A Texton for Fast and Flexible Gaussian Texture Synthesis
{Bruno.Galerne, Arthur.Leclaire, Lionel.Moisson}
@PARISDESCARTES.FR

Introduction
Gaussian textures allow for
- by-example synthesis [4],
- dynamic texture synthesis [8],
- texture mixing [8].

The classical FFT-based synthesis algorithm has some limitations:
1) The underlying model is implicitly periodic;
2) It does not allow for local variations of the kernel or the grid.

We propose here
- to approximate a Gaussian texture by a
  Discrete Spot Noise
- to compute a Synthesis-Oriented Texton

Gaussian textures can then be generated
on-demand in a faster, simpler, and more flexible way.

Spot Noise Model
Let \( h : \mathbb{Z} \rightarrow \mathbb{R}^d \) be a function with finite support \( S_h \) and let us define \( h(x) = h(-x) \).

The Discrete Spot Noise (DSN) on \( \mathbb{Z}^2 \) is
\[
F_{h,x}(y) = \sum_{y \in S_h} P_y(y) h(x - y),
\]
where \( P_y \) is a Poisson white noise with intensity \( \lambda \).

The renormalized DSN
\[
G_{\lambda,h}(x) = \frac{F_{h,x} - E(F_{h,x})}{\sqrt{\lambda}} = \frac{1}{\sqrt{\lambda}} \left( h \ast P_{\lambda} - \lambda \sum_{y \in S_h} h(y) \right)
\]
has zero-mean and covariance function \( h \ast h^T \), and
\[
G_{\lambda,h}(x) \ast \text{ADSN}(h) = h \ast W
\]
where \( W \) is a normalized Gaussian white noise on \( \mathbb{Z}^2 \).

One can also define a circular DSN on a \( M \times N \) rectangle \( \Omega \) with periodic boundary conditions. Circular convolutions can then be computed using the FFT. In particular, we can compute the \( L^2 \) optimal transport distance between circular ADSN \( \rho_{u_1}, \mu_1 \) associated to \( h_{u_1}, h_1 \):
\[
d_{OT}^2(\rho_{u_1}, \mu_1) = \sum_{\xi \in \mathbb{C}} \left( \|h_{u_1}\|^2 + \|h_1\|^2 - 2\tilde{h}_{u_1}^* h_1 \right) (\xi).
\]

We define the (squared) relative model error
\[
\text{RME}(h, h_0) = \frac{\sum_{\xi \in \mathbb{C}} \left( \|h_{u_1}\|^2 + \|h_1\|^2 - 2\tilde{h}_{u_1}^* h_1 \right) (\xi)}{\sum_{\xi \in \mathbb{C}} \|h_0\|^2(\xi)}.
\]

Synthesis-Oriented Texton
A Synthesis-Oriented Texton (SOT) for the model \( \text{ADSN}(h) \) is any kernel \( k \) such that
\[
\text{Supp}(k) \subset S \quad \text{(with prescribed } S) \quad \text{and } k \ast k^T = h \ast h^T \quad \text{for } \lambda = 1.
\]

Algorithm
Let \( u : \Omega \rightarrow \mathbb{R}^d \) be an exemplar texture.

Empirical mean:
\[
\hat{u} = \frac{1}{|\Omega|} \sum_{x \in \Omega} u(x) .
\]

Circular autocorrelation:
\[
c_{u} = t_u \ast t_u^T , \quad \text{where } t_u = \frac{1}{\sqrt{|\Omega|}} (u - \hat{u}) .
\]

The algorithm alternates between
\[
qu(h) = t_u \ast h \quad \text{(support projection)} ,
\]
\[
\hat{u}_l(h) = \frac{t_u^T h}{\|t_u\|^2} \mathcal{P}_{0}(h) \quad \text{(spectral projection)} .
\]

Convergence: Since \( \hat{u}_l \) is not the projection on the convex set, the convergence is not proved. In practice, the iterates stabilize after 50 iterations.

Comparison

<table>
<thead>
<tr>
<th>Original</th>
<th>Exemplar SOT</th>
<th>DSN(t), 10 imp./px</th>
<th>DSN(t), 30 imp./px</th>
<th>ADSN(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSN((t_u))| 0.51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(t_u))| 30 imp./pix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(t_u))| 30 imp./pix</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Color Correction

One can better preserve the color distribution by reimposing the color covariance of the exemplar texture (which amounts to apply a \( 3 \times 3 \) transformation in the color space).

Conclusion
Given an exemplar texture image \( u \), the proposed algorithm computes a synthesis-oriented texton having a prescribed small support and for which the associated DSN is close to the Gaussian texture associated with \( u \), even for a low intensity \( \lambda \). This SOT can be considered as an inverse texture synthesis solution [7] for the Gaussian model.

For an average number of 30 impacts per pixels, the DSN associated with the SOT produces visually satisfying results, and is thus more competitive than the spectral simulation algorithm. The direct simulation of the DSN is simple and allows parallel local evaluation using standard computer graphics techniques for the Poisson process simulation [5] (a GPU implementation can produce 60fps for a \( 1024 \times 1024 \) image on a CUDA server, using a DSN with 30 impacts per pixel).

An interesting perspective would be to extend this procedure to a continuous framework for procedural texture synthesis.

References
    tion of random phase and Gaussian textures,” proc. IACSSP,
    Phase and Gaussian textures,” in preparation.
    Theory and Synthesis,” IEEE Trans. on Image Processing
    and Mixing Stationary Gaussian Texture Models”, SIAM J.