine tuning of the decomposition that enhances the results of synthesis [8] with respect to isotropic gaussian noise [23], but does not provide a completely sparse decomposition of oriented textures. Adapted transforms such as bandlets [14, 19] allow to sparsify cartoon images and the grouplet transform [18] extends these ideas to compress locally oriented patterns.

In order to model textures, Meyer proposed the used of banach spaces that favor oscillating patterns, such as clouds or marble [23, 8, 4]. In order to incorporate elongated geometrical features such as oriented patterns, one needs to add advanced multiscale statistical models to capture the redundancy of wavelet coefficients [36, 25].

2 Multiscale Non-parameteric Modeling of Textures

Non parametric modeling of marginal distributions. Given some random variable $X$ from which a set of samples $x = \{x_i\}_i$ is available, a non-parametric model is estimated from its histogram $H(x)$. Starting from some initial set of samples $x^0$ representing features extracted from a texture, synthesis is performed by modifying $x^0$ so that its histogram matches the one of an exemplar set of samples $x$. This amount to computing the projection $P_{H(x)}x^0$ of $x^0$ on the set of vectors $\{y \in H(y) = H(x)\}$. If both $x$ and $x^0$ have $m$ points, one can perform this equalization $\tilde{x} = P_{H(x)}x^0$ by sorting the samples

$$\forall i, \quad \tilde{x}(\gamma^0(i)) = x(\gamma(i)) \quad \text{where} \quad x^0(\gamma^0(i)) \leq x^0(\gamma^0(i)) \quad \text{and} \quad x(\gamma(i-1)) \leq x(\gamma(i)). \quad (1)$$

In the case where $x$ and $x^0$ do not have the same number of samples, formula (1) needs to be applied with interpolation.

Orientation modeling. In the following, we also model circle valued samples $x_i \in [0, \lambda \pi)$ for $\lambda = 1$ (orientations) or $\lambda = 2$ (directions). The values of such data are defined modulo $\lambda \pi$, so that treating $x$ as real valued leads to undesirables discontinuities. To avoid this difficulty, the data is embedded in the circle

$$(a_i, b_i) = (\cos(2x_i/\lambda), \sin(2x_i/\lambda))$$

The equalization of a set of circle valued samples $x^0$ in order to match some input data $x$ is done by equalizing each coordinate of $(a^0, b^0)$ and computing back the corresponding angle

$$P_{H(x)}x^0 \stackrel{\text{def}}{=} \frac{\lambda}{2} \angle \left(P_{H(a)}a^0, P_{H(b)}b^0\right).$$

2
This process is similar to the averaging of angles performed during the orientation diffusion process [11, 24].

**Multiscale transform.** In order to model some slowly varying function $g(x)$, we use a multiscale transform that computes inner products against translated and dilated version of mother wavelets functions $\{\psi^k\}_{k=1}^K$ [17]

$$g^k_j(n) \overset{def.}{=} \langle g, \psi^k_{j,n} \rangle \quad \text{where} \quad \psi^k_{j,n}(x) = \frac{1}{2^j} \psi^k(2^j x - n).$$

For the numerical results, we have used the steerable pyramid wavelets [28] with $K = 4$ directional subbands.

**Multiscale feature model.** A multiscale model for set of spatially defined features $x \in [0, 1]^2 \mapsto g(x)$ consists in imposing the marginal distribution of the set of coefficients $g^k_j$ for each orientation $k$ and scale $j$. We denote by $\mathcal{H}(g) = H(g) \cup \{H(g^k_j)\}_{k,j}$ the union of all the histograms that compose this multiscale model, as originally proposed by Heeger and Bergen [8].

The projection of $g^0$ on each of the constraint is denoted as $\mathcal{P}_{\mathcal{H}(g)}g^0$. This projection is computed as follow

1. Compute the set of multiscale coefficients $(g^0)^k_j$ of $g^0$.
2. Impose the histograms of the multiscale decomposition
3. Inverse the transform to retrieve the feature mapping. If the set of basis function $\psi^k_{j,n}$ is a tight frame (as this is the case for the steerable pyramid [28]), this reconstruction is
4. Perform matching over the spatial domain to complete the overall projection:

$$\mathcal{P}_{\mathcal{H}(g)}g^0 \overset{def.}{=} \mathcal{P}_{\mathcal{H}(g)}g^0.$$ 

In order to match all these constraints simultaneously, Heeger and Bergen [8] iterate the projector $\mathcal{P}_{\mathcal{H}(g)}$ directly over the image to synthesize. This leads to the following algorithm

1. (Initialization) Start from an initial white noise texture: $f^0 \leftarrow \text{noise}$. Set $k = 0$.
2. (Multiscale matching) Impose the multiscale constraints over the image $f^{k+1} \leftarrow \mathcal{P}_{\mathcal{H}(f)}f^k$.
3. While not converged, set $k \leftarrow k + 1$ and go back to 2.

Figure 2 shows an example of these iterative projections. The multiscale constraints are not able to characterize the geometry of locally parallel textures. Sections 3 and 4 show how one can build constraints over alternative feature spaces that can capture this geometrical structure. Section 3 considers the local phase of the texture and build the multiscale set of constraints over this space. Section 4 considers a tensor domain and does the synthesis over the parameters of these tensors.

More advanced modeling requires to the use of adapted filtering processes such as the Frame model of Zhu et al. [36]. The resulting model requires a Gibbs sampling in order to take into account the dependancy between the constraints. Another option is to use higher order statistical constraints, see the work of Portilla and Simoncelli [25]. The resulting model is more difficult to analyze due to the large number of parameters, although it can synthesize textures with complex geometrical patterns. In this paper, we show that for the specific class of oriented patterns, a multiscale model defined over a well chosen set of parameters captures the geometry of the texture.
3 Phase Domain Modeling and Synthesis

3.1 Locally Parallel Textures

Some natural textures are composed of nearly parallel stripes that can be modeled as local oscillations. A locally parallel texture is defined as

$$f(x) = A(x) \cos(\Phi(x))$$ (2)

where $\nabla_x \Phi$ controls the direction and frequency of the oscillations around the point $x$ and $A(x)$ controls the local energy of the texture. A locally parallel texture is characterized by a slowly varying amplitude $A$ and phase $\nabla_x \Phi$. Figure 1 shows some examples of such locally parallel textures.

Parameters estimation. From a given input texture $f(x)$, one can estimate the local frequency $\eta(x) \approx \nabla_x \Phi$ and amplitude $A(x)$ of $f$ using a windowed Fourier transform. We use a smooth window function $h$ supported on $[-\sigma/2, \sigma/2]$ where the scale $\sigma$ should be larger than the length of oscillations. The spectrogram of $f$ is defined as

$$\hat{f}(x, \omega) \overset{\text{def.}}{=} \int h(t) f(x + t) \exp(-i \langle \omega, t \rangle) dt.$$ (3)

Similarly to the estimation of the instantaneous frequency of sounds (see Delprat et al. [5] and [17]), the local parameters are estimated using

$$\eta(x) \overset{\text{def.}}{=} \arg \max_{\omega} |\hat{f}(x, \omega)| \quad \text{and} \quad A(x) = |\hat{f}(x, \eta(x))|.$$ (3)

Figure 3 shows how this local spectral analysis allows to estimate $\varphi$ on a natural image. In areas containing a locally parallel texture, the Fourier expansion exhibits two symmetric bumps. In areas containing an edge, the Fourier expansion has high amplitude along a 1D segment orthogonal to the direction of the edge.

Three local parameters are defined from the local phase and amplitude

$$(A(x), \rho(x), \theta(x)) \in \mathbb{R}^+ \times \mathbb{R}^+ \times [-\pi/2, \pi/2] \quad \text{where} \quad \eta(x) = \pm \rho(x)(\cos(\theta(x)), \sin(\theta(x))).$$ (4)

These three mappings $(A, \rho, \theta)$ characterize locally the amplitude, frequency and orientation of the oscillations of the texture $f$.

Unwrapping the local orientation. The major difficulty with this Fourier domain estimation is that the local direction $\theta(x)$ is defined only modulo $\pi$. In order to retrieve the direction of this orientation, one needs to compute a sign field $\varepsilon(x) \in \{-1, +1\}$ such that the vector field $\varepsilon(x)(\cos(\theta(x)), \sin(\theta(x)))$ is as smooth as possible. By relaxing the binary constraint $\varepsilon \in \{-1, +1\}$, we solve the following Laplacian problem

$$\forall x, \quad \varepsilon(x) = \frac{1}{|V_x|} \sum_{y \in V_x} \varepsilon(y) \cos(\theta(x) - \theta(y)),$$ (5)
Figure 3: Examples of local Fourier expansions showing the estimation of the local phase parameters \((A(x), \rho(x), \theta(x))\).

where \(V_x\) is the set of neighbors around a point \(x\) (for instance using the 4 or 8 connectivity over the image domain). One needs to add the constraint \(\varepsilon(0) = 1\) to fix the overall sign under-determinacy and (5) requires to solve a sparse linear system. The set of orientations \(\theta(x)\) is then turned into a direction field

\[
\forall x, \quad \Theta(x) \stackrel{\text{def.}}{=} \theta(x) + (\text{sign}(\varepsilon(x)) + 1)\pi/2.
\]

Figure 4 shows an example of estimated parameters. One can see how the direction field \(\theta\), displayed as a modulo \(2\pi\) field, is discontinuous, whereas \(\Theta\) is smooth.

Figure 4: (a) original image \(f\), (b) estimated \(A(x)\), (c) estimated \(\rho(x)\), (d) estimated \(\theta(x)\), (e) estimated \(\Theta(x)\). The color mapping for the display of \(\theta\) (and also \(\Theta\)) uses the red channel for \(\cos(\theta(x))\) and the green channel for \(\sin(\theta(x))\).

3.2 Parametric texture synthesis

Given an input local frequency field \(\rho^0(x)\), one can synthesize a locally parallel texture \(\tilde{f}(x) = \cos(\tilde{\Phi}(x))\) such that \(\nabla_x \tilde{\Phi} = \rho^0(x)\). Such a synthesis does not model the orientation \(\theta^0(x)\) of the texture and one thus needs to carefully initialize the iterations. This leads to the following algorithm

1. (Initialization) Set \(\Phi^0 = W \ast G_\mu\), where \(W\) is a gaussian white noise and the smoothing scale \(\mu\) controls the overall regularity of the synthesized phase field. Set \(k = 0\).
2. (Impose local frequency) Compute the gradient field $v^k \equiv \nabla \Phi^k$. Normalize the vector field

$$v^{k+1}(x) = \rho^0(x) \frac{v^k(x)}{\sqrt{|v^k(x)|^2 + \varepsilon^2}},$$

where $\varepsilon$ is a small value that prevent ill-conditionned iterations.

3. (Recover the phase) $\Phi^{k+1}$ is the solution to the Poisson equation

$$\Delta \Phi^{k+1} = \text{div}(v^{k+1}).$$

with periodic boundary conditions. The texture is $f^{k+1}(x) = \cos(\Phi^k(x)+1)$

4. While not converged, set $k \leftarrow k + 1$ and go back to 2.

These iterations can be thought as iterative projections on the linear constraint $\nabla \Phi^k = 0$ and the manifold constraint $|v^k(x)| = \rho^0(x)$. Such iterative projections have been proven to converge locally $f^k \to f$ by Lewis and Malick [15].

Figure 5: Examples of locally parallel textures with various phase properties.

Figure 5 (a) shows an example of a synthetic locally parallel texture where the value of of $\rho^0(x)$ is slowly varying around 1. Figure 5 (b,c) shows an example where $\rho^0$ is constant in (b) and horizontally varying in (c). Figure 6 shows the effect of varying the smoothness $\mu$ of the initial field $\Phi^0$.

Figure 6: Examples of locally parallel textures with the same local frequency field $\rho^0(x)$ but initial potential field $\Phi^0$ of increasing smoothness $\mu$.

### 3.3 Non-Parametric texture synthesis

Instead of fixing the frequency of the synthesized texture to match a given field $\rho^0$, one can enforce the consistency of the parameters $(A^k(x), \rho^k(x), \Theta^k(x))$ of $f^k$ with a given input exemplar $f(x)$ and let these parameters evolve through the iterations. This leads to the following algorithm

1. (Initialization) Start from an initial white noise amplitude and phase $(A^0, \Phi^0) \leftarrow \text{noise}$. Set $k = 0$. 
2. **(Orientation matching)** Compute the phase gradient field and define $\Theta^k$ and $\rho^k$

$$v^k(x) = \nabla_x(\Phi^k) \quad \text{and} \quad \rho^k(x)(\cos(\Theta^k(x)), \sin(\Theta^k(x))) = v^k(x).$$

Match the parameters

$$(A^{k+1}, \rho^{k+1}, \theta^{k+1}) \leftarrow (P_{\mathcal{H}(A)}A^k, P_{\mathcal{H}(\rho)}\rho^k, P_{\mathcal{H}(\theta)}\Theta^k).$$

where $\Theta^k$ is matched as a modulo $2\pi$ field.

3. **(Recovering the phase)** The new gradient field is

$$v^{k+1}(x) = \rho^{k+1}(x)(\cos(\Theta^{k+1}(x)), \sin(\Theta^{k+1}(x))).$$

$\Phi^{k+1}$ is the solution to the Poisson equation

$$\Delta \Phi^{k+1} = \text{div}(v^{k+1}).$$

with periodic boundary conditions. The texture is

$$f^{k+1}(x) = A^{k+1}(x) \cos(\Phi^{k+1}(x))$$

4. While not converged, go back to 2.

Figure 7 shows an example of such an iterative process.

![Image](image_url)

**Figure 7:** Left: original image $f$. Right: several iteration $f^k$ of the synthesis algorithm.

## 4 Tensor Domain Modeling and Synthesis

This section proposes an alternative texture model to the local frequency parameters (4) used in section 3 to describe the phase of locally parallel textures. This model is based on local averaging of the orientation information encoded in a structure tensor. Such a tensor representation does not encode the local frequency of the oscillation but instead extract information on the local anisotropy of the texture.

### 4.1 The Structure Tensor

**Direction estimation.** From an input vector field $v : x \in [0, 1]^2 \mapsto v(x) \in \mathbb{R}^2$, the structure tensor $T_v$ is defined as a local averaging of the rank-1 tensor field $vv^T$

$$T_v(x) = (G_\sigma * (vv^T))(x) = G_\sigma * \begin{pmatrix} v_1^2 & v_1 v_2 \\ v_1 v_2 & v_2^2 \end{pmatrix} \quad \text{where} \quad G_\sigma(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp(-|x/\sigma|^2)$$

and where the convolution is applied on each component of the tensor.

In order to estimate the local orientation of an image $f$, one typically uses the gradient field $v(x) = \nabla_x(f)$, see for instance [10, 12]. This gradient is estimated numerically as the convolution against two directional derivative filters

$$v(x) = (f * h^1, f * h^2) \quad \text{where} \quad h^i(x) = \frac{\partial G_\epsilon}{\partial x_i}(x),$$
Figure 8: Left: original image. Right: two structure tensor fields computed with an increasing smoothing factor $\sigma$.

where the scale $\varepsilon$ should be of the order of 1 pixel for discrete images. More advanced filtering schemes could be used to have a better estimate of the gradient or to have a multiscale gradient flow, see for instance the steerable wavelets decomposition [28].

Figure 8 shows an example of tensor field $T$ estimated from the gradient at two different scales $\sigma$. To capture oriented patterns efficiently, this scale should approximately match the length of the oscillations of the pattern.

**Tensor decomposition.** A tensor $t \in \mathbb{R}^{2 \times 2}$ is a symmetric semi-definite matrix. It can thus be decomposed as a sum of two rank-1 tensors

$$ t = \begin{pmatrix} a & c \\ c & b \end{pmatrix} = \lambda_1 (u_1 u_1^T) + \lambda_2 (u_2 u_2^T), \quad (6) $$

The eigenvalues of $t$ are $\lambda_1 \geq \lambda_2 \geq 0$ and $u_i$ are the corresponding orthogonal eigenvectors. The set of tensors is a half-cone defined by

$$ T \triangleq \left\{ t = \begin{pmatrix} a & c \\ c & b \end{pmatrix} \mid ab - c^2 \geq 0 \quad \text{and} \quad a + b \geq 0 \right\} $$

Similarly to the polar parameterization of colors in the HSV color space, one can parameterize the set of tensors by

$$ \begin{align*}
\alpha &\triangleq \lambda_1 + \lambda_2 = a + b, \\
\beta &\triangleq \left( \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 = 4 \frac{ab - c^2}{(a + b)^2}, \\
\theta &\triangleq \text{angle}(v_1) = \frac{1}{2} \tan^{-1} \left( \frac{2c}{a+b} \right). 
\end{align*} \quad (7) $$

The parameter $\alpha$ is the energy of the tensor, $\beta$ is its anisotropy and $\theta$ its orientation. This leads to the following parameterization of the cone $T$

$$ \varphi : \left\{ \begin{array}{c}
T \\
t = \begin{pmatrix} a & c \\ c & b \end{pmatrix}
\end{array} \right\} \longrightarrow \mathbb{R}^+ \times [0,1]^2 \times [\pi/2, \pi/2] (\alpha, \beta, \theta). $$

This parameterization is inverted by using the decomposition (6) together with

$$ u_1 = (\cos(\theta), \sin(\theta)), \quad \text{and} \quad \lambda_1 = \frac{1}{2} (a + \sqrt{\beta a^2/4}), \quad \text{and} \quad \lambda_2 = a - \lambda_1. $$

**Tensor field analysis.** From a tensor field $T_v(x) \in T$ computed from a gradient field $v = \nabla_x f$, one can apply the parameterization $\varphi$

$$ T_v(x) = \begin{pmatrix} a(x) & c(x) \\ c(x) & b(x) \end{pmatrix} \quad \xrightarrow{\varphi} \quad \varphi(T_v(x)) = (\alpha(x), \beta(x), \theta(x)). $$
The map $\alpha(x)$ describes the local energy of $f$, $\beta(x)$ the anisotropy of the geometric structures around $x$ and $\theta(x)$ the local orientation of the patterns.

Figure 9 shows an example of maps $(a, b, c)$ and $(\alpha, \beta, \theta)$. The description of the tensor field through these three parameters $(\alpha, \beta, \theta)$ allows to modify each one while keeping the semi-definite property of tensor field. Further more, each one of these parameters has a meaningful signification.

Relation with the locally parallel model. Under the assumptions of the locally parallel texture model (2), the texture $f$, around a point $x$, is approximately a pure oscillating wave

$$\forall |t| \leq \sigma, \quad f(x + t) \approx A(x) \cos(\rho(x) \omega(x), t) + \delta(x)$$

where $(A(x), \rho(x), \theta(x))$ are the parameters introduced in equation (4) and $\delta(x)$ is some dephasing term. This local model implies that the structure tensor $T_v(x)$ computed from $v = \nabla_x f$ at $x$ is approximately parameterized as

$$T_v(x) = G_\sigma * (\nabla f \nabla f^T)(x) \approx \frac{1}{2} A(x)^2 \rho(x)^2 (\omega(x) \omega(x)^T).$$

The tensor $T_v(x)$ is thus approximately flat with its eigenvalues satisfying $\lambda_1(x) \approx \frac{1}{2} A(x)^2 \rho(x)^2$ and $\lambda_2 \approx 0$ and its first eigenvector being $u_1(x) \approx \omega(x)$. This allows to make the following partial connexion between the parameters $(\alpha, \beta, \theta)$ of equation (7) and the parameters $(A, \rho, \theta)$ of equation (4)

$$\alpha(x) \approx \frac{1}{2} A(x)^2 \rho(x)^2,$$

$\theta$ being consistently defined in both settings as the orientation of the geometry. One sees that both models keep track of the local energy $\alpha(x)$ and direction $\theta(x)$ of the oscillations of the texture. The phase model presented in section 3 is also able to track the local periodicity of the oscillations through the frequency parameter $\rho(x)$. In contrast, the tensor domain model cannot track this frequency, but encodes an additional anisotropy information $\beta(x)$. This anisotropy can detect local changes of orientation such as the local curvature of the geometry.

4.2 Recovering an Image from the Tensor Field

To modify the geometric content of an image, one needs to go back and forth from the image domain to the tensor domain. Unfortunately, the tensor representation of a vector field looses
Compute the gradient field. While not converged, go back to 1.
Perform one step of gradient descent with step size \( \eta \) for its structure tensor field to match a given tensor field. More formally, we look for a gradient \( \nabla_x \Phi \) and not of the image gradient \( \nabla_x f \).

To avoid the issue of recovering this direction information, we modify a given image \( f^0 \) in order for its structure tensor field to match a given tensor field. More formally, we look for a gradient field \( v \) close to \( \nabla_x f^0 \) such that \( T_v \approx T \) where \( T(x) = \begin{pmatrix} a(x) & c(x) \\ c(x) & b(x) \end{pmatrix} \) is a given tensor field. This is performed by the minimization of the following energy

\[
E(v) = \int_{[0,1]^2} |T(x) - T_v(x)|^2 dx = \sum_{i=1}^3 E_i(v)
\]

where

\[
\begin{align*}
E_1(v) &= \int ((G_\sigma * v_1^2)(x) - a(x))^2 dx, \\
E_2(v) &= \int ((G_\sigma * v_2^2)(x) - b(x))^2 dx, \\
E_3(v) &= \int ((G_\sigma * (v_1 v_2))(x) - c(x))^2 dx.
\end{align*}
\]

The gradient of this energy is a vector field whose coordinates are

\[
\nabla_v E = \begin{pmatrix}
(G_\sigma * (G_\sigma * v_1^2 - a))v_1 + (G_\sigma * (G_\sigma * (v_1 v_2) - c))v_2 \\
(G_\sigma * (G_\sigma * v_2^2 - b))v_2 + (G_\sigma * (G_\sigma * (v_1 v_2) - c))v_1
\end{pmatrix}
\]

Starting from an initial image \( f^0 \), we perform a projected gradient descent in order to match the tensor field with \( T \). This leads to the following iterations

1. Compute the gradient field \( v^k(x) = \nabla_x f^k \) of the current image.
2. Perform one step of gradient descent with step size \( \eta \)
   \[ v^{k+1} = v^k - \eta(\nabla_v E). \]
3. Compute the image \( f^{k+1} \) whose gradient field is close to \( v^{k+1} \) by solving the poisson equation
   \[ \Delta f^{k+1} = \text{div}(v^{k+1}) \]
   subject to periodic boundary conditions. This equation can be solved efficiently by inverting the laplacian over the Fourier domain.
4. While not converged, go back to 1.

For a small enough step size \( \eta \), the vector fields \( v^k \) converges to a local minimum \( v \) of the energy \( E \) constrained to rot\( (v) = 0 \). We denote by \( \mathcal{M}_T(f^0) = \lim_k f^k \) the image whose tensor field has been modified to match \( T \). This gradient descent modifies the image in a way similar to tensor-based adaptive anisotropic diffusions [34] but with an additional control over the resulting tensors field at convergence.

4.3 Tensor Domain Texture Synthesis

We model the tensor field over the \((\alpha, \beta, \theta)\) domain by assuming that these three maps are smooth and slowly varying. This assumption is true for textures composed of locally parallel patterns such as those depicted on the left of figure 11.

The synthesis algorithm works by iteratively imposing each constraint given by the multiscale histograms of \( f \) and \((\alpha, \beta, \theta)\).

1. (Initialization) Start from an initial white noise texture: \( f^0 \leftarrow \text{noise} \). Set \( k = 0 \).
2. (Multiscale image matching) Impose the multiscale constraint over the image: \( j^k \leftarrow \mathcal{P}_{\mathcal{H}(f)} f^k \).
3. (Tensor matching) Compute the structure tensor $T^k$ of $\tilde{f}^k$ and the corresponding parameters $(\alpha^k, \beta^k, \theta^k) = \varphi(T^k)$. Match the histograms of these three parameters

$$(\alpha^{k+1}, \beta^{k+1}, \theta^{k+1}) \leftarrow (\mathcal{P}_{\mathcal{H}(\alpha)}^k, \mathcal{P}_{\mathcal{H}(\beta)}^k, \mathcal{P}_{\mathcal{H}(\theta)}^k).$$

4. (Recovering the image) The modified tensor field is

$$T^{k+1} \leftarrow \varphi^{-1}(\alpha^{k+1}, \beta^{k+1}, \theta^{k+1}).$$

The image is recovered by using the gradient descent exposed in section (4.2)

$$f^{k+1} \leftarrow \mathcal{M}_{T^{k+1}}(\tilde{f}^k).$$

5. While not converged, set $k \leftarrow k + 1$ and go back to 2.

Step 2 this algorithm is the same as the synthesis algorithm of Heeger and Bergen [8]. Steps 3 and 4 allows to synthesize oriented patterns by imposing regularity in the orientation of the texture. Figure 11 shows some results of this synthesis process.

![Figure 10: Iterations of the synthesis algorithm.](image)

**Conclusion**

This paper presents two models of locally parallel textures. The first model captures the local orientation and frequency of oscillating patterns using a windowed Fourier expansion. The phase of the oscillations can be analyzed and synthesized using alternating projections on a set of direction and frequency constraints. The second model captures the local orientation and anisotropy of the patterns using a field of structure tensors. Such a tensor field can be modified in order to match the statistics of a given input texture. The resulting texture synthesis directly manipulate the image without reconstructing the phase.

**References**


Figure 11: Top row: original image. Middle row: synthesis using Heeger and Bergen algorithm [8]. Bottom row: synthesis using iterative matching of the structure tensor.


